A Bregman-Kaczmarz method for nonlinear systems of equations

Robert Gower [*]   Dirk A. Lorenz [†‡]   Maximilian Winkler [§¶]

February 26, 2024

Abstract

We propose a new randomized method for solving systems of nonlinear equations, which can find sparse solutions or solutions under certain simple constraints. The scheme only takes gradients of component functions and uses Bregman projections onto the solution space of a Newton equation. In the special case of euclidean projections, the method is known as nonlinear Kaczmarz method. Furthermore if the component functions are nonnegative, we are in the setting of optimization under the interpolation assumption and the method reduces to SGD with the recently proposed stochastic Polyak step size. For general Bregman projections, our method is a stochastic mirror descent with a novel adaptive step size. We prove that in the convex setting each iteration of our method results in a smaller Bregman distance to exact solutions as compared to the standard Polyak step. Our generalization to Bregman projections comes with the price that a convex one-dimensional optimization problem needs to be solved in each iteration. This can typically be done with globalized Newton iterations. Convergence is proved in two classical settings of nonlinearity: for convex nonnegative functions and locally for functions which fulfill the tangential cone condition. Finally, we show examples in which the proposed method outperforms similar methods with the same memory requirements.

AMS Classification: 49M15, 90C53, 65Y20

Keywords: Nonlinear systems, stochastic methods, randomized Kaczmarz, Bregman projections

---

[*]CCM, Flatiron Institute, Simons Foundation, gowerrobert@gmail.com

[†]Institute of Analysis and Algebra, TU Braunschweig, d.lorenz@tu-braunschweig

[‡]Center for Industrial Mathematics, Fachbereich 3, University of Bremen, d.lorenz@uni-bremen.de

[§]Insitute of Analysis and Algebra, TU Braunschweig, maximilian.winkler@tu-braunschweig.de

[¶]Center for Industrial Mathematics, Fachbereich 3, University of Bremen, maxwin@uni-bremen.de

1 Introduction

We consider a constrained nonlinear system of equations

$$f(x) = 0 \qquad \text{s.t. } x \in C, \tag{1}$$

where $f \colon D \subset \mathbb{R}^d \to \mathbb{R}^n$ is a nonlinear differentiable function and $C \subset D \subset \mathbb{R}^d$ is a nonempty closed convex set. Let $S \subset C$ be the set of solutions of (1). Our aim is to design an iterative method which approximates a solution of (1) and in each step uses first-order information of just a single component function f_i.

The idea of our method is as follows. Given an appropriate convex function $\varphi \colon \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ with

$$\overline{\text{dom } \partial\varphi} = C, \tag{2}$$

our method computes the *Bregman projection* w.r.t. φ onto the solution set of the local linearization of a component function f_i around the current iterate x_k. Here, the underlying distance is the *Bregman distance* defined by

$$D_\varphi^{x^*}(x, y) = \varphi(y) - \varphi(x) - \langle x^*, y - x \rangle,$$

where x^* is a subgradient of φ at x. That is, the method we study is given by

$$x_{k+1} = \arg\min_{x \in \mathbb{R}^d} D_\varphi^{x_k^*}(x_k, x) \qquad \text{s.t. } x \in H_k, \tag{3}$$

where

$$H_k := \{x \in \mathbb{R}^d : f_{i_k}(x_k) + \langle \nabla f_{i_k}(x_k), x - x_k \rangle = 0\},$$

where $i_k \in \{1, \dots, n\}$ and x_k^* is in the subgradient $\partial\varphi(x_k)$. Since one can show that Bregman projections are always contained in dom $\partial\varphi$, the condition (2) guarantees that $x_k \in C$ holds for all k and hence, if the x_k converge, they converge to a point in C. In order for the Bregman projection x_{k+1} to exist, we need that the hyperplanes H_k have nonempty intersection with dom φ. Proposition 2.3 below will show that the slightly stronger condition

$$H_k \cap \text{ri dom } \varphi \neq \emptyset \tag{4}$$

is sufficient for existence and uniqueness of the Bregman projection under regularity assumptions on φ. If (4) is violated, we propose to compute a relaxed projection, which is always defined and inspired by the recently proposed *mSPS method* [23].

1.1 Related work and our contributions

Nonlinear Kaczmarz method and Sparse Kaczmarz. In the pioneering work by Stefan Kaczmarz [35], the idea of solving systems of equations by cycling through the separate equations and solving them incrementally was first executed on linear systems in finite dimensional spaces, an approach which is

known henceforth as *Kaczmarz method*. In this conceptually simple method, an update is computed by selecting one equation of the system according to a rule that may be random, cyclic or adaptive, and computing an orthogonal projection onto its solution space, which is given by a hyperplane.

Recently, two completely different extensions of the Kaczmarz method have been developed. One idea was to transfer the method to systems with nonlinear differentiable functions by considering its local linearizations: In each step k, an equation i_k is chosen and the update x_{k+1} is defined as the orthogonal projection

$$x_{k+1} = \operatorname{argmin} \|x - x_k\|_2^2 \quad \text{s.t.} \quad f_{i_k}(x_k) + \langle \nabla f_{i_k}(x_k), x - x_k \rangle = 0.$$

It is easy to check that this update can be computed by

$$x_{k+1} = x_k - \frac{f_{i_k}(x_k)}{\|\nabla f_{i_k}(x_k)\|_2^2} \nabla f_{i_k}(x_k).$$

This method was studied under the names *Sketched Newton-Raphson* [61] or *Nonlinear Kaczmarz method* [58]. Convergence was shown for two kinds of mild nonlinearities, namely star convex functions [61] and functions which obey a local tangential cone condition [58].

A different kind of extension of the Kaczmarz method has been proposed by [39]. Here, the notion of projection was replaced by the (more general) Bregman projection, giving rise to the 'sparse' Kaczmarz method, which can find sparse solutions of the system. The method has been further extended to inconsistent systems [54], accelerated by block averaging [57] and investigated as a regularization method in Banach spaces [33]. But so far only linear systems have been addressed.

The present article unifies these two generalizations, that is, we study the case of nonlinearity and general Bregman projections onto linearizations and derive convergence rates in the two aforementioned nonlinear settings. We also demonstrate that instead of sparsity, the proposed method is able to handle simple constraints such as simplex constraints as well.

Stochastic Polyak step size (SPS). One popular method for solving the finite-sum problem $\min \frac{1}{n} \sum_{i=1}^{n} \ell_i(x)$ is stochastic gradient descent (SGD), which is defined by the update $x_{k+1} = x_k - \gamma_k \nabla \ell_{i_k}(x_k)$. It is still a challenging question if there exist good choices of step sizes which are adaptive in the sense that no hyperparameter tuning is necessary. In this context, the *stochastic Polyak step size* (SPS)

$$\gamma_k = \frac{\ell_{i_k}(x_k) - \hat{\ell}_{i_k}}{c \cdot \|\nabla \ell_{i_k}(x_k)\|_2^2} \tag{5}$$

was proposed in [38], where $c > 0$ is a fixed constant and $\hat{\ell}_i = \inf \ell_i$. It was shown that the iterates of this method converge for convex lower bounded functions f_i for which the *interpolation* condition holds, meaning that there exists $\hat{x} \in \mathbb{R}^d$ with $\ell_i(\hat{x}) = \hat{\ell}_i$ for all $i = 1, ..., n$. This assumption is strong, but can be fulfilled e.g. by modern machine learning applications such as non-parametric regression

or over-parametrized deep neural networks [40, 63]. We cover these assumptions with our framework as a special case by requiring that the functions f_i in (1) are nonnegative, which is clear by setting $f_i = \ell_i - \hat{\ell}_i$. The SPS method applied to $\ell_1, ..., \ell_n$ then coincides with the Nonlinear Kaczmarz method applied to $f_1, ..., f_n$.

Mirror Descent and SPS. For incorporating additional constraints or attraction to sparse solutions into SGD, a well-known alternative to projected SGD is the *stochastic mirror descent* method (SMD) [3, 44, 64], which is defined by the update

$$x_{k+1} \in \operatorname*{argmin}_{x \in \mathbb{R}^d} \ \gamma_k \langle \nabla f_{i_k}(x_k), x - x_k \rangle + D_\varphi^{x_k^*}(x_k, x).$$

Here, φ is a convex function with additional properties which will be refined later on, which is then called the *distance generating function* (DGF), x_k^* is a subgradient of φ at x_k and D_φ is the Bregman distance associated to φ. We demonstrate that our proposed method can be reinterpreted as mirror descent with a novel adaptive step size in case that the f_i are nonnegative. Moreover, for $\varphi(x) = \frac{1}{2}\|x\|_2^2$, we obtain back the SGD method with the stochastic Polyak step size. For general φ, computing the step size requires the solution of a convex one-dimensional minimization problem. This is a similar situation as in the update of the stochastic dual coordinate ascent method [56], a popular stochastic variance reduced method for minimizing regularized general linear models.

The two recent independent works [23] and [60] propose to use the stochastic Polyak step size from SGD in mirror descent. This update has the advantage that it is relatively cheap to compute. However, we prove that for convex functions, our proposed method takes bigger steps in terms of Bregman distance towards the solution of (1). We generalize the step size from [23] to the case in which the functions f_i are not necessarily nonnegative and employ this update as a *relaxed projection* whenever our iteration is not defined. We compare our proposed method with the method which always performs relaxed projections in our convergence analysis and experiments. As an additional contribution, we improve the analysis for the method in [23] for the case of smooth strongly convex functions f_i (Theorem 4.16).

Finally, our method is by definition scaling-invariant in the sense that a multiplicative change $\varphi \mapsto \alpha\varphi$ of the DGF φ with a constant $\alpha > 0$ does not affect the method. To the best of our knowledge, this is the first mirror descent method which has this property.

Bregman projection methods. The idea of using Bregman projections algorithmically dates back to the seminal paper [11], which proposed to solve the *feasibility problem*

$$\text{find} \quad \hat{x} \in \bigcap_{i=1}^n C_i$$

for convex sets C_i by iterated Bregman projections onto the sets C_i. This idea initiated an active line of research [1, 4, 5, 6, 15, 16, 17, 18, 36, 50] with

4

applications in fields such as matrix theory [21], image processing [19, 20, 46] and optimal transport [9, 37]. We can view problem (1) as a feasibility problem by setting $C_i = \{x \mid f_i(x) = 0\}$. Our approach to compute Bregman projections onto linearizations has already been proposed for the case of convex inequalities $C_i = \{x \mid f_i(x) \leq 0\}$ under the name *outer Bregman projections* [13, 14]. Convergence of this method was studied in general Banach spaces. Obviously, the two problems coincide for convex nonnegative functions f_i. However, to the best of our knowledge, convergence rates have been given recently only in the case that the C_i are hyperplanes [36]. In this paper, we derive rates in the space $\mathbb{R}^d$. Also, we extend our analysis to the nonconvex setting for equality constraints.

Bregman-Landweber methods. There are a couple of works in inverse problems, typically studied in Banach spaces, which already incorporate Bregman projections into first-order methods with the aim of finding sparse solutions. Bregman projections were combined with the nonlinear Landweber iteration the first time in [55]. Later, [10] employed Bregman projections for L_1- and TV-regularization. A different nonlinear Landweber iteration with Bregman projections for sparse solutions of inverse problems was investigated in [41]. All of these methods use the full Jacobian $Df(x)$ in each iteration. In [32, 34], a deterministic Kaczmarz method incorporating convex penalties was proposed which performs a similar mirror update as our method, but with a different step size which does not originate from a Bregman projection. The apparently closest related method to our proposed one was recently suggested in [28], where the step size was calculated as the solution of a quite similar optimization problem, which is still different and also does not come with the motivation of a Bregman projection.

Sparse and Bregman-Newton methods. Finally, since our proposed method can be seen as a stochastic first-order Newton iteration, we briefly point out that a link of Newton's method to topics like Bregman distances and sparsity has already been established in the literature. Iusem and Solodov [30] introduced a regularization of Newton's method by a Bregman distance. Nesterov and Doikov [22] continued this work by introducing an additional nonsmooth convex regularizer. Polyak and Tremba [48] proposed a sparse Newton method which solves a minimum norm problem subject to the full Newton equation in each iteration, and presented an application in control theory [49].

1.2 Notation

For a set $S \subset \mathbb{R}^d$, we write its interior as S°, its closure as $\overline{S}$ and its relative interior as ri(S). The Cartesian product of sets $S_i \subset \mathbb{R}^d$, $i = 1, ..., m$, is written as $\bigtimes_{i=1}^m S_i$. The set span(S) is the linear space generated by all elements of S. We denote by $\mathbb{1}_d$ the vector in $\mathbb{R}^d$ with constant entries 1. For two vectors $x, y \in \mathbb{R}^d$, we express the componentwise (Hadamard) product as $x \cdot y$ and the componentwise logarithm and exponential as $\log(x)$ and $\exp(x)$. For a given norm $\| \cdot \|$ on $\mathbb{R}^d$, by $\| \cdot \|_*$ we denote the corresponding *dual norm*, which is

given as

$$\|x\|_* = \sup_{\|y\|=1} \langle x, y \rangle, \qquad x \in \mathbb{R}^d.$$

2 Basic notions and assumptions

We collect some basic notions and results as well as our standing assumptions for problem (1).

2.1 Convex analysis and standing assumptions

Let $\varphi \colon \mathbb{R}^d \to \overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$ be convex with

$$\operatorname{dom} \varphi = \{x \in \mathbb{R}^d : \varphi(x) < \infty\} \neq \emptyset.$$

We also assume that φ is *lower semicontinuous*, i.e. $\varphi(x) \leq \liminf_{y \to x} \varphi(y)$ holds for all $x \in \mathbb{R}^d$, and *supercoercive*, meaning that

$$\lim_{\|x\| \to \infty} \frac{\varphi(x)}{\|x\|} = +\infty.$$

The *subdifferential* at a point $x \in \operatorname{dom} \varphi$ is defined as

$$\partial \varphi(x) = \big\{x^* \in \mathbb{R}^d : \varphi(x) + \langle x^*, y - x \rangle \leq \varphi(y) \quad \text{for all } y \in \operatorname{dom} \varphi\big\}.$$

An element $x^* \in \partial \varphi(x)$ is called a *subgradient* of φ at x. The set of all points x with $\partial \varphi(x) \neq \emptyset$ is denoted by $\operatorname{dom} \partial \varphi$. Note that the relative interior of $\operatorname{dom} \varphi$ is a convex set, while $\operatorname{dom} \partial \varphi$ may not be convex, for a counterexample see [52, p.218]. In general, convexity of φ guarantees the inclusions ri $\operatorname{dom} \varphi \subset \operatorname{dom} \partial \varphi \subset \operatorname{dom} \varphi$. For later purposes, we require that $\operatorname{dom} \partial \varphi = $ ri $\operatorname{dom} \varphi$, which will be fulfilled in all our examples. We further assume that φ is *essentially strictly convex*, i.e. strictly convex on ri $\operatorname{dom} \varphi$. (In general, this property only means strict convexity on every convex subset of $\operatorname{dom} \partial \varphi$.). The convex conjugate (or Fenchel-Moreau-conjugate) of φ is defined by

$$\varphi^*(x^*) = \sup_{x \in \mathbb{R}^d} \langle x^*, x \rangle - \varphi(x), \qquad x^* \in \mathbb{R}^d.$$

The function φ^* is convex and lower semicontinuous. Moreover, the essential strict convexity and supercoercivity imply that $\operatorname{dom} \varphi^* = \mathbb{R}^d$ and φ^* is differentiable, since φ is essentially strictly convex and supercoercive, see [7, Proposition 14.15] and [52, Theorem 26.3].

