IMA Journal of Numerical Analysis (2020) Page 1 of 30
doi:10.1093/imanum/drnxxx

Randomized sketch descent methods for non-separable linearly constrained optimization

ION NECOARA†
Automatic Control and Systems Engineering Department, University Politehnica Bucharest, 060042 Bucharest, Romania

AND

MARTIN TAKÁČ‡
Industrial and Systems Engineering Department, Lehigh University, Bethlehem, PA 18015, USA

[Received on 30 January 2019]

In this paper we consider large-scale smooth optimization problems with multiple linear coupled constraints. Due to the non-separability of the constraints, arbitrary random sketching would not be guaranteed to work. Thus, we first investigate necessary and sufficient conditions for the sketch sampling to have well-defined algorithms. Based on these sampling conditions we develop new sketch descent methods for solving general smooth linearly constrained problems, in particular, random sketch descent and accelerated random sketch descent methods. From our knowledge, this is the first convergence analysis of random sketch descent algorithms for optimization problems with multiple non-separable linear constraints. For the general case, when the objective function is smooth and non-convex, we prove for the non-accelerated variant sublinear rate in expectation for an appropriate optimality measure. In the smooth convex case, we derive for both algorithms, non-accelerated and accelerated random sketch descent, sublinear convergence rates in the expected values of the objective function. Additionally, if the objective function satisfies a strong convexity type condition, both algorithms converge linearly in expectation. In special cases, where complexity bounds are known for some particular sketching algorithms, such as coordinate descent methods for optimization problems with a single linear coupled constraint, our theory recovers the best known bounds. Finally, we present several numerical examples to illustrate the performances of our new algorithms.

Keywords: smooth optimization problems, linear coupled constraints, random sketch descent methods, convergence rates.

1. Introduction

During the last decade first order methods, that eventually utilize also some curvature information, have become the methods of choice for solving optimization problems of large sizes arising in all areas of human endeavor where data is available, including machine learning Necoara & Patrascu (2014); Richtárik & Takáč (2014); Shalev-Shwartz & Zhang (2013), portfolio optimization Markowitz (1952); Frongillo & Reid (2015), internet and multi-agent systems Ishii *et al.* (2012), resource allocation Necoara (2013); Xiao & Boyd (2006) and image processing Wright (2012). These large-scale problems are often highly structured (e.g., we encounter sparsity in data, separability in objective