The *Bregman distance* $D_\varphi^{x^*}(x, y)$ between $x, y \in \operatorname{dom} \varphi$ with respect to φ and a subgradient $x^* \in \partial \varphi(x)$ is defined as

$$D_\varphi^{x^*}(x, y) = \varphi(y) - \varphi(x) - \langle x^*, y - x \rangle.$$

Using Fenchel's equality $\varphi^*(x^*) = \langle x^*, x \rangle - \varphi(x)$ for $x^* \in \partial \varphi(x)$, one can rewrite the Bregman distance with the conjugate function as

$$D_\varphi^{x^*}(x, y) = \varphi^*(x^*) - \langle x^*, y \rangle + \varphi(y). \tag{6}$$

If φ is differentiable at x, then the subdifferential $\partial\varphi(x)$ contains the single element $\nabla\varphi(x)$ and we can write

$$D_\varphi(x,y) := D_\varphi^{\nabla\varphi(x)}(x,y) = \varphi(y) - \varphi(x) - \langle\nabla\varphi(x), y - x\rangle.$$

The function φ is called σ-*strongly* convex w.r.t. a norm $\|\cdot\|$ for some $\sigma > 0$, if for all $x, y \in \mathrm{dom}\,\partial\varphi$ it holds that $\frac{\sigma}{2}\|x - y\|^2 \le D_\varphi^{x^*}(x,y)$.

In conclusion, we require the following standing assumptions for problem (1):

Assumption 1.

(i) *The set C is nonempty, convex and closed.*

(ii) *It holds that $\varphi\colon \mathbb{R}^d \to \overline{\mathbb{R}}$ is essentially strictly convex, lower semicontinuous and supercoercive.*

(iii) *The function φ fulfills that $\overline{\mathrm{dom}\,\partial\varphi} = C$ and $\mathrm{dom}\,\partial\varphi = \mathrm{ri}\,\mathrm{dom}\,\varphi$.*

(iv) *For each $x \in \mathrm{dom}\,\varphi$ and each sequence $x_k \in \mathrm{dom}\,\partial\varphi$ with $x_k^* \in \partial\varphi(x_k)$ and $x_k \to x$ it holds that $D_\varphi^{x_k^*}(x_k, x) \to 0$.*

(v) *The function $f\colon D \to \mathbb{R}^n$ is continuously differentiable with $D \supset C$.*

(vi) *The set of solutions S of (1) is non-empty, that is $S := C \cap f^{-1}(0) \ne \emptyset$.*

2.2 Bregman projections

Definition 2.1. *Let $E \subset \mathbb{R}^d$ be a nonempty convex set, $x \in \mathrm{dom}\,\partial\varphi$ and $x^* \in \partial\varphi(x)$. Assume that $E \cap \mathrm{dom}\,\varphi \ne \emptyset$. The* Bregman projection *of x onto E with respect to φ and x^* is the point $\Pi_{\varphi,E}^{x^*}(x) \in E \cap \mathrm{dom}\,\varphi$ such that*

$$D_\varphi^{x^*}\left(x, \Pi_{\varphi,E}^{x^*}(x)\right) = \min_{y \in E} D_\varphi^{x^*}(x, y).$$

Existence and uniqueness of the Bregman projection is guaranteed if $E \cap \mathrm{dom}\,\varphi \ne \emptyset$ by our standing assumptions due to the fact that the function $y \mapsto D_\varphi^{x^*}(x,y)$ is lower bounded by zero, coercive, lower semicontinuous and strictly convex. For the standard quadratic $\varphi = \frac{1}{2}\|\cdot\|_2^2$, the Bregman projection is just the orthogonal projection. Note that if $E \cap \mathrm{dom}\,\varphi = \emptyset$, then for all $y \in E$ it holds that $D_\varphi^{x^*}(x,y) = +\infty$.

The Bregman projection can be characterized by variational inequalities, as the following lemma shows.

Lemma 2.2 ([39]). *A point $z \in E$ is the Bregman projection of x onto E with respect to φ and $x^* \in \partial\varphi(x)$ if and only if there exists $z^* \in \partial\varphi(z)$ such that one of the following conditions is fulfilled:*

(i) $\langle z^* - x^*, z - y\rangle \le 0$ *for all $y \in E$,*

(ii) $D_\varphi^{z^*}(z,y) \le D_\varphi^{x^*}(x,y) - D_\varphi^{x^*}(x,z)$ *for all $y \in E$.*

We consider Bregman projections onto hyperplanes

$$H(\alpha, \beta) := \{x \in \mathbb{R}^d : \langle \alpha, x \rangle = \beta\}, \qquad \alpha \in \mathbb{R}^d, \ \beta \in \mathbb{R},$$

and halfspaces

$$H^{\leq}(\alpha, \beta) := \{x \in \mathbb{R}^d : \langle \alpha, x \rangle \leq \beta\}, \qquad \alpha \in \mathbb{R}^d, \ \beta \in \mathbb{R},$$

and analoguously we define $H^{\geq}(\alpha, \beta)$.

The following proposition shows that the Bregman projection onto a hyperplane can be computed by solving a one-dimensional dual problem under a qualification constraint. We formulate this one-dimensional dual problem under slightly more general assumptions than previous versions, e.g. we neither assume smoothness of φ (as e.g. [5, 6, 11, 17, 21]) nor strong convexity of φ (as in [39]).

Proposition 2.3. *Let φ fulfill Assumption 1(ii). Let $\alpha \in \mathbb{R}^d \setminus \{0\}$ and $\beta \in \mathbb{R}$ such that*

$$H(\alpha, \beta) \cap \operatorname{ri} \operatorname{dom} \varphi \neq \emptyset.$$

Then, for all $x \in \operatorname{dom} \partial\varphi$ and $x^* \in \partial\varphi(x)$, the Bregman projection $\Pi_{\varphi, H(\alpha,\beta)}^{x^*}(x)$ exists and is unique. Moreover, the Bregman projection is given by

$$x_+ := \Pi_{\varphi, H(\alpha,\beta)}^{x^*}(x) = \nabla\varphi^*(x_+^*),$$

where $x_+^* = x^* - \hat{t}\alpha \in \partial\varphi(x_+)$ and $\hat{t}$ is a solution to

$$\min_{t \in \mathbb{R}} \varphi^*(x^* - t\alpha) + \beta t. \tag{7}$$

Proof. The assumptions guarantee that φ^* is finite and differentiable on the full space $\mathbb{R}^d$. We already know that the Bregman projection x_+ exists and is unique. Fermat's condition applied to the projection problem $\min_{y \in H(\alpha,\beta)} D_\varphi^{x^*}(x, y)$ states that

$$0 \in \partial\left(D_\varphi^{x^*}(x, \cdot) + \iota_{H(\alpha,\beta)}\right)(x_+),$$

where the indicator function $\iota_M \colon \mathbb{R}^d \to \overline{\mathbb{R}}$ is defined by

$$\iota_M(x) = \begin{cases} 0, & x \in M, \\ +\infty, & \text{otherwise.} \end{cases}$$

Applying subdifferential calculus [52, Theorem 23.8], where we make use of the fact that $H(\alpha, \beta)$ is a polyhedral set, we conclude that $x_+ \in \operatorname{dom} \partial\varphi$ and

$$0 \in \partial\varphi(x_+) - x^* + \operatorname{span}(\{\alpha\}),$$

where we used the fact that it holds $\partial\iota_{H(\alpha,\beta)} = \operatorname{span}(\{\alpha\})$ on $H(\alpha, \beta)$. Using subgradient inversion $(\partial\varphi)^{-1} = \nabla\varphi^*$, we arrive at the identity

$$x_+ = \nabla\varphi^*(x^* - \hat{t}\alpha)$$

with some $\hat{t} \in \mathbb{R}$. Inserting this equation into the constraint $\langle x_+, \alpha \rangle = \beta$, we conclude that $\hat{t}$ minimizes (7). $\qquad \square$

3 Realizations of the method

To solve problem (1), we propose the following method. In each step, we randomly pick a component equation $f_{i_k}(x) = 0$ and consider the set of zeros of its linearization around the current iterate x_k. This set is just the hyperplane

$$H_k := \left\{ x \in \mathbb{R}^d : f_{i_k}(x_k) + \langle \nabla f_{i_k}(x_k), x - x_k \rangle = 0 \right\} = H(\nabla f_{i_k}(x_k), \beta_k),$$

where

$$\beta_k = \langle \nabla f_{i_k}(x_k), x_k \rangle - f_{i_k}(x_k). \tag{8}$$

For later purposes, we also consider the halfspace

$$H_k^{\leq} := \{ x \in \mathbb{R}^d : f_{i_k}(x_k) + \langle f_{i_k}(x_k), x - x_k \rangle \leq 0 \}.$$

As the update x_{k+1}, we now propose to take the Bregman projection of x_k onto the set H_k using Proposition 2.3, which is possible if

$$H_k \cap \operatorname{dom} \partial \varphi \neq \emptyset. \tag{9}$$

The update is then given by $x_{k+1}^* = x_k^* - t_{k,\varphi} \nabla f_{i_k}(x_k)$ and $x_{k+1} = \nabla \varphi^*(x_{k+1}^*)$ with

$$t_{k,\varphi} \in \operatorname*{argmin}_{t \in \mathbb{R}} \varphi^*(x_k^* - t \nabla f_{i_k}(x_k)) + \beta_k t. \tag{10}$$

Note that, although the Bregman projection x_{k+1} is unique, $t_{k,\varphi}$ might not be unique. If (9) is not fulfilled, we define an update inspired from [23] by setting $x_{k+1} = \nabla \varphi^*(x_k^* - t_{k,\sigma} \nabla f_{i_k}(x_k))$ with the Polyak-like step size [1]

$$t_{k,\sigma} = \sigma \frac{f_{i_k}(x_k)}{\|\nabla f_{i_k}(x_k)\|_*^2} \tag{11}$$

with some norm $\| \cdot \|_*$ and some constant $\sigma > 0$. We will refer to the resulting update as the *relaxed projection*. We note that it is always defined and gives a new point $x_{k+1} \in \operatorname{dom} \partial \varphi$. However, x_{k+1} does not lie in H_k: Indeed, if it would lie in $H_k \cap \operatorname{dom} \partial \varphi$, this would contradict the assumption that (9) is not fulfilled. In [23], the similar step size $t = \sigma \frac{f_{i_k}(x_k) - \inf_{i_k} f_{i_k}}{c \|\nabla f_{i_k}(x_k)\|_*^2}$ with some constant $c > 0$ was proposed for minimization with mirror descent under the name *mirror-stochastic Polyak step size* (mSPS). Both the projection and relaxed projection guarantee that $x_{k+1} \in \operatorname{dom} \varphi$ and deliver a new subgradient x_{k+1}^* for the next update. The steps are summarized in Algorithm 1 below.

As an alternative method, we also consider the method which always chooses the step size $t_{k,\sigma}$ from (11).

Note that the problem (10) is convex and one-dimensional and can be solved with the bisection method, if φ^* is a C^1-function, or (globalized) Newton methods, if φ^* is a C^2-function, see Appendix A. In Example 3.2, Example 3.3, Example 3.4 and Example 3.5, we show how to implement the steps of Algorithm 1 for typical constraints.

---

[1] The typical setting in convergence analysis will be that φ is σ-strongly convex with respect to a norm $\| \cdot \|$, and $\| \cdot \|_*$ will be its dual norm.

Algorithm 1 Nonlinear Bregman-Kaczmarz (NBK) method

1: Input: $\sigma > 0$ and probabilities $p_i > 0$ for $i = 1, ..., n$
2: Initialization: $x_0^* \in \mathbb{R}^d, x_0 = \nabla\varphi^*(x_0^*)$
3: **for** $k = 0, 1, ...$ **do**
4:      choose $i_k \in \{1, ..., n\}$ according to the probabilities $p_1, ..., p_n$
5:      **if** $f_{i_k}(x_k) \neq 0$ and $\nabla f_{i_k}(x_k) \neq 0$ **then**
6:          ▷ otherwise, the component equation is solved already, or $H_k = \emptyset$
7:          set $\beta_k = \langle \nabla f_{i_k}(x_k), x_k \rangle - f_{i_k}(x_k)$
8:          **if** $H_k \cap \operatorname{dom} \partial\varphi \neq \emptyset$ **then**
9:             Find t_k: $t_k \in \operatorname{argmin}_{t \in \mathbb{R}} \varphi^*(x_k^* - t\nabla f_{i_k}(x_k)) + t\beta_k$
10:          **else** set $t_k = \sigma \frac{f_{i_k}(x_k)}{\|\nabla f_{i_k}(x_k)\|_*^2}$
11:          update $x_{k+1}^* = x_k^* - t_k \nabla f_{i_k}(x_k)$
12:          update $x_{k+1} = \nabla\varphi^*(x_{k+1}^*)$

Algorithm 2 Relaxed Nonlinear Bregman-Kaczmarz (rNBK) method

1: Input: $\sigma > 0$ and probabilities $p_i > 0$ for $i = 1, ..., n$
2: Initialization: $x_0^* \in \mathbb{R}^d, x_0 = \nabla\varphi^*(x_0^*)$
3: **for** $k = 0, 1, ...$ **do**
4:      choose $i_k \in \{1, ..., n\}$ according to the probabilities $p_1, ..., p_n$
5:      **if** $f_{i_k}(x) \neq 0$ and $\nabla f_{i_k}(x_k) \neq 0$ **then**
6:          set $t_k = \sigma \frac{f_{i_k}(x_k)}{\|\nabla f_{i_k}(x_k)\|_*^2}$
7:          update $x_{k+1}^* = x_k^* - t_k \nabla f_{i_k}(x_k)$
8:          update $x_{k+1} = \nabla\varphi^*(x_{k+1}^*)$

Remark 3.1 (Choice of σ in Algorithm 1 and Algorithm 2). *In this paper, we focus on the case that φ is a strongly convex function. In this setting, we propose to choose the parameter σ in Algorithm 1 and Algorithm 2 as the modulus of strong convexity, since in this case our theorems in Section 3 guarantee convergence.*

Example 3.2. (Unconstrained case and sparse Kaczmarz)
In the unconstrained case $\operatorname{dom} \varphi = \mathbb{R}^d$, condition (9) is always fulfilled whenever $\nabla f_{i_k}(x_k) \neq 0$.
For $\varphi(x) = \frac{1}{2}\|x\|_2^2$, we obtain back the nonlinear Kaczmarz method [43, 58, 61]

$$x_{k+1} = x_k - \frac{f_{i_k}(x_k)}{\|\nabla f_{i_k}(x_k)\|_2^2} \nabla f_{i_k}(x_k).$$

For the function

$$\varphi(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2, \tag{12}$$

Assumptions 1(i-iv) are fulfilled and it holds that $\varphi^*(x) = \frac{1}{2}\|S_\lambda(x)\|_2^2$ with the

soft-shrinkage function

$$S_\lambda(x) = \begin{cases} x + \lambda, & x < -\lambda, \\ 0, & |x| \leq \lambda, \\ x - \lambda, & x > \lambda \end{cases}$$

Hence, in this case lines 9, 11 and 12 of Algorithm 1 read

$$\text{find } t_k \in \underset{t \in \mathbb{R}}{\operatorname{argmin}} \beta_k t + \frac{1}{2} \|S_\lambda(x_k^* - t \nabla f_{i_k}(x_k))\|_2^2,$$
$$\text{update } x_{k+1}^* = x_k^* - t_k \nabla f_{i_k}(x_k),$$
$$\text{update } x_{k+1} = S_\lambda(x_{k+1}^*).$$

For affine functions $f_i(x) = \langle a^{(i)}, x \rangle - b_i$ with $a^{(i)} \in \mathbb{R}^d$, $b_i \in \mathbb{R}$, Algorithm 1 has been studied under the name Sparse Kaczmarz method *and converges to a sparse solution of the linear system $f(x) = 0$, see [39, 53]. The update with t_k from (10) is also called the* Exact step Sparse Kaczmarz *method. The line-search problem can be solved exactly with reasonable effort, as φ^* is a continuous piecewise quadratic function with at most $2d$ discontinuities. The corresponding solver is based on a sorting procedure and has complexity $\mathcal{O}(d \log(d))$, see [39] for details.*

Example 3.3. (Simplex constraints) *We consider the probability simplex*

$$C = \Delta^{d-1} := \big\{ x \in \mathbb{R}_{\geq 0}^d : \sum_{i=1}^d x_i = 1 \big\}.$$

The restriction of the negative entropy function

$$\varphi(x) = \begin{cases} \sum_{i=1}^d x_i \log(x_i), & x \in \Delta^{d-1}, \\ +\infty, & \text{otherwise.} \end{cases} \tag{13}$$

fulfills Assumption 1(i-iv) and is 1-strongly convex with respect to $\|\cdot\|_1$ due to Pinsker's inequality, see [25, 47] and [8, Example 5.27]. We have that

$$\operatorname{dom} \partial \varphi = \operatorname{ri} \Delta^{d-1} = \big\{ x \in \mathbb{R}_{>0}^d : \sum_{i=1}^d x_i = 1 \big\} =: \Delta_+^{d-1}.$$

We can characterize condition (9) easily as follows: The hyperplane $H(\alpha, \beta)$ intersects $\operatorname{dom} \partial \varphi = \Delta_+^{d-1}$ if and only if

- *$\alpha = \beta \mathbb{1}_d$ or*

- *there exist $r, s \in \{1, ..., d\}$ with $\alpha_r < \beta < \alpha_s$.*

11

This condition is quickly established by the intermediate value theorem and can be easily checked during the method. When verifying the condition in practice, in case of instabilities one may consider the restricted index set

$$\{i = 1, ..., n \mid |x_i| > \delta\}$$

for some positive δ.

The Bregman distance induced by φ is the Kullback-Leibler divergence for probability vectors

$$D_\varphi(x, y) = \sum_{i=1}^{d} y_i \log\left(\frac{y_i}{x_i}\right), \qquad x \in \Delta_+^{d-1}, y \in \Delta^{d-1}.$$

The convex conjugate of φ is the log-sum-exp-function $\varphi^*(p) = \log\left(\sum_{i=1}^{d} e^{p_i}\right)$. Since φ is differentiable, the steps of Algorithm 1 can be rewritten by substituting x_k^* by $\nabla\varphi(x_k) = 1 + \log(x_k)$. Denoting the ith component of an iterate x_l by $x_{l,i}$, lines 9, 11 and 12 of Algorithm 1 read

$$\text{find } t_k \in \operatorname*{argmin}_{t \in \mathbb{R}} \beta_k t + \log\left(\sum_{i=1}^{d} x_{k,i} e^{-t\partial_i f_{i_k}(x_k)}\right), \tag{14}$$

$$x_{k+1} = \frac{x_k \cdot e^{-t_k \nabla f_{i_k}(x_k)}}{\left\| x_k \cdot e^{-t_k \nabla f_{i_k}(x_k)} \right\|_1}, \tag{15}$$

where multiplication and exponentiation of vectors are understood component-wise. The method (15) is known with the name exponentiated gradient descent or entropic mirror descent, provided that t_k is nonnegative. We claim that our proposed step size $t_{k,\varphi}$ is new. Note that some convex polyhedra, such as ℓ^1-balls, can be transformed to $\Delta^{d'-1}$ for some $d' \in \mathbb{N}$ by writing a point as a convex combination of certain extreme points [23].

Example 3.4. (Cartesian products of constraints) For $i \in \{1, ..., m\}$, let φ_i be a DGF for $C_i \subset D_i \subset \mathbb{R}^{d_i}$ fulfilling Assumption 1(i-iv) and let $f : D := \bigtimes_{i=1}^{m} D_i \to \mathbb{R}^n$ fulfill Assumption 1(v-vi). Then

$$\varphi(x) = \sum_{i=1}^{m} \varphi_i(x_i), \quad x = (x_1, ..., x_m) \text{ with } x_i \in \mathbb{R}^d$$

is a DGF for $C = \bigtimes_{i=1}^{m} C_i$ fulfilling Assumption 1(i-iv) with

$$\operatorname{dom} \partial\varphi = \bigtimes_{i=1}^{m} \operatorname{dom} \partial\varphi_i \quad and \quad \partial\varphi(x) = \bigtimes_{i=1}^{m} \partial\varphi_i(x_i) \text{ for all } x_i \in \operatorname{dom} \partial\varphi_i.$$

Denoting the ith component of an iterate $x_l^{(*)}$ by $x_{l,i}^{(*)}$, the lines 9, 11 and 12 of Algorithm 1 for $i \in \{1, ..., m\}$ read as follows:

$$\text{find } t_k \in \operatorname*{argmin}_{t \in \mathbb{R}} \beta_k t + \sum_{i=1}^{m} \varphi_i^*\left(x_{k,i}^* - t\nabla_i f_{i_k}(x_k)\right),$$

12

$$x^*_{k+1,i} = x^*_{k,i} - t_k \nabla_i f_{i_k}(x_k) \qquad \text{for } i = 1, ..., m,$$
$$x_{k+1,i} = \nabla \varphi^*_i(x^*_{k+1,i}) \qquad \text{for } i = 1, ..., m,$$

where ∇_i stands for the gradient w.r.t. the ith block of variables.

Finally, we give a suggestion which constant σ and norm $\|\cdot\|_\infty$ should be used in Algorithm 2/ line 10 in Algorithm 1. Let us assume that φ is σ_i-strongly convex w.r.t. a norm $\|\cdot\|_{(i)}$ on $\mathbb{R}^{d_i}$. Then, the function φ is σ-strongly convex with $\sigma = \min_{i=1,...,m} \sigma_i$ w.r.t. the mixed norm

$$\|u\| := \sqrt{\sum_{i=1}^m \|u_i\|^2_{(i)}}.$$

Indeed, for all x, y with $x_i, y_i \in \mathbb{R}^{d_i}$ it holds that

$$\frac{\sigma}{2}\|x-y\|^2 = \frac{\sigma}{2}\sum_{i=1}^m \|x_i - y_i\|^2_{(i)} \leq \sum_{i=1}^m D^{x^*_i}_{\varphi_i}(x_i, y_i) = D^{x^*}_\varphi(x, y).$$

A quick calculation using Cauchy-Schwarz' inequality shows that the dual norm of $\|\cdot\|$ is given by

$$\|u^*\|_* := \sqrt{\sum_{i=1}^m \|u^*_i\|^2_{(i,*)}}, \qquad (16)$$

where $\|\cdot\|_{(i,*)}$ is the dual norm of $\|\cdot\|_i$ on $\mathbb{R}^{d_i}$. Hence, we recommend to use Algorithm 2 with (16) and $\sigma = \min_{i=1,...,m} \sigma_i$, if φ_i is σ_i-strongly convex w.r.t. $\|\cdot\|_{(i)}$.

Example 3.5. (Two-fold Cartesian product of simplex constraints) *As a particular instance of Example 3.4 we consider the 2-fold product of the probability simplex $C_i = \Delta^{d-1}$, $i \in \{1, 2\}$ with $\varphi_i = \varphi$ from Example 3.3. The properties from Assumption 1 are inherited from the φ_i. We denote the iterates of Algorithm 1 by $x_k, y_k \in \Delta^{d-1}$ and address its components by $x_{k,i}, y_{k,i}$ for $i = 1, ..., d$. Similar to Example 3.3, the steps of the method can be rewritten as*

$$\text{find } t_k \in \operatorname*{argmin}_{t \in \mathbb{R}} \beta_k t + \log\Big(\sum_{l=1}^d x_{k,l} e^{-t(\nabla_x f_{i_k}(x_k))_l}\Big) + \log\Big(\sum_{l=1}^d y_{k,l} e^{-t(\nabla_y f_{i_k}(x_k))_l}\Big),$$

$$(17)$$

$$x_{k+1} = \frac{x_k \cdot e^{-t_k \nabla_x f_{i_k}(x_k)}}{\|x_k \cdot e^{-t_k \nabla_x f_{i_k}(x_k)}\|_1}, \qquad y_{k+1} = \frac{y_k \cdot e^{-t_k \nabla_y f_{i_k}(y_k)}}{\|y_k \cdot e^{-t_k \nabla_y f_{i_k}(y_k)}\|_1},$$

where ∇_x stands for the gradient w.r.t. x and ∇_y for the gradient w.r.t. y. Also here, we can give a characterization of condition (9): For $\alpha = (\alpha_1, \alpha_2)$ with $\alpha_1, \alpha_2 \in \mathbb{R}^d$ and $\beta \in \mathbb{R}$, the hyperplane $H(\alpha, \beta)$ intersects dom $\partial\varphi = \Delta^{d+1}_+ \times \Delta^{d+1}_+$ if and only if for $(i, j) = (1, 2)$ or $(i, j) = (2, 1)$ one of the following conditions is fulfilled:

- $\alpha_i = c\mathbb{1}_d$ *with some* $c \in \mathbb{R}$ *and* $\alpha_j = (\beta - c)\mathbb{1}_d$,

- $\alpha_i = c\mathbb{1}_d$ *with some* $c \in \mathbb{R}$ *and there exist* $r, s \in \{1, ..., d\}$ *with*
  $\alpha_{j,r} < \beta - c < \alpha_{j,s}$ *or*

- $] \min \alpha_i, \max \alpha_i [\ \cap \ ] \beta - \max \alpha_j, \beta - \min \alpha_j [\ \neq \ \emptyset$.

To prove this condition, we can invoke Proposition 2.3 which states that (9) *is fulfilled if and only if the objective function g from* (7) *has a minimizer. Next, we note that, for each* $c \in \mathbb{R}$, *the objective in* (17) *can be rewritten as*

$$g(t) = \log \Big(\sum_{l=1}^{d} x_{k,l} e^{-(\alpha_{1,l}-c)t} \Big) + \log \Big(\sum_{l=1}^{d} y_{k,l} e^{(\beta-c-\alpha_{2,l})t} \Big)$$

$$= \log \Big(\sum_{l=1}^{d} y_{k,l} e^{-(\alpha_{2,l}-c)t} \Big) + \log \Big(\sum_{l=1}^{d} x_{k,l} e^{(\beta-c-\alpha_{1,l})t} \Big)$$

and a case-by-case analysis shows that g has a minimizer if and only if one of the above assertions is fulfilled. Note that also the here discussed condition can be easily checked during the method. We remind that, in case of instabilities one may consider the restricted index set

$$\{i = 1, ..., n \mid |x_i| > \delta \ \text{and} \ |y_i| > \delta\}$$

for some positive δ. *As derived in Example 3.4, in Algorithm 2/ line 10 of Algorithm 1 we use* $\sigma = 1$ *and* $\|u^*\|_* = \sqrt{\|u_1^*\|_\infty^2 + \|u_2^*\|_\infty^2}$.

4 Convergence

In this section we do the convergence analysis of Algorithm 1.

At first, we characterize fixed points of Algorithm 1 and Algorithm 2 and provide necessary lemmas for the subsequent analysis. In Section 4.1, we prove that for nonnegative star-convex functions f_i, condition (9) is always fulfilled and the step size (10) is better than the relaxed step size (11) in the sense that it results in an iterate with a smaller Bregman distance to all solutions of (1). Finally, we present convergence results for Algorithm 1 for this setting. In Section 4.2, we prove convergence in a second setting, namely in the case that the functions f_i fulfill a local tangential cone condition as in [34, 41, 58].

As a first result, we determine the fixed points of Algorithm 1. The proposition states in particular that, in the unconstrained case dom $\varphi = \mathbb{R}^d$, fixed points are exactly the stationary points of the least-squares function $\|f(x)\|_2^2$.

Proposition 4.1. *Let Assumption 1 hold and let* $x \in$ dom $\partial\varphi$ *and* $x^* \in \partial\varphi(x)$. *The pair* (x, x^*) *is a fixed point of Algorithm 1 if and only if for all* $i \in \{1, ..., n\}$ *it holds that* $f_i(x) = 0$ *or* $\nabla f_i(x) = 0$.

Proof. If $f_i(x) = 0$ or $\nabla f_i(x) = 0$ holds for all $i \in \{1, ..., n\}$, then (x, x^*) is a fixed point by definition of the steps. Next, we assume that $x \in \operatorname{dom} \partial\varphi$ is a fixed point of Algorithm 1 and $\nabla f_i(x) \neq 0$. First, assume that condition (9) is not fulfilled, then the update for x^* shows that $t_{k,\sigma} = 0$, since $\nabla f_i(x) \neq 0$, and hence, $f_i(x) = 0$. Finally, we assume that condition (9) holds. Then, from Proposition 2.3 we know that Algorithm 1 computes the Bregman projection $x = \Pi_{\varphi,H}^{x^*}(x)$ with

$$H = \big\{ y \in \mathbb{R}^d : f_i(x) + \langle \nabla f_i(x), y - x \rangle = 0 \big\}.$$

But this means that $x \in H$ and hence, $f_i(x) = 0$ holds also in this case. $\qquad \square$

The following fact will be useful in the convergence analysis. It shows that Algorithm 1 performs a mirror descent step whenever $f_i(x) > 0$, and a mirror ascent step whenever $f_i(x) < 0$.

Lemma 4.2. *Consider the kth iterate x_k of Algorithm 1 and consider the case that $f_{i_k}(x_k) \neq 0$ and $\nabla f_{i_k}(x_k) \neq 0$. Let Assumption 1 hold. Then, the step size t_k in Algorithm 1 fulfills*

$$\operatorname{sign}(t_k) = \operatorname{sign}(f_{i_k}(x_k)).$$

Proof. If condition (9) is not fulfilled, the assertion is clear by definition of the step size. Next, we assume that (9) holds. Then, the function

$$g_{i_k, x_k^*}(t) = \varphi^*(x_k^* - t\nabla f_{i_k}(x_k)) + t\big(\langle \nabla f_{i_k}(x_k), x_k \rangle - f_{i_k}(x_k) \big) \qquad (18)$$

is minimized by $t_{k,\varphi}$. (Note that the expression is indeed fully determined by i_k and x_k^* by the fact that $x_k = \nabla\varphi^*(x_k^*)$.) We compute

$$g'_{i_k, x_k^*}(0) = -\langle \nabla f_{i_k}(x_k), \ \nabla\varphi^*(x_k^*) \rangle + \langle \nabla f_{i_k}(x_k), x_k \rangle - f_{i_k}(x_k) = -f_{i_k}(x_k). \tag{19}$$

Since g_{i_k, x_k^*} is convex, its derivative is monotonically increasing and it vanishes at $t_{k,\varphi}$. Since it holds $f_{i_k}(x_k) \neq 0$ by assumption, we conclude that $t_{k,\varphi}$ and $f_{i_k}(x_k)$ have the same sign. $\qquad \square$

To exploit strong convexity and smoothness, we will use the following.

Lemma 4.3. *If $\varphi \colon \mathbb{R}^d \to \mathbb{R}$ is proper, convex and lower semicontinuous, then the following statements are equivalent:*

(i) *φ is σ-strongly convex w.r.t. $\|\cdot\|$.*

(ii) *For all $x, y \in \mathbb{R}^d$ and $x^* \in \partial\varphi(x)$, $y^* \in \partial\varphi(y)$,*

$$\langle x^* - y^*, x - y \rangle \geq \sigma \|x - y\|^2.$$

(iii) *The function φ^* is $\frac{1}{\sigma}$-smooth w.r.t. $\|\cdot\|_*$.*

15

Proof. See [62, Corollary 3.5.11 and Remark 3.5.3]. □

Lemma 4.4. *If $\varphi \colon \mathbb{R}^d \to \mathbb{R}$ is convex and lower semicontinuous, then the following statements are equivalent:*

(i) φ is L-smooth w.r.t. a norm $\|\cdot\|$,

(ii) $\varphi(y) \leq \varphi(x) + \langle \nabla\varphi(x), y - x \rangle + \frac{L}{2}\|x - y\|^2$ for all $x, y \in \mathbb{R}^d$,

(iii) $\langle \nabla\varphi(y) - \nabla\varphi(x), y - x \rangle \leq L\|x - y\|^2$ for all $x, y \in \mathbb{R}^d$.

Proof. See [62, Corollary 3.5.11 and Remark 3.5.3]. □

4.1 Convergence for nonnegative star-convex functions

In this subsection we assume in addition that the functions f_i are either nonnegative and star-convex or affine.

Definition 4.5 ([45]). *Let $f \colon D \to \mathbb{R}$ be differentiable. We say that f is called star-convex, if the set $\operatorname{argmin} f$ is nonempty and for all $x \in D$ and $\hat{x} \in \operatorname{argmin} f$ it holds that*

$$f(x) + \langle \nabla f(x), \hat{x} - x \rangle \leq f(\hat{x}).$$

Moreover, we call f strictly star-convex, if the above inequality is strict, and μ-strongly star-convex relative to φ, if for all $x \in D$, $x^* \in \partial\varphi(x)$ and $\hat{x} \in \operatorname{argmin} f$ it holds that

$$f(x) + \langle \nabla f(x), \hat{x} - x \rangle + \mu D_\varphi^{x^*}(x, \hat{x}) \leq f(\hat{x}).$$

We recall that the first assumption of nonnegativity and star-convexity covers two settings:

- Minimizing a sum-of-terms

$$\min \frac{1}{n} \sum_{i=1}^{n} f_i(x) \qquad \text{s.t. } x \in C, \tag{20}$$

  under the interpolation assumption

$$\exists x : \quad x \in \bigcap_{i=1}^{n} \operatorname{argmin} f_i|_C, \tag{21}$$

  where f_i is a star-convex function with known optimal value $\hat{f}_i$ on C. Under the interpolation assumption every point in the intersection on the right hand side of (21) is a solution to (20). Furthermore, we will construct a sequence which converges to this intersection point by applying Algorithm 1 to the nonnegative function $\tilde{f}$ where $\tilde{f}_i = f_i|_C - \hat{f}_i$. When $n = 1$, we cover the setting of mirror descent for the problem

$$\min f(x) \quad \text{s.t. } x \in C \tag{22}$$

  with known optimal value $\hat{f}$.

16

- Systems of nonlinear equations

$$f(x) = 0 \qquad \text{s.t. } x \in C$$

with star-convex component functions f_i, where we apply Algorithm 1 to $f_i^+ = \max(f_i, 0)$. Note that f_i^+ is not differentiable only at points x with $f_i(x) = 0$, which is anyway checked during the method.

Precisely, we will use the following assumption.

Assumption 2. *For each f_i one of the following conditions is fulfilled:*

(i) f_i *is nonnegative and star-convex and it holds that $f_i^{-1}(0) \cap \operatorname{dom} \partial\varphi \neq \emptyset$,*

(ii) f_i *is nonnegative and strictly star-convex or*

(iii) f_i *is affine.*

The first theorem states that Algorithm 1 always computes nonrelaxed Bregman projections under Assumption 2 outside of the fixed points.

Theorem 4.6. *Let (x_k, x_k^*) be given by Algorithm 1 and consider the case that $f_{i_k}(x) \neq 0$ and $\nabla f_{i_k}(x) \neq 0$. Let Assumptions 1-2 hold true. Then, the hyperplane H_k separates x_k and $f_{i_k}^{-1}(0)$, the condition*

$$H_k \cap \operatorname{dom} \partial\varphi \neq \emptyset$$

holds and the Bregman projection of x_k onto H_k is defined, namely it holds that

$$x_{k+1} = \Pi_{\varphi, H_k}^{x_k^*}(x_k).$$

In particular, Algorithm 1 always chooses the step size $t_k = t_{k,\varphi}$ from (10).

Proof. For $x \in D$ we define the affine function

$$\ell_x(y) := f_{i_k}(x) + \langle \nabla f_{i_k}(x), y - x \rangle.$$

We consider the cases (i) and (ii) from Assumption 2 first. Here we have that $\ell_{x_k}(x_k) = f_{i_k}(x_k) > 0$ and for all $\hat{x} \in f_{i_k}^{-1}(0)$, star-convexity of f_{i_k} shows that $\ell_{x_k}(\hat{x}) \leq 0$. This means that the hyperplane H_k separates x_k and $f_{i_k}^{-1}(0)$. By the intermediate value theorem there exists $x^\lambda = \lambda x_k + (1 - \lambda)\hat{x}$ for some $\lambda \in [0, 1[$ such that $\ell_{x_k}(x^\lambda) = 0$. Now let us assume that Assumption 2(i) holds, so we can choose $\hat{x} \in \operatorname{dom} \partial\varphi$ with $f_{i_k}(\hat{x}) = 0$. By Assumption 1(iii) it holds that $\operatorname{dom} \partial\varphi = \operatorname{ri} \operatorname{dom} \varphi$, which is a convex set and hence, $x^\lambda \in \operatorname{dom} \partial\varphi$. This proves that the claimed condition (9) is fulfilled and the update in Algorithm 1 computes a Bregman projection onto H_k by Proposition 2.3. In case of Assumption 2(ii) we have that $\ell_{x_k}(\hat{x}) < 0$ and therefore it even holds that $\lambda \in]0, 1[$. Assumption 1(iii) guarantees that $\hat{x} \in \overline{\operatorname{dom} \varphi}$ and $x_k \in \operatorname{dom} \partial\varphi = \operatorname{ri} \operatorname{dom} \varphi$. Hence, we have $x^\lambda \in \operatorname{ri} \operatorname{dom} \varphi$ by [52, Theorem 6.1], which again implies that $x^\lambda \in \operatorname{dom} \partial\varphi$ by Assumption 1(iii). This proves that condition (9)

17

is fulfilled in this case, too.

Under Assumption 2(iii) it holds that $\ell_x = f_{i_k}$ for all $x \in D$. This already implies that H_k separates x_k and $f_{i_k}^{-1}(0)$. Condition (9) is fulfilled by the assumption that $f_{i_k}^{-1}(0) \neq \emptyset$ and so, the update computes the claimed Bregman projection by Proposition 2.3 also in this case. $\qquad\square$

As an immediate consequence, we see that Algorithm 1 is stable in terms of Bregman distance.

Corollary 4.7. *If $\hat{x} \in S$ is a solution to* (1) *and the assumptions from Theorem 4.6 hold true, then it holds that*

$$D_\varphi^{x_{k+1}^*}(x_{k+1}, \hat{x}) \leq D_\varphi^{x_k^*}(x_k, \hat{x}) - D_\varphi^{x_k^*}(x_k, x_{k+1}).$$

Proof. By Theorem 4.6, x_{k+1} is the Bregman projection of x_k onto H_k with respect to x_k^*. If $f_{i_k}(x_k) = 0$, then x_k is a fixed point by Proposition 4.1and the statement holds trivially. Next, assume that $f_{i_k}(x_k) > 0$. Then by Lemma 4.2, we have that $t_k > 0$. As $x_{k+1} \in H$, we have that $\langle \nabla f_{i_k}(x_k), x_{k+1} - x_k \rangle = 0$. We conclude for all $y \in H_k^{\leq}$ that

$$
\begin{aligned}
\langle x_{k+1}^* - x_k^*, x_{k+1} - y \rangle &= -t_k \langle \nabla f_{i_k}(x_k), x_{k+1} - y \rangle \\
&= t_k \langle \nabla f_{i_k}(x_k), y - x_k \rangle - t_k \langle \nabla f_{i_k}(x_k), x_{k+1} - x_k \rangle \\
&\leq -t_k f_{i_k}(x_k) \\
&\leq 0,
\end{aligned}
$$

which by Lemma 2.2 shows that $x_{k+1} = \Pi_{\varphi, H_k^{\leq}}^{x_k^*}(x_k)$. As $\hat{x} \in H_k^{\leq}$, the claim follows from Lemma 2.2(ii). An analoguous argument shows the claim in the case $f_{i_k}(x_k) < 0$. $\qquad\square$

Next, we prove that the exact Bregman projection moves the iterates closer to solutions of (1) than the relaxed projections, where the distance is in the sense of the used Bregman distance (see Theorem 4.11). To that end, for (x_k, x_k^*) from Algorithm 1 we define an update with variable step size

$$x_{k+1}^*(t) = x_k^* - t\nabla f_{i_k}(x_k), \qquad x_{k+1}(t) = \nabla\varphi^*(x_{k+1}^*(t)), \quad t \in \mathbb{R}. \qquad (23)$$

Lemma 4.8. *Let (x_k, x_k^*) be given by Algorithm 1 and consider $(x_{k+1}(t), x_{k+1}^*(t))$ from* (23) *for some $t \in \mathbb{R}$. Let Assumption 1 hold true. Then, for all $x \in \mathbb{R}^d$ it holds that*

$$
\begin{aligned}
D_\varphi^{x_{k+1}^*(t)}(x_{k+1}(t), x) &= \varphi^*(x_k^* - t\nabla f_{i_k}(x_k)) + t\beta_k + t\langle \nabla f_{i_k}(x_k), x - x_k \rangle \\
&\quad + t f_{i_k}(x_k) - \langle x_k^*, x \rangle + \varphi(x).
\end{aligned}
$$

Proof. Rewriting the Bregman distance as in (6) shows that

$$D_\varphi^{x_{k+1}^*(t)}(x_{k+1}(t), x) = \varphi^*(x_{k+1}^*(t)) - \langle x_{k+1}^*(t), x \rangle + \varphi(x)$$

18

$$= \varphi^*(x_k^* - t\nabla f_{i_k}(x_k)) + t\langle \nabla f_{i_k}(x_k), x\rangle - \langle x_k^*, x\rangle + \varphi(x)$$
$$= \varphi^*(x_k^* - t\nabla f_{i_k}(x_k)) + t\beta_k + t\langle \nabla f_{i_k}(x_k), x - x_k\rangle$$
$$+ t f_{i_k}(x_k) - \langle x_k^*, x\rangle + \varphi(x).$$

$\square$

Proposition 4.9. *Let (x_k, x_k^*) and t_k be given by Algorithm 1 and consider $(x_{k+1}(t), x_{k+1}^*(t))$ from (23) for some $t \in \mathbb{R}$. Let Assumption 1 hold true. Then, for all $x \in \mathbb{R}^d$ it holds that*

$$D_\varphi^{x_{k+1}^*}(x_{k+1}, x) \le D_\varphi^{x_{k+1}^*(t)}(x_{k+1}(t), x) + (t_k - t) \cdot \left(f_{i_k}(x_k) + \langle \nabla f_{i_k}(x_k), x - x_k\rangle \right).$$

Proof. Note that t_k equals $t_{k,\varphi}$ from (10), since condition (9) is fulfilled by Theorem 4.6. Hence, the optimality property (10) shows that for any t we have that

$$\varphi^*(x_k^* - t_k\nabla f_{i_k}(x_k)) + t_k\beta_k \le \varphi^*(x_k^* - t\nabla f_{i_k}(x_k)) + t\beta_k.$$

Lemma 4.8 then shows that for any x it holds that

$$D_\varphi^{x_{k+1}^*}(x_{k+1}, x) \le \varphi^*(x_k^* - t\nabla f_{i_k}(x_k)) + t\beta_k + t_k\langle \nabla f_{i_k}(x_k), x - x_k\rangle + t_k f_{i_k}(x_k)$$
$$- \langle x_k^*, x\rangle + \varphi(x).$$

We use the definitions of β_k, $x_{k+1}(t)$ and $x_{k+1}^*(t)$ and get

$$D_\varphi^{x_{k+1}^*}(x_{k+1}, x) \le \varphi^*(x_k^* - t\nabla f_{i_k}(x_k)) + t\beta_k + t_k\langle \nabla f_{i_k}(x_k), x - x_k\rangle + t_k f_{i_k}(x_k)$$
$$- \langle x_k^*, x\rangle + \varphi(x)$$
$$= \varphi^*(x_k^* - t\nabla f_{i_k}(x_k)) + t\langle \nabla f_{i_k}(x_k), x\rangle - \langle x_k^*, x\rangle + \varphi(x)$$
$$+ (t_k - t) \cdot \left(f_{i_k}(x_k) + \langle \nabla f_{i_k}(x_k), x - x_k\rangle \right)$$
$$= D_\varphi^{x_{k+1}^*(t)}(x_{k+1}(t), x) + (t_k - t) \cdot \left(f_{i_k}(x_k) + \langle \nabla f_{i_k}(x_k), x - x_k\rangle \right).$$

$\square$

In order to draw a conclusion from Proposition 4.9, we relate the step sizes $t_{k,\varphi}$ from (10) and $t_{k,\sigma}$ from (11). We already know by Lemma 4.2 that both step sizes have the same sign. The next lemma gives upper and lower bounds for $t_{k,\varphi}$ with respect to $t_{k,\sigma}$ under additional assumptions on φ.

Lemma 4.10. *Let (x_k, x_k^*) be the iterates from Algorithm 1 and let Assumption 1 hold true. We consider $t_{k,\varphi}$ and $t_{k,\sigma}$ from (10) and (11) and the function g_{i_k, x_k^*} from (18).*

(i) *If φ is σ-strongly convex w.r.t. $\|\cdot\|$, then g_{i_k, x_k^*} is $\frac{\|\nabla f_{i_k}(x_k)\|_*^2}{\sigma}$-smooth and*

$$|t_{k,\varphi}| \ge \sigma \frac{|f_{i_k}(x_k)|}{\|\nabla f_{i_k}(x_k)\|_*^2} = |t_{k,\sigma}|. \tag{24}$$

(ii) If φ is M-smooth w.r.t. $\|\cdot\|$, then g_{i_k,x_k^*} is $\frac{\|\nabla f_{i_k}(x_k)\|_*^2}{M}$-strongly convex and

$$|t_{k,\varphi}| \le M \frac{|f_{i_k}(x_k)|}{\|\nabla f_{i_k}(x_k)\|_*^2} = \frac{M}{\sigma} \cdot |t_{k,\sigma}|. \tag{25}$$

Proof. For $s, t \in \mathbb{R}$ with $s < t$ it holds that

$$g'_{i_k,x_k^*}(t) - g'_{i_k,x_k^*}(s) = \langle \nabla\varphi^*(x_k^* - t\nabla f_{i_k}(x_k)) - \nabla\varphi^*(x_k^* - s\nabla f_{i_k}(x_k)), -\nabla f_{i_k}(x_k)\rangle$$

$$= \frac{1}{(t-s)}\langle \nabla\varphi^*(x_k^* - t\nabla f_{i_k}(x_k)) - \nabla\varphi^*(x_k^* - s\nabla f_{i_k}(x_k)),$$

$$x_k^* - t\nabla f_{i_k}(x_k) - (x_k^* - s\nabla f_{i_k}(x_k))\rangle. \tag{26}$$

(i) If φ is σ-strongly convex w.r.t. $\|\cdot\|$, then φ^* is $\frac{1}{\sigma}$-smooth w.r.t. $\|\cdot\|_*$ by Lemma 4.3(iv). Hence, by Lemma 4.4(iii) we can estimate

$$0 \le g'_{i_k,x_k^*}(t) - g'_{i_k,x_k^*}(s) \le \frac{\|(t-s)\nabla f_{i_k}(x_k)\|_*^2}{\sigma \cdot (t-s)} = \frac{\|\nabla f_{i_k}(x_k)\|_*^2}{\sigma} \cdot (t-s),$$

which proves by the same lemma that g_{i_k,x_k^*} is $\frac{\|\nabla f_{i_k}(x_k)\|_*^2}{\sigma}$-smooth. Hence, (24) follows by choosing $t = \max(t_{k,\varphi}, 0)$, $s = \min(t_{k,\varphi}, 0)$ and inserting (19).

(ii) Here, by Lemma 4.3 and the Fenchel-Moreau-identity $\varphi = \varphi^{**}$, the function φ^* is $\frac{1}{M}$-strongly convex w.r.t. $\|\cdot\|_*$. Using Lemma 4.3(ii) we can estimate

$$g'_{i_k,x_k^*}(t) - g'_{i_k,x_k^*}(s) \ge \frac{\|(t-s)\nabla f_{i_k}(x_k)\|_*^2}{M \cdot (t-s)} = \frac{\|\nabla f_{i_k}(x)\|_*^2}{M} \cdot (t-s)$$

which shows that g_{i_k,x_k^*} is $\frac{\|\nabla f_{i_k}(x_k)\|_*^2}{M}$-strongly convex. Inequality (25) then follows as in (i).

$\square$

Theorem 4.11. *Let (x_k, x_k^*) and t_k be given by Algorithm 1. Let $t \in [0, t_k]$ and let $(x_{k+1}(t), x_{k+1}^*(t))$ be as in (23). Let Assumptions 1-2 hold true and assume that $f_{i_k}(x_k) > 0$. Then for every solution $\hat{x} \in S$ it holds that*

$$D_\varphi^{x_{k+1}^*}(x_{k+1}, \hat{x}) \le D_\varphi^{x_{k+1}^*(t)}(x_{k+1}(t), \hat{x}).$$

If φ is σ-strongly convex, the inequality holds as well for $t = t_{k,\sigma}$.

Proof. We recall that t_k equals $t_{k,\varphi}$ from (10), since condition (9) is fulfilled by Theorem 4.6. By Lemma 4.2 we have that $t_k > 0$. Since f_{i_k} is star-convex and $\hat{x} \in S$, it holds that

$$f_{i_k}(x_k) + \langle \nabla f_{i_k}(x_k), \hat{x} - x_k\rangle \le 0.$$

The statement now follows from Proposition 4.9. The theorem applies in particular to $t = t_{k,\sigma}$ if φ is σ-strongly convex, as Lemma 4.10(i) ensures that $0 < t_{k,\sigma} \leq t_{k,\varphi}$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\square$

For mirror descent or stochastic mirror descent under interpolation, Theorem 4.11 tells that a choice of a smaller step size than $t_{k,\varphi}$ results in a larger distance to solutions $\hat{x}$ of problem (1) in Bregman distance.

The following lemma is the key element of our convergence analysis.

Lemma 4.12. Let (x_k, x_k^*) be the iterates of either Algorithm 1 or Algorithm 2. Let Assumptions 1-2 hold true and assume that φ is σ-strongly convex w.r.t. a norm $\|\cdot\|$. Then for every solution $\hat{x} \in S$ it holds that

$$D_\varphi^{x_{k+1}^*}(x_{k+1}, \hat{x}) \leq D_\varphi^{x_k^*}(x_k, \hat{x}) - \frac{\sigma}{2} \frac{\big(f_{i_k}(x_k)\big)^2}{\|\nabla f_{i_k}(x_k)\|_*^2}. \tag{27}$$

Proof. We bound the right-hand side in Lemma 4.8 from above for $t \in \{t_{k,\varphi}, t_{k,\sigma}\}$. As φ is σ-strongly convex, φ^* is $\frac{1}{\sigma}$-smooth by Lemma 4.3(iii). Hence, by Lemma 4.4(ii), for all $t \in \mathbb{R}$ we can estimate that

$$\varphi^*(x_k^* - t\nabla f_{i_k}(x_k)) + t\beta_k$$
$$= \varphi^*(x_k^* - t\nabla f_{i_k}(x_k)) + t\big(\langle \nabla f_{i_k}(x_k), x_k\rangle - f_{i_k}(x_k)\big)$$
$$\leq \varphi^*(x_k^*) - t\langle \nabla \varphi^*(x_k^*), \nabla f_{i_k}(x_k)\rangle + \frac{1}{2\sigma}t^2\|\nabla f_{i_k}(x_k)\|_*^2$$
$$\quad + t\big(\langle \nabla f_{i_k}(x_k), x_k\rangle - f_{i_k}(x_k)\big)$$
$$= \varphi^*(x_k^*) - tf_{i_k}(x_k) + \frac{1}{2\sigma}t^2\|\nabla f_{i_k}(x_k)\|_*^2.$$

Minimizing the right hand side over $t \in \mathbb{R}$ gives $\hat{t} = \sigma \frac{f_{i_k}(x_k)}{\|\nabla f_{i_k}(x_k)\|_*^2} = t_{k,\sigma}$ and

$$\varphi^*(x_k^*) - \hat{t}f_{i_k}(x_k) + \frac{1}{2\sigma}\hat{t}^2\|\nabla f_{i_k}(x_k)\|_*^2 = \varphi^*(x_k^*) - \frac{\sigma}{2}\frac{\big(f_{i_k}(x_k)\big)^2}{\|\nabla f_{i_k}(x_k)\|_*^2}.$$

By optimality of $t_{k,\varphi}$ we have that

$$\varphi^*(x_k^* - t_{k,\varphi}\nabla f_{i_k}(x_k)) + t_{k,\varphi}\beta_k \leq \varphi^*(x_k^* - t\nabla f_{i_k}(x_k)) + t\beta_k$$

for all $t \in \mathbb{R}$. Hence, we have shown that

$$\varphi^*(x_k^* - t\nabla f_{i_k}(x_k)) + t\beta_k \leq \varphi^*(x_k^*) - \frac{\sigma}{2}\frac{\big(f_{i_k}(x_k)\big)^2}{\|\nabla f_{i_k}(x_k)\|_*^2} \tag{28}$$

holds for $t \in \{t_{k,\sigma}, t_{k,\varphi}\}$. If Assumption 2(i) or 2(ii) are fulfilled, Lemma 4.2 and star-convexity of f_{i_k} show that

$$t_k\big(f_{i_k}(x_k) + \langle \nabla f_{i_k}(x_k), \hat{x} - x_k\rangle\big) \leq 0. \tag{29}$$

Under Assumption 2(iii), we have equality in (29). Inserting this inequality into Lemma 4.8, we obtain the claimed bound

$$
D_\varphi^{x^*_{k+1}}(x_{k+1}, \hat{x}) \leq \varphi^*(x^*_k) - \frac{\sigma}{2} \frac{\big(f_{i_k}(x_k)\big)^2}{\|\nabla f_{i_k}(x_k)\|_*^2} - \langle x^*_k, \hat{x}\rangle + \varphi(\hat{x})
$$

$$
= D_\varphi^{x^*_k}(x_k, \hat{x}) - \frac{\sigma}{2} \frac{\big(f_{i_k}(x_k)\big)^2}{\|\nabla f_{i_k}(x_k)\|_*^2},
$$

where we used (6) in the last step.

$\square$

We can now establish almost sure (a.s.) convergence of Algorithm 1. The expectations are always taken with respect to the random choice of the indices. Sometimes we also take conditional expectations conditioned on choices of indices in previous iterations, which we will indicate explicitly.

Theorem 4.13. *Let Assumptions 1-2 hold true and assume that φ is σ-strongly convex w.r.t. a norm $\|\cdot\|$. Then it holds that*

$$
\mathbb{E}\Big[\sum_{i=1}^n p_i\big(f_i(x_k)\big)^2\Big] \to 0 \quad as\ k \to \infty
$$

and we have the rate

$$
\mathbb{E}\Big[\min_{l=1,\dots,k} \sum_{i=1}^n p_i\big(f_i(x_l)\big)^2\Big] \leq \frac{c}{\sigma k}
$$

with some constant $c > 0$. Moreover, the iterates x_k of Algorithm 1 converge a.s. to a random variable whose image is contained in the solution set S.

Proof. By σ-strong convexity of φ and Lemma 4.12, we have that

$$
\frac{\sigma}{2}\|x_k - \hat{x}\|^2 \leq D_\varphi^{x^*_k}(x_k, \hat{x}) \leq D_\varphi^{x^*_0}(x_0, \hat{x})
$$

holds for all $k \in \mathbb{N}$ and $\hat{x} \in S$. Hence, the sequence x_k is bounded and we have

$$
\|\nabla f_{i_k}(x_k)\|_*^2 \leq M
$$

with some constant $M > 0$. Inserting this into (27) gives that for all $l \in \mathbb{N}$ it holds

$$
D_\varphi^{x^*_{l+1}}(x_{l+1}, \hat{x})) \leq D_\varphi^{x^*_l}(x_l, \hat{x}) - \frac{\sigma}{2M}\big(f_{i_l}(x_l)\big)^2.
$$

Taking conditional expectation w.r.t. $i_0, \dots, i_{l-1}$, we obtain

$$
\mathbb{E}\big[D_\varphi^{x^*_{l+1}}(x_{l+1}, \hat{x}) \mid i_0, \dots, i_{l-1}\big] \leq D_\varphi^{x^*_l}(x_l, \hat{x}) - \frac{\sigma}{2M}\sum_{i=1}^n p_i\big(f_{i_l}(x_l)\big)^2. \tag{30}
$$

22

By rearranging and using the tower property of conditional expectation, we conclude that

$$\mathbb{E}\Big[\sum_{i=1}^n p_i\big(f_{i_l}(x_l)\big)^2\Big] \leq \frac{2M}{\sigma}\Big(\mathbb{E}\big[D_\varphi^{x_l^*}(x_l,\hat{x})\big] - \mathbb{E}\big[D_\varphi^{x_{l+1}^*}(x_{l+1},\hat{x})\big]\Big)$$

The convergence rate now follows with $c = 2 \cdot M \cdot D_\varphi^{x_0^*}(x_0,\hat{x})$ for any $\hat{x} \in S$ by averaging over $l = 0, ..., k$ and telescoping.

Next, we prove the a.s. iterate convergence. Using (30) gives that

$$\mathbb{E}\big[D_\varphi^{x_{k+1}^*}(x_{k+1},\hat{x}) \mid i_0, ..., i_{k-1}\big] \leq D_\varphi^{x_k^*}(x_k,\hat{x}) - \frac{\sigma \cdot \min_i p_i}{2M} \cdot \|f(x_k)\|_2^2.$$

The Robbins-Siegmund Lemma [51] proves that $f(x_k) \to 0$ holds with probability 1. Along any sample path in $\{f(x_k) \to 0\}$, due to boundedness of the sequence x_k, there exists a subsequence x_{k_l} converging to some point x. By continuity, we have that $f(x) = 0$ and hence, $x \in S$. Due to Assumption 1(iv), it holds that $D_\varphi^{x_{k_l}^*}(x_{k_l},x) \to 0$ and since $D_\varphi^{x_k^*}(x_k,\hat{x})$ is a decreasing sequence in k for $\hat{x} \in S$ by Lemma 4.12, we conclude that $D_\varphi^{x_k^*}(x_k,x) \to 0$. Finally, strong convexity of φ implies that $x_k \to x$. $\qquad\square$

If the functions f_i have Lipschitz continuous gradient, we can derive a sublinear rate for the ℓ_1-kind loss $\mathbb{E}\big[\min_{l=1,...,k} \sum_{i=1}^n p_i f_i(x_l)\big]$, which coincides with the rate in [23, Theorem 4]. Note that without this assumption, Jensen's inequality gives the asymptotically slower rate

$$\mathbb{E}\Big[\min_{l=1,...,k} \sum_{i=1}^n p_i f_i(x_l)\Big] \leq \mathbb{E}\Big[\frac{1}{k} \sum_{l=1}^k \sum_{i=1}^n p_i f_i(x_l)\Big] \leq \frac{c}{\sqrt{k}}$$

for some constant $c > 0$. We will need the following lemma.

Lemma 4.14. *[38, Lemma 3] Let φ be σ-strongly convex w.r.t. $\|\cdot\|$. Moreover, let the functions f_i be L-smooth w.r.t. $\|\cdot\|$. Then it holds that*

$$t_{k,\sigma} \geq \frac{\sigma}{2L}.$$

Theorem 4.15. *Let Assumptions 1-2 hold true and assume that φ is σ-strongly convex w.r.t. a norm $\|\cdot\|$ and all functions f_i are L-smooth w.r.t. $\|\cdot\|$. Then the iterates x_k of Algorithm 1 fulfill that*

$$\mathbb{E}\Big[\min_{l=1,...,k} \sum_{i=1}^n p_i f_i(x_l)\Big] \leq \frac{4L}{\sigma k} \cdot \inf_{\hat{x}\in S} D_\varphi^{x_0^*}(x_0,\hat{x}).$$

Proof. Combining Lemma 4.12 and Lemma 4.14 yields that

$$D_\varphi^{x_{k+1}^*}(x_{k+1},\hat{x}) \leq D_\varphi^{x_k^*}(x_k,\hat{x}) - \frac{1}{2} f_{i_k}(x_k) \cdot t_{k,\sigma} \leq D_\varphi^{x_k^*}(x_k,\hat{x}) - \frac{\sigma}{4L} f_{i_k}(x_k).$$

The assertion now follows as in the proof of Theorem 4.13 by taking expectation and telescoping. $\qquad\square$

For strongly star-convex functions f_i, we can prove a linear convergence rate, where we recover the contraction factor from [23, Theorem 3]. Moreover, we can even improve this factor for smooth φ.

Theorem 4.16. *Let Assumptions 1-2 hold true and assume that φ is σ-strongly convex w.r.t. a norm $\|\cdot\|$ and all functions f_i are L-smooth w.r.t. $\|\cdot\|$. Moreover, assume that $\overline{f} := \sum_{i=1}^{d} p_i f_i$ is μ-strongly star-convex w.r.t. S relative to φ. Then there exists an element $\hat{x} \in S$ such that the iterates x_k of Algorithm 1 converge to $\hat{x}$ at the rate*

$$\mathbb{E}\big[D_\varphi^{x_{k+1}^*}(x_{k+1}, \hat{x})\big] \leq \big(1 - \frac{\mu\sigma}{2L}\big)\mathbb{E}\big[D_\varphi^{x_k^*}(x_k, \hat{x})\big] - \frac{\sigma}{4L}\overline{f}(x_k).$$

Moreover, if φ is M-smooth, it holds that

$$\mathbb{E}\big[D_\varphi^{x_{k+1}^*}(x_{k+1}, \hat{x})\big] \leq \big(1 - \frac{\mu\sigma}{2L} - \frac{\mu\sigma^2}{4LM}\big)\mathbb{E}\big[D_\varphi^{x_k^*}(x_k, \hat{x})\big].$$

Proof. By Lemma 4.14 we have that

$$t_k\big(\langle \nabla f_{i_k}(x_k), \hat{x} - x_k\rangle + f_{i_k}(x_k)\big) \leq \frac{\sigma}{2L}\big(\langle \nabla f_{i_k}(x_k), \hat{x} - x_k\rangle + f_{i_k}(x_k)\big).$$

Taking expectation and using the assumption of relative strong convexity, we obtain that

$$\mathbb{E}\big[t_k\big(\langle \nabla f_{i_k}(x_k), \hat{x} - x_k\rangle + f_{i_k}(x_k)\big)\big] \leq -\frac{\sigma}{2L}\mathbb{E}\big[\overline{f}(\hat{x}) - \overline{f}(x_k) - \langle \nabla\overline{f}(x_k), \hat{x} - x_k\rangle\big]$$
$$\leq -\frac{\mu\sigma}{2L}\mathbb{E}\big[D_\varphi^{x_k^*}(x_k, \hat{x})\big].$$

The first convergence rate then follows by the steps in Lemma 4.12 and Theorem 4.15, replacing (29) by the above inequality. Finally, let φ be additionally M-smooth. Using that $\nabla\overline{f}(\hat{x}) = 0$, we can further bound

$$\overline{f}(x_k) = \overline{f}(x_k) - \overline{f}(\hat{x}) - \langle \nabla\overline{f}(\hat{x}), x_k - \hat{x}\rangle$$
$$\geq \mu D_\varphi(\hat{x}, x_k) \geq \frac{\mu\sigma}{2}\|x_k - \hat{x}\|^2 \geq \frac{\mu\sigma}{M}D_\varphi^{x_k^*}(x_k, \hat{x}).$$

$\square$

Since the proofs of Theorem 4.13, Theorem 4.15 and Theorem 4.16 rely on Lemma 4.12, they also hold for Algorithm 2.

4.2   Convergence under the local tangential cone condition

Inspired by [58], we consider functions fulfilling the so-called tangential cone condition, which was introduced in [29] as a sufficient condition for convergence of the Landweber iteration for solving ill–posed nonlinear problems.

Definition 4.17. *A differentiable function* $f\colon D \to \mathbb{R}$ *fulfills the* local tangential cone condition (η-TCC) *on* $U \subset D$ *with constant* $0 < \eta < 1$, *if for all* $x, y \in U$ *it holds that*

$$|f(x) + \langle \nabla f(x), y - x \rangle - f(y)| \leq \eta |f(x) - f(y)|. \tag{31}$$

Under this condition, we are able to formulate a variant of Lemma 4.12 and derive corresponding convergence rates. Precisely, we will assume the following.

Assumption 3. *There exist a point* $\hat{x} \in S$ *and constants* $\eta \in]0, 1[$ *and* $r > 0$ *such that each function* f_i *fulfills* η-*TCC w.r.t.* η *on*

$$B_{r,\varphi}(\hat{x}) := \big\{ x \in C : D_\varphi^{x^*}(x, \hat{x}) \leq r \quad \text{for all } x^* \in \partial\varphi(x) \big\}.$$

Lemma 4.18. *Let Assumption 1 and Assumption 3 hold true and assume that* φ *is* σ-*strongly convex w.r.t. a norm* $\| \cdot \|$. *Let* $\hat{x} \in S$. *Then, the iterates of Algorithm 1 fulfill*

$$D_\varphi^{x_{k+1}^*}(x_{k+1}, \hat{x}) \leq D_\varphi^{x_k^*}(x_k, \hat{x}) - \tau \frac{\big(f_{i_k}(x_k)\big)^2}{\|\nabla f_{i_k}(x_k)\|_*^2},$$

if one of the following conditions is fulfilled:

(i) $t_k = t_{k,\sigma}$, $\eta < \frac{1}{2}$ *and* $\tau = \sigma\big(\frac{1}{2} - \eta\big)$,

(ii) $t_k = t_{k,\varphi}$, φ *is additionally* M-*smooth w.r.t.* $\| \cdot \|$, $\eta < \frac{\sigma}{2M}$ *and* $\tau = \sigma\big(\frac{1}{2} - \eta\frac{M}{\sigma}\big)$.

In particular, if $x_0 \in B_{r,\varphi}(\hat{x})$, *then in both cases we have that* $x_k \in B_{r,\varphi}(\hat{x})$ *for all* $k \in \mathbb{N}$.

Proof. For (i), by definition of $t_{k,\sigma}$ and η-TCC we have that

$$t_{k,\sigma}\big(f_{i_k}(x_k) + \langle f_{i_k}(x_k), \hat{x} - x_k \rangle\big) \leq \eta\sigma \frac{\big(f_{i_k}(x_k)\big)^2}{\|\nabla f_{i_k}(x_k)\|_*^2}.$$

The first convergence rate then follows by the steps in Lemma 4.12, replacing (29) by the above inequality. For (ii), using Lemma 4.10(ii) and the fact that f_{i_k} fulfills η-TCC, we estimate

$$t_{k,\varphi}\big(f_{i_k}(x_k) + \langle \nabla f_{i_k}(x_k), \hat{x} - x_k \rangle\big) \leq \eta M \frac{f_{i_k}(x)^2}{\|\nabla f_{i_k}(x)\|_*^2},$$

so that the assertion follows as in (i). $\qquad\square$

The condition on η in (i) is the classical condition for Landweber methods (see e.g. [29, Theorem 3.8]) and is also required in the work [58], which studies Algorithm 1 for $\varphi(x) = \frac{1}{2}\|x\|_2^2$ under η-TCC. The constant τ in (ii) can not be greater than τ in (i), as it holds that $\sigma \leq M$.

Theorem 4.19. *Let Assumption 1 hold and let φ be σ-strongly convex. Moreover, let Assumption 3 hold with some $\eta > 0$ and $\hat{x} \in S$ and let $x_0 \in B_{r,\varphi}(\hat{x})$.*

(i) *If $\eta < \frac{1}{2}$, then the iterates x_k of Algorithm 2 converge a.s. to a random variable whose image is contained in the solution set $S \cap B_{r,\varphi}(\hat{x})$ and it holds that*

$$\mathbb{E}\Big[\min_{l=1,\dots,k} \sum_{i=1}^{n} p_i\big(f_i(x_l)\big)^2\Big] \leq \frac{C \cdot D_\varphi^{x_0^*}(x_0, \hat{x})}{\sigma\big(\frac{1}{2} - \eta\big)k}.$$

(ii) *Let φ be additionally M-smooth and $\eta < \frac{\sigma}{2M}$. Assume that x_k are the iterates of Algorithm 1 and the condition $H_k \cap \operatorname{dom} \varphi \neq \emptyset$ is fulfilled for all k. Then the x_k converge a.s. to a random variable whose image is contained in the solution set $S \cap B_{r,\varphi}(\hat{x})$ and it holds that*

$$\mathbb{E}\Big[\min_{l=1,\dots,k} \sum_{i=1}^{n} p_i\big(f_i(x_l)\big)^2\Big] \leq \frac{C \cdot D_\varphi^{x_0^*}(x_0, \hat{x})}{\sigma\big(\frac{1}{2} - \eta\frac{M}{\sigma}\big)k}.$$

Proof. By Lemma 4.18, the x_k stay in $B_{r,\varphi}(\hat{x})$. The statements now follow as in Theorem 4.13 by invoking Lemma 4.18 instead of Lemma 4.12. $\qquad\square$

Finally, we can give a local linear convergence rate under the additional assumption that the Jacobian has full column rank. For $\varphi(x) = \frac{1}{2}\|x\|_2^2$, in part (i) of the theorem we recover the result from [58, Theorem 3.1] as a special case. In both Theorem 4.19 and Theorem 4.20, unfortunately we obtain a more pessimistic rate for Algorithm 1 compared to Algorithm 2, as the τ in (ii) is upper bounded by the τ in (i).

Theorem 4.20. *Let Assumption 1 hold true and let φ be σ-strongly convex and M-smooth. Let Assumption 3 hold with some $\eta > 0$ and $\hat{x} \in S$ and let $x_0 \in B_{r,\varphi}(\hat{x})$. Moreover, assume that the Jacobian $Df(x)$ has full column rank for all $x \in B_{r,\varphi}(\hat{x})$ and $p_{\min} = \min_{i=1,\dots,n} p_i > 0$. We set*

$$\kappa_{\min} := \min_{x \in B_{r,\varphi}(\hat{x})} \min_{\|y\|_2 = 1} \frac{\|Df(x)\|_F}{\|Df(x)y\|_2}.$$

(i) *If $\eta < \frac{1}{2}$, then the iterates x_k of Algorithm 2 fulfill that*

$$\frac{\sigma}{2}\mathbb{E}\big[\|x_k - \hat{x}\|_2^2\big] \leq \mathbb{E}\big[D_\varphi(x_k, \hat{x})\big] \leq \Big(1 - \frac{\sigma\big(\frac{1}{2} - \eta\big)p_{\min}}{(1 + \eta)^2 \kappa_{\min}^2}\Big)^k \mathbb{E}\big[D_\varphi(x_0, \hat{x})\big]. \tag{32}$$

(ii) *Let $\eta < \frac{\sigma}{2M}$. Assume that x_k are the iterates of Algorithm 1 and the condition $H_k \cap \operatorname{dom} \varphi \neq \emptyset$ is fulfilled for all k. Then it holds that*

$$\frac{\sigma}{2}\mathbb{E}\big[\|x_k - \hat{x}\|_2^2\big] \leq \mathbb{E}\big[D_\varphi(x_k, \hat{x})\big] \leq \Big(1 - \frac{\sigma\big(\frac{1}{2} - \eta\frac{M}{\sigma}\big)p_{\min}}{M(1 + \eta)^2 \kappa_{\min}^2}\Big)^k \mathbb{E}\big[D_\varphi(x_0, \hat{x})\big]. \tag{33}$$

For the proof, as in [58] we use the following auxiliary lemma.

Lemma 4.21. *Let $a_1, ..., a_n \geq 0$ and $b_1, ..., b_n > 0$. Then it holds that*

$$\sum_{i=1}^n \frac{a_i}{b_i} \geq \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i}.$$

Proof. Since $a_i, b_i \geq 0$ it holds that

$$\Big(\sum_{i=1}^n b_i\Big)\Big(\sum_{j=1}^n \frac{a_j}{b_j}\Big) = \sum_{i,j=1}^n b_i \frac{a_j}{b_j} \geq \sum_{i=1}^n b_i \frac{a_i}{b_i} = \sum_{i=1}^n a_i.$$

$\square$

Proof of Theorem 4.20. By Assumption 3 and the fact that $f(\hat{x}) = 0$, we can estimate

$$|\langle \nabla f_{i_k}(x_k), \hat{x} - x_k \rangle| \leq |f_{i_k}(x_k) + \langle \nabla f_{i_k}(x_k), \hat{x} - x_k \rangle - f_{i_k}(\hat{x})| + |f_{i_k}(x_k) - f_{i_k}(\hat{x})|$$
$$\leq (1+\eta)|f_{i_k}(x_k) - f_{i_k}(\hat{x})| = (1+\eta)|f_{i_k}(x_k)|.$$

In all cases of the assumption, inserting the above estimate we respectively conclude that

$$D_\varphi(x_{k+1}, \hat{x}) \leq D_\varphi(x_k, \hat{x}) - \frac{\tau}{(1+\eta)^2} \cdot \frac{|\langle \nabla f_{i_k}(x_k), \hat{x} - x_k \rangle|^2}{\|\nabla f_i(x_k)\|_2^2}.$$

Taking expectation and using Lemma 4.21 as well as the definition of $\kappa_{\min}$, we conclude that

$$\mathbb{E}\big[D_\varphi(x_{k+1}, \hat{x})\big] \leq \mathbb{E}\big[D_\varphi(x_k, \hat{x})\big] - \frac{\tau}{(1+\eta)^2} \cdot \mathbb{E}\Big[\sum_{i=1}^n p_i \frac{|\langle \nabla f_i(x_k), \hat{x} - x_k \rangle|^2}{\|\nabla f_i(x_k)\|_2^2}\Big]$$
$$\leq \mathbb{E}\big[D_\varphi(x_k, \hat{x})\big] - \frac{\tau p_{\min}}{(1+\eta)^2} \cdot \mathbb{E}\Big[\frac{\|Df(x_k)(\hat{x} - x_k)\|_2^2}{\|Df(x_k)\|_F^2}\Big]$$
$$\leq \mathbb{E}\big[D_\varphi(x_k, \hat{x})\big] - \mathbb{E}\Big[\frac{\tau p_{\min}}{(1+\eta)^2 \kappa_{\min}^2} \cdot \|x_k - \hat{x}\|_2^2\Big]$$
$$\leq \Big(1 - \frac{2\tau p_{\min}}{(1+\eta)^2 M \kappa_{\min}^2}\Big)\mathbb{E}\big[D_\varphi(x_k, \hat{x})\big].$$

$\square$

5 Numerical experiments

In this section, we evaluate the performance of the NBK method. In the first experiment we used NBK to find sparse solutions with the nonsmooth DGF $\varphi(x) = \frac{1}{2}\|x\|_2^2 + \lambda\|x\|_1$ for unconstrained quadratic equations, that is, with $C = \mathbb{R}^d$. Next, we employed the negative entropy DGF over the probability

simplex $C = \Delta^{d-1}$ to solve simplex-constrained linear equations as well as the *left-stochastic decomposition problem*, a quadratic problem over a product of probability simplices with applications in clustering [2]. All the methods were implemented in MATLAB on a macbook with 1,2 GHz Quad-Core Intel Core i7 processor and 16 GB memory. Code is available at `https://github.com/MaxiWk/Bregman-Kaczmarz`.

5.1   Sparse solutions of quadratic equations

As the first example, we considered multinomial quadratic equations

$$f_i(x) = \frac{1}{2}\langle x, A^{(i)}x \rangle + \langle b^{(i)}, x \rangle + c^{(i)} = 0$$

with $A^{(i)} \in \mathbb{R}^{d \times d}$, $b^{(i)} \in \mathbb{R}^d$, $c^{(i)} \in \mathbb{R}$ and $i = 1, ..., n$. We investigated if Algorithm 1 (NBK method) and Algorithm 2 (rNBK method) are capable of finding a sparse solution $\hat{x} \in \mathbb{R}^d$ by using the DGF $\varphi(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$ and tested both methods against the euclidean nonlinear Kaczmarz method (NK). As it holds dom $\varphi = \mathbb{R}^d$, it is always possible to choose the step size $t_{k,\varphi}$ from (10) in the NBK method. Moreover, the step size can be computed exactly by a sorting procedure, as φ^* is a continuous piecewise quadratic function, see Example 3.2. In order to guarantee existence of a sparse solution, we chose a sparse vector $\hat{x} \in \mathbb{R}^d$, sampled the data $A^{(i)}, b^{(i)}$ randomly with entries from the standard normal distribution and set

$$c^{(i)} = -\left(\frac{1}{2}\langle x, A^{(i)}x \rangle + \langle b^{(i)}, x \rangle\right).$$

In all examples, the nonzero part of $\hat{x}$ and the initial subgradient x_0^* were sampled from the standard normal distribution. The initial vector x_0 was computed by $x_0 = \nabla\varphi^*(x_0^*) = S_\lambda(x_0^*)$.

From the updates it is evident that computational cost per iteration is cheapest for the NK method, slightly more expensive for the rNBK method and most expensive for the NBK method. To examine the case $d < n$, we chose $A^{(i)} \sim \mathcal{N}(0,1)^{500 \times 500}$ for $i = 1, ..., 1000$, $\hat{x}$ with 25 nonzero entries and $\lambda = 10$ and performed 20 random repeats. Figure 1 shows that the NBK method clearly outperforms the other two methods in this situation, even despite the higher cost per iteration.

Figure 2 illustrates that in the case $d > n$, both the NBK method and the rNBK method can fail to converge or converge very slowly.

5.2   Linear systems on the probability simplex

We tested our method on linear systems constrained to the probability simplex

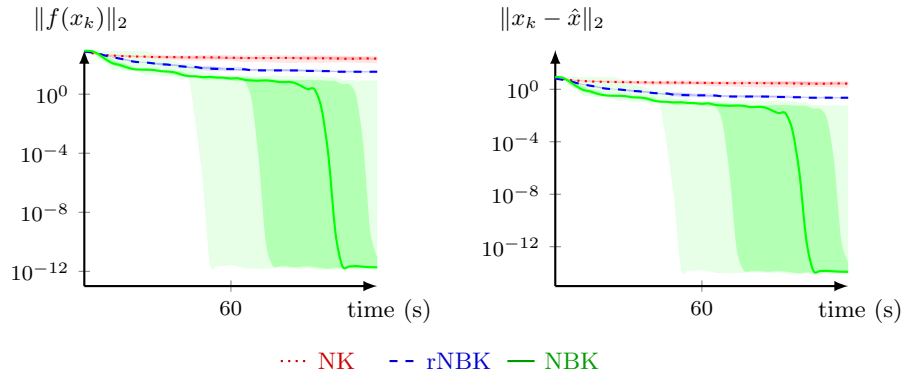$$\text{find } x \in \Delta^{d-1} : \qquad Ax = b. \tag{34}$$

28

Figure 1: Experiment with quadratic equations, $(n, d) = (1000, 500)$, $\hat{x}$ with 50 nonzero entries, 20 random repeats. Left: plot of residual $\|f(x_k)\|_2$, right: plot of distance to solution $\hat{x}$, both over computation time. Thick line shows median over all trials, light area is between min and max, darker area indicates 25th and 75th quantile.

That is, in problem (1) we chose $f_i = \langle a_i, x \rangle - b_i$ with $D = \mathbb{R}^d$ and viewed $C = \Delta^{d-1}$ as the additional constraint. For Algorithm 1, we used the simplex-restricted negative entropy function from Example 3.3, i.e. we set

$$\varphi(x) = \begin{cases} \sum_{i=1}^{d} x_i \log(x_i), & x \in \Delta^{d-1}, \\ +\infty, & \text{otherwise.} \end{cases}$$

We know from Example 3.3 that φ is 1-strongly convex w.r.t. the 1-norm $\|\cdot\|_1$. Therefore, as the second method we considered the rNBK iteration given by Algorithm 2 with $\sigma = 1$ and $\|\cdot\|_* = \|\cdot\|_\infty$. As a benchmark, we considered a POCS (orthogonal projection) method which computes an orthogonal projection onto a row equation, followed by an orthogonal projection onto the probability simplex, see Algorithm 3 listed below. We note that in [36, Theorem 3.3] it has been proved that the distance of the iterates of the POCS method and the NBK method to the set of solutions on Δ_+^{d-1} decays with an expected linear rate, if there exists a solution in Δ_+^{d-1}. Theorem 4.13 shows at least a.s. convergence of the iterates towards a solution for all three methods.

We note that it holds $\nabla f_{i_k}(x) = a_{i_k}$ for all x and $\beta_k = b_{i_k}$ in the NBK method. If problem (34) has a solution, then condition (9) is fulfilled in each step of the NBK method, so the method takes always the step size $t_k = t_{k,\varphi}$ from the exact Bregman projection. For the projection onto the simplex in Algorithm 3, we used the pseudocode from [59], see also [12, 24, 26].

In our experiments we noticed that in large dimensions, such as $d \geq 100$, solving the Bregman projection (10) up to a tolerance of $\epsilon = 10^{-9}$ takes less than half as much computation time as the simplex projection. As these two are the dominant operations in these methods, the NBK updates are computationally cheaper than the NK updates in the high dimensional setting. However, the examples will show that convergence quality of the methods depends on the
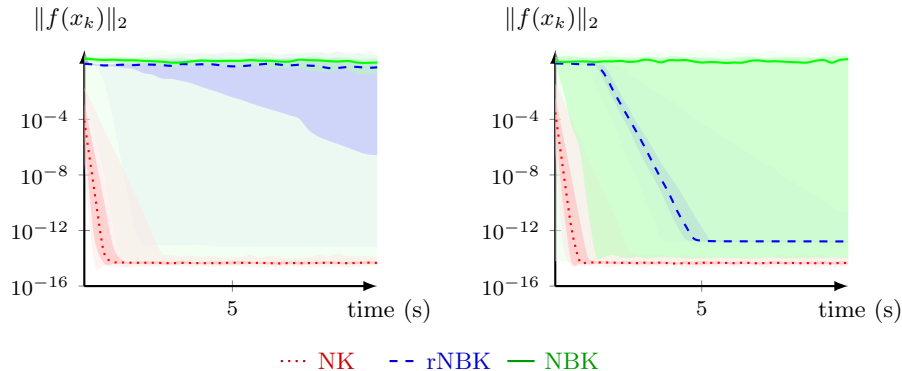
Figure 2: Experiment with quadratic equations, $(n, d) = (50, 100)$, $\hat{x}$ with 5 nonzero entries, 50 random repeats, plot of residual $\|f(x_k)\|_2$ against computation time. Left: $\lambda = 2$, right: $\lambda = 5$. Thick line shows median over all trials, light area is between min and max, darker area indicates 25th and 75th quantile.

Algorithm 3 Alternating euclidean projections (POCS method) for (34)

1: Input: probabilities $p_i > 0$ for $i = 1, ..., n$
2: Initialization: $x_0 \in \Delta^{d-1}$
3: **for** $k = 0, 1, ...$ **do**
4:     choose $i_k \in \{1, ..., n\}$ according to the probabilities $p_1, ..., p_n$
5:     project $y_{k+1} = \Pi_{H(a_i, b_i)}(x_k) = x_k - \frac{\langle a_{i_k}, x_k \rangle - b_{i_k}}{\|a_{i_k}\|_2^2} a_{i_k}$
6:     project $x_{k+1} = \Pi_{\Delta^{d-1}}(y_{k+1})$

---

distribution of the entries of A. All methods were observed to converge linearly.

In the following experiments, we took different choices of A and set the right-hand side to $b = A\hat{x}$ with a point $\hat{x}$ drawn from the uniform distribution on the probability simplex Δ^{d-1}. All methods were initialized with the center point $x_0 = (\frac{1}{d}, ..., \frac{1}{d})$.

For our first experiment, we chose standard normal entries $A \sim \mathcal{N}(0, 1)^{n \times d}$ with $(n, d) = (500, 200)$ and $(n, d) = (200, 500)$. Figure 3 shows that in this setting, the POCS method achieves much faster convergence in the overdetermined case $(n, d) = (500, 200)$ than the NBK method, whereas both methods perform roughly the same in the underdetermined case $(d, n) = (200, 500)$. The rNBK method is considerably slower than the other two methods, which shows that the computation of the $t_{k,\varphi}$ step size for NBK pays off.

In our second experiment, we built up the matrix from uniformly distributed entries $A \sim \mathcal{U}([0, 1])^{n \times d}$ and $A \sim \mathcal{U}([0.9, 1])^{n \times d}$ with $(n, d) = (200, 500)$. The results are summarized in Figure 4. For the Kaczmarz method it has been observed in practice that so called 'redundant' rows of the matrix A deteriorate the convergence of the method [31]. This effect can also occur with the POCS method, as it also relies on euclidean projections. Remarkably, we can see that this is not the case for the NBK method and it clearly outperforms the POCS

method and the rNBK method. This in particular shows that the multiplicative update used in both the rNBK method and the NBK method is not enough to overcome the difficulty of redundancy- to achieve fast convergence, it must be combined with the appropriate step size which is used by the proposed NBK method.

Finally, we illustrate the effect of the accuracy ϵ in step size computation for the NBK method. We chose $A \sim U([0,1])^{n \times d}$ and $\epsilon = 10^{-9}$ and compared with the larger tolerance $\epsilon = 10^{-5}$. Figure 5 shows that, with $\epsilon = 10^{-5}$, the residual plateaus at a certain threshold. In contrast with $\epsilon = 10^{-9}$, the residual does not plateau, and despite the more costly computation of the step size, the NBK method is still competitive with respect to time. Hence, for the problem of linear equations over the probability simplex we recommend to solve the step size problem up to high precision.

Figure 3: Experiment with linear equations on the probability simplex, plot of relative residuals averaged over 50 random examples against iterations (k) and computation time. Left column: $A \sim \mathcal{N}(0,1)^{500 \times 200}$, right column: $A \sim \mathcal{N}(0,1)^{200 \times 500}$. Thick line shows median over all trials, light area is between min and max, darker area indicates 25th and 75th quantile.
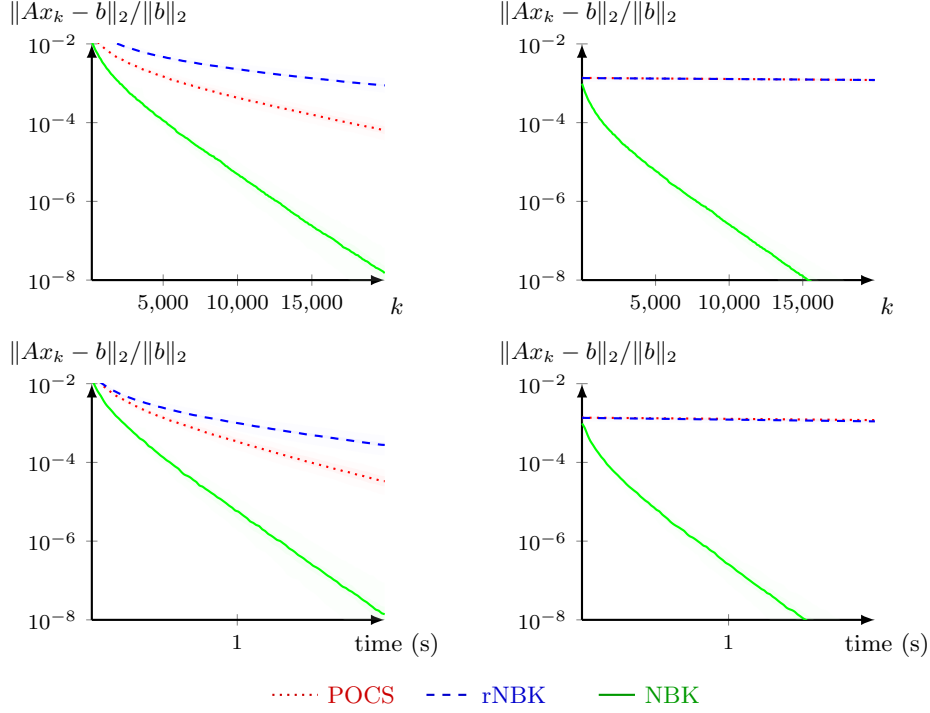
Figure 4: Experiment with linear equations on the probability simplex, plot of relative residuals averaged over 50 random examples against iterations (k) and computation time. Left column: $A \sim \mathcal{U}([0,1])^{200 \times 500}$, right column: $A \sim \mathcal{U}([0.9,1])^{200 \times 500}$. Thick line shows median over all trials, light area is between min and max, darker area indicates 25th and 75th quantile.

5.3   Left stochastic decomposition

The *left stochastic decomposition* (LSD) problem can be formulated as follows:

$$\text{find } X \in L^{r \times m} : \qquad X^T X = A, \tag{35}$$

where

$$L^{r,m} := \left\{ P \in \mathbb{R}_{\geq 0}^{r \times m} : P^T \mathbb{1}_r = \mathbb{1}_m \right\}$$

is the set of left stochastic matrices and $A \in \mathbb{R}^{r \times m}$ is a given nonnegative matrix. The problem is equivalent to the so-called *soft-K-means* problem and hence has applications in clustering [2]. We can view (35) as an instance of problem (1) with component equations

$$f_{i,j}(X) = \langle X_{:,i}, X_{:,j} \rangle - A_{i,j} = 0 \qquad \text{for } i = 1, ..., r, \ j = 1, ..., m$$

and $C = L^{r \times m} \cong \left(\Delta^{r-1} \right)^m$, where $X_{:,i}$ denotes the ith column of X. For Algorithm 1 we chose the DGF from Example 3.4 with the simplex-restricted negative entropy $\varphi_i = \varphi$ from Example 3.3. Since $f_{i,j}$ depends on at most
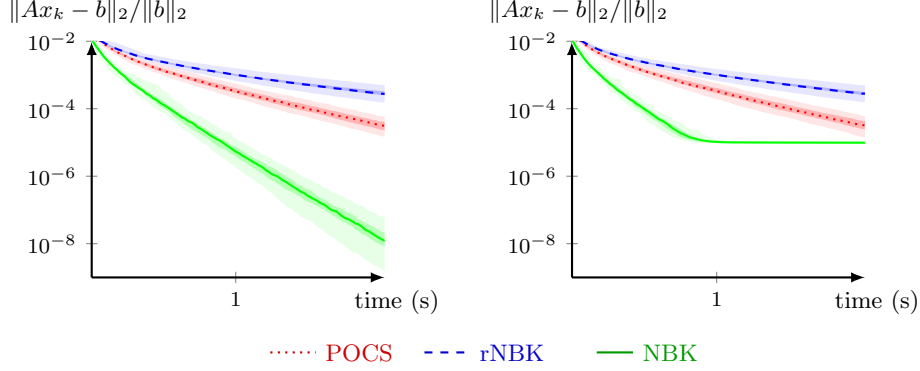
32

Figure 5: Experiment with linear equations on the probability simplex, plot of relative residuals averaged over 50 random examples against computation time. In both examples, $A \sim \mathcal{U}([0,1])^{200 \times 500}$. Left: $\epsilon = 10^{-9}$, right: $\epsilon = 10^{-5}$ in NBK method. Thick line shows median over all trials, light area is between min and max, darker area indicates 25th and 75th quantile.

two columns of X, Algorithm 1 acts on Δ^{r-1} or $\Delta^{r-1} \times \Delta^{r-1}$ in each step. Therefore, we applied the steps from Example 3.3 in the first case, and from Example 3.5 in the second case.

We compared the performance of Algorithm 1 and Algorithm 2 to a projected nonlinear Kaczmarz method given by Algorithm 4. Here, by $X_{k,:,i}$ we refer to the ith column of the kth iterate matrix. In all examples, we set $A = \hat{X}^T \hat{X}$, where the columns of $\hat{X}$ were sampled according to the uniform distribution on Δ^{r-1}.

Algorithm 4 Projected nonlinear Kaczmarz method (PNK) for (35)

1: Input: $\sigma > 0$ and probabilities p_{ij} for $i = 1, ..., r$ and $j = 1, ..., m$
2: Initialization: $X_0 \in L^{r \times m}$
3: **for** $k = 0, 1, ...$ **do**
4:     choose $i_k \in \{1, ..., r\}$ and $j_k \in \{1, ..., m\}$ according to $p_{1r}, ..., p_{rm}$
5:     set $\beta_k = \langle \nabla f_{i_k}(x_k), x_k \rangle - f_{i_k}(x_k) = \langle X_{k,:,i_k}, X_{k,:,j_k} \rangle + A_{i_k, j_k}$
6:     **if** $i_k = j_k$ **then**
7:         project $Y_{k+1,:,i_k} = \Pi_{H(\alpha_k, \beta_k)} X_{k,:,i_k}$ with $\alpha_k = 2 X_{k,:,i_k}$
8:         project $X_{k+1,:,i_k} = \Pi_{\Delta^{m-1}}(Y_{k+1,:,i_k})$
9:     **if** $i_k \neq j_k$ **then**
10:         set $t_k = \frac{\langle X_{k,:,i_k}, X_{k,:,j_k} \rangle - A_{i_k, j_k}}{\|X_{k,:,i_k}\|_2^2 + \|X_{k,:,j_k}\|_2^2}$
11:         set $Y_{k+1,:,i_k} = Y_{k,:,i_k} - t_k Y_{k,:,j_k}$
12:         set $Y_{k+1,:,j_k} = Y_{k,:,j_k} - t_k Y_{k,:,i_k}$
13:         project $X_{k+1,:,i_k} = \Pi_{\Delta^{r-1}}(Y_{k+1,:,i_k})$
14:         project $X_{k+1,:,j_k} = \Pi_{\Delta^{r-1}}(Y_{k+1,:,j_k})$

We observed that Algorithm 1 (NBK method) gives the fastest convergence,

if r is not much smaller than m, see Figure 6 and Figure 7. In both experiments, we noticed that condition (9) was actually fulfilled in each step, but checking did not show a notable difference in performance. The most interesting setting for clustering is that r is very small and m is large, as r is the number of clusters [2]. However, it appears unclear if the NBK or the PNK method is a better choice for this problem size, as Figure 8 shows. In this experiment, condition (9) was not always fulfilled in the NBK method and we needed to employ the globalized Newton method together with an additional condition to the step size approximation, see Appendix for details. Finally, we can again see that Algorithm 2 is clearly outperformed by the other two methods in all experiments.
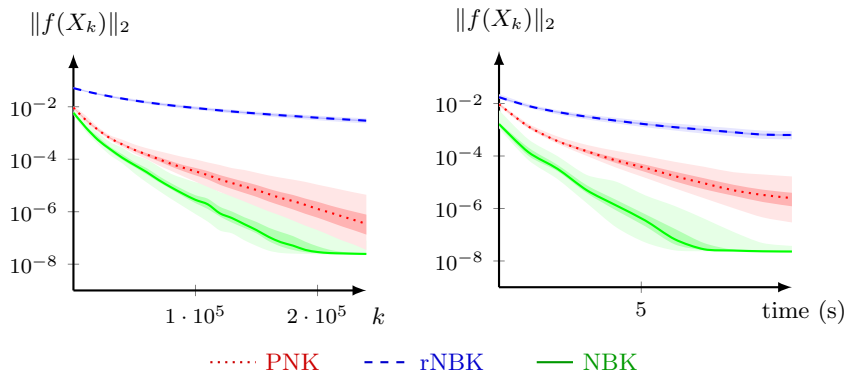

Figure 6: Experiment 'Left stochastic decomposition problem' with $r = 100, m = 50$. Residuals $\|f(X^{(k)})\|_2$ averaged over 50 random examples against outer iterations k (left) and computation time (right). Thick line shows median over all trials, light area is between min and max, darker area indicates 25th and 75th quantile.

6   Conclusions and further research

We provided a general Bregman projection method for solving nonlinear equations, where each iteration needs only to sample one equation to make progress towards the solution. As such, the cost of one iteration scales independently of the number of equations. Our method is also a generalization of the nonlinear Kaczmarz method which allows for additional simple constraints or sparsity inducing regularizers. We provide two global convergence theorems under different settings and find a number of relevant experimental settings where instantiations of our method are efficient.

Convergence for non-strongly convex distance generating functions φ, as well as a suitable scope of σ in this setting, has so far not been explored.

Our work also opens up the possibility of incorporating more structure into SGD type methods in the interpolation setting as has been done in [33] for the linear case. In this setting each $f_i(x)$ is a positive loss function over the ith data
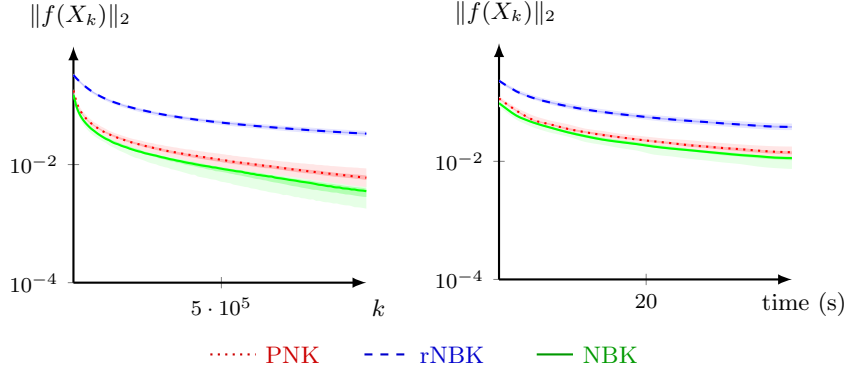
34

Figure 7: Experiment 'Left stochastic decomposition problem' with $r = 50, m = 100$, plot of residuals $\|f(X^{(k)})\|_2$ averaged over 50 random examples against iterations (left) and computation time (right). Thick line shows median over all trials, light area is between min and max, darker area indicates 25th and 75th quantile.

point. If we knew in addition that some of the coordinates of x are meant to be positive, or that x is a discrete probability measure, then our nonlinear Bregman projection methods applied to the interpolation equations would provide new adaptive step sizes for stochastic mirror descent. Further venues for exploring would be to relax the interpolation equations, say into inequalities [27], and applying an analogous Bregman projections to incorporate more structure. We will leave this to future work.

A   Newton's method for line search problem (10)

We compute the Newton update for problem (10) for general φ with C^2-smooth conjugate φ^*. The function g_{i_k, x_k^*} from (18) has first derivative

$$g'_{i_k, x_k^*}(t) = \left\langle \nabla \varphi^*(x_k^* - t\nabla f_{i_k}(x_k)), -\nabla f_{i_k}(x_k) \right\rangle + \beta_k$$
$$= \left\langle x_k - \nabla \varphi^*(x_k^* - t\nabla f_{i_k}(x_k)), \ \nabla f_{i_k}(x_k) \right\rangle - f_{i_k}(x_k)$$

and second derivative

$$g''_{i_k, x_k^*}(t) = \left\langle \nabla^2 \varphi^*(x_k^* - t\nabla f_{i_k}(x_k))\nabla f_{i_k}(x_k), \ \nabla f_{i_k}(x_k) \right\rangle \geq 0.$$

If it holds $g''_{f_{i_k}, x_k^*}(t) > 0$, Newton's method for (10) reads

$$t_{k,l+1} = t_{k,l} - \frac{g'_{i_k, x_k^*}(t_{k,l})}{g''_{i_k, x_k^*}(t_{k,l})}.$$

As an initial value we use the step size $t_{k,0} := \frac{f_{i_k}(x_k)}{\|\nabla f_{i_k}(x_k)\|_2^2}$ from the ℓ_2-projection of x_k onto H_k. We propose to stop the method if $|g'_{i_k, x_k^*}(t_{k,l})| < \epsilon$. Typical values we used for our numerical examples were $\epsilon \in \{10^{-5}, 10^{-6}, 10^{-9}, 10^{-15}\}$.
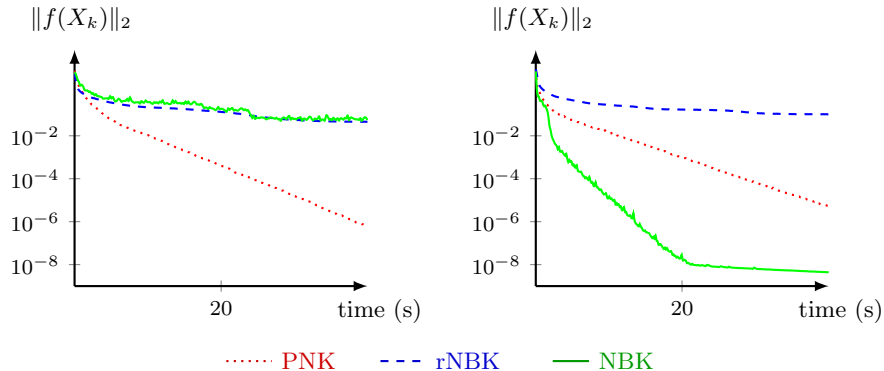
Figure 8: Experiment 'Left stochastic decomposition problem' with $r = 3, m = 100$, residuals $\|f(X_k)\|_2$ against computation time. Left and right: Two random examples with different convergence behavior. Thick line shows median over all trials, light area is between min and max, darker area indicates 25th and 75th quantile.

It may happen that problem (10) is ill-conditioned, in which case the Newton iterates $t_{k,l}$ may diverge quickly to $\pm\infty$ or alternate between two values. We have observed this can e.g. happen for the problem on left stochastic decomposition in Subsection 5.3, if the number m of rows of the matrix X in the problem is small.

In case that the Newton method diverges, we used the recently proposed globalized Newton method from [42], which reads

$$ t_{k,l+1} = t_{k,l} - \frac{g'_{i_k, x_k^*}(t_{k,l})}{H \cdot \sqrt{|g'_{i_k, x_k^*}(t_{k,l})|} + g''_{i_k, x_k^*}(t_{k,l})} $$

with a fixed constant $H > 0$. Also here, we stop if $|g'_{i_k, x_k^*}(t_{k,l})| < \epsilon$. Convergence of the $t_{k,l}$ for $l \to \infty$ is guaranteed, if φ^* is strongly convex, i.e. if φ is everywhere finite with Lipschitz continuous gradient and the values $g_{i_k, x_k^*}(t_{k,l})$ are guaranteed to converge to the minimum value if φ^* has Lipschitz continuous Hessian [42]. We have also observed good convergence for the negative entropy function on $\mathbb{R}^d_{\geq 0}$ with this method when Newton's method is unstable. For problems constrained to the probability simplex Δ^{d-1}, the globalized Newton method converged more slowly than the vanilla Newton method. For the problem in subsection 5.3 with $(r, m) = (3, 100)$ we chose $H = 0.1$. In addition, we performed a relaxed Bregman projection (line 10 of Algorithm 1) with step size (11) if $|t_{k,l}| > 100$.

36

Declarations

Competing interests

The authors have no competing interests to declare that are relevant to the content of this article.

Data availability

We do not analyze or generate any datasets, because our work proceeds within a theoretical and mathematical approach. However, the code that generates the figures in this article can be found at `https://github.com/MaxiWk/Bregman-Kaczmarz`.

References

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

[2] R. Arora, M. R. Gupta, A. Kapila, and M. Fazel. Similarity-based clustering by left-stochastic matrix factorization. *The Journal of Machine Learning Research*, 14(1):1715–1746, 2013.

[3] N. Azizan, S. Lale, and B. Hassibi. Stochastic mirror descent on overparameterized nonlinear models. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[4] H. H. Bauschke, J. M. Borwein, and P. L. Combettes. Bregman monotone optimization algorithms. *SIAM Journal on control and optimization*, 42(2):596–636, 2003.

[5] H. H. Bauschke, J. M. Borwein, et al. Legendre functions and the method of random Bregman projections. *Journal of Convex Analysis*, 4(1):27–67, 1997.

[6] H. H. Bauschke and P. L. Combettes. Iterating Bregman retractions. *SIAM Journal on Optimization*, 13(4):1159–1173, 2003.

[7] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011.

[8] A. Beck. *First-order methods in optimization*. SIAM, 2017.

[9] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

[10] R. I. Boţ and T. Hein. Iterative regularization with a general penalty term—theory and application to L1 and TV regularization. *Inverse Problems*, 28(10):104010, 2012.

[11] L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

[12] P. Brucker. An O(n) algorithm for quadratic knapsack problems. *Operations Research Letters*, 3(3):163–166, 1984.

[13] D. Butnariu, A. N. Iusem, and C. Zalinescu. On uniform convexity, total convexity and convergence of the proximal point and outer Bregman projection algorithm in Banach spaces. *Journal of Convex Analysis*, 10(1):35–62, 2003.

[14] D. Butnariu and E. Resmerita. The outer Bregman projection method for stochastic feasibility problems in Banach spaces. In *Studies in Computational Mathematics*, volume 8, pages 69–86. Elsevier, 2001.

[15] D. Butnariu and E. Resmerita. Bregman distances, totally convex functions, and a method for solving operator equations in Banach spaces. In *Abstract and Applied Analysis*, volume 2006. Hindawi, 2006.

[16] Y. Censor, T. Elfving, and G. Herman. Averaging strings of sequential iterations for convex feasibility problems. In *Studies in Computational Mathematics*, volume 8, pages 101–113. Elsevier, 2001.

[17] Y. Censor and A. Lent. An iterative row-action method for interval convex programming. *Journal of Optimization Theory and Applications*, 34(3):321–353, 1981.

[18] Y. Censor and S. Reich. Iterations of paracontractions and firmaly nonexpansive operators with applications to feasibility and optimization. *Optimization*, 37(4):323–339, 1996.

[19] A. E. Cetin. Reconstruction of signals from Fourier transform samples. *Signal Processing*, 16(2):129–148, 1989.

[20] A. E. Cetin. An iterative algorithm for signal reconstruction from bispectrum. *IEEE Transactions on Signal Processing*, 39(12):2621–2628, 1991.

[21] I. S. Dhillon and J. A. Tropp. Matrix nearness problems with Bregman divergences. *SIAM Journal on Matrix Analysis and Applications*, 29(4):1120–1146, 2008.

[22] N. Doikov and Y. Nesterov. Gradient regularization of Newton method with Bregman distances. *arXiv preprint arXiv:2112.02952*, 2021.

[23] R. D'Orazio, N. Loizou, I. Laradji, and I. Mitliagkas. Stochastic mirror descent: Convergence analysis and adaptive variants via the mirror stochastic Polyak stepsize. arXiv preprint arXiv:2110.15412, 2021.

[24] 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 25th international conference on Machine learning*, pages 272–279, 2008.

[25] A. A. Fedotov, P. Harremoës, and F. Topsoe. Refinements of Pinsker's inequality. *IEEE Transactions on Information Theory*, 49(6):1491–1498, 2003.

[26] E. M. Gafni and D. P. Bertsekas. Two-metric projection methods for constrained optimization. *SIAM Journal on Control and Optimization*, 22(6):936–964, 1984.

[27] R. M. Gower, M. Blondel, N. Gazagnadou, and F. Pedregosa. Cutting some slack for SGD with adaptive Polyak stepsizes. *arXiv:2202.12328*, 2022.

[28] R. Gu, B. Han, S. Tong, and Y. Chen. An accelerated Kaczmarz type method for nonlinear inverse problems in Banach spaces with uniformly convex penalty. *Journal of Computational and Applied Mathematics*, 385:113211, 2021.

[29] M. Hanke, A. Neubauer, and O. Scherzer. A convergence analysis of the Landweber iteration for nonlinear ill-posed problems. *Numerische Mathematik*, 72(1):21–37, 1995.

[30] N. A. Iusem and V. M. Solodov. Newton-type methods with generalized distances for constrained optimization. *Optimization*, 41(3):257–278, 1997.

[31] B. Jarman, Y. Yaniv, and D. Needell. Online signal recovery via heavy ball Kaczmarz. *arXiv preprint arXiv:2211.06391*, 2022.

[32] Q. Jin. Landweber-Kaczmarz method in Banach spaces with inexact inner solvers. *Inverse Problems*, 32(10):104005, 2016.

[33] Q. Jin, X. Lu, and L. Zhang. Stochastic mirror descent method for linear ill-posed problems in banach spaces. *Inverse Problems*, 39(6):065010, 2023.

[34] Q. Jin and W. Wang. Landweber iteration of Kaczmarz type with general non-smooth convex penalty functionals. *Inverse Problems*, 29(8):085011, 2013.

[35] S. Kaczmarz. Angenäherte Auflösung von Systemen linearer Gleichungen. *Bull. Internat. Acad. Polon. Sci. Lettres A*, pages 355–357, 1937.

[36] V. Kostic and S. Salzo. The method of Bregman projections in deterministic and stochastic convex feasibility problems. *arXiv preprint arXiv:2101.01704*, 2021.

[37] T. Lin, N. Ho, and M. Jordan. On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms. In *International Conference on Machine Learning*, pages 3982–3991. PMLR, 2019.

[38] N. Loizou, S. Vaswani, I. H. Laradji, and S. Lacoste-Julien. Stochastic Polyak step-size for SGD: An adaptive learning rate for fast convergence. In *International Conference on Artificial Intelligence and Statistics*, pages 1306–1314. PMLR, 2021.

[39] D. A. Lorenz, F. Schöpfer, and S. Wenger. The linearized Bregman method via split feasibility problems: Analysis and generalizations. *SIAM Journal on Imaging Sciences*, 7(2):1237–1262, 2014.

[40] S. Ma, R. Bassily, and M. Belkin. The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. In *International Conference on Machine Learning*, pages 3325–3334. PMLR, 2018.

[41] P. Maaß and R. Strehlow. An iterative regularization method for nonlinear problems based on Bregman projections. *Inverse Problems*, 32(11):115013, 2016.

[42] K. Mishchenko. Regularized Newton method with global $O(1/k^2)$ convergence. arXiv preprint arXiv:2112.02089, 2021.

[43] A. Nedić. Random algorithms for convex minimization problems. *Mathematical Programming*, 129(2):225–253, 2011.

[44] A. S. Nemirovskij and D. B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience, 1983.

[45] Y. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[46] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.

[47] M. S. Pinsker. *Information and information stability of random variables and processes (in Russian)*. Holden-Day, 1964.

[48] B. Polyak and A. Tremba. New versions of Newton method: step-size choice, convergence domain and under-determined equations. *Optimization Methods and Software*, 35(6):1272–1303, 2020.

[49] B. Polyak and A. Tremba. Sparse solutions of optimal control via Newton method for under-determined systems. *Journal of Global Optimization*, 76(3):613–623, 2020.

[50] D. Reem, S. Reich, and A. De Pierro. Re-examination of Bregman functions and new properties of their divergences. *Optimization*, 68(1):279–348, 2019.

[51] H. Robbins and D. Siegmund. A convergence theorem for non negative almost supermartingales and some applications. In *Optimizing methods in statistics*, pages 233–257. Elsevier, 1971.

[52] R. T. Rockafellar. *Convex Analysis*, volume 36. Princeton University Press, 1970.

[53] F. Schöpfer and D. A. Lorenz. Linear convergence of the randomized sparse Kaczmarz method. *Mathematical Programming*, 173(1):509–536, 2019.

[54] F. Schöpfer, D. A. Lorenz, L. Tondji, and M. Winkler. Extended randomized Kaczmarz method for sparse least squares and impulsive noise problems. *Linear Algebra and its Applications*, 652:132–154, 2022.

[55] F. Schöpfer, A. K. Louis, and T. Schuster. Nonlinear iterative methods for linear ill-posed problems in Banach spaces. *Inverse problems*, 22(1):311, 2006.

[56] S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(1), 2013.

[57] L. Tondji and D. A. Lorenz. Faster randomized block sparse Kaczmarz by averaging. *Numerical Algorithms*, pages 1–35, 2022.

[58] Q. Wang, W. Li, W. Bao, and X. Gao. Nonlinear Kaczmarz algorithms and their convergence. *Journal of Computational and Applied Mathematics*, 399:113720, 2022.

[59] W. Wang and M. A. Carreira-Perpinán. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *arXiv preprint arXiv:1309.1541*, 2013.

[60] J.-K. You, H.-C. Cheng, and Y.-H. Li. Minimizing quantum Rényi divergences via mirror descent with Polyak step size. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 252–257. IEEE, 2022.

[61] R. Yuan, A. Lazaric, and R. M. Gower. Sketched Newton-Raphson. *SIAM Journal on Optimization*, 32(3):1555–1583, 2022.

[62] C. Zalinescu. *Convex analysis in general vector spaces*. World scientific, 2002.

[63] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[64] Z. Zhou, P. Mertikopoulos, N. Bambos, S. Boyd, and P. W. Glynn. Stochastic mirror descent in variationally coherent optimization problems. *Advances in Neural Information Processing Systems*, 30:7040–7049, 2017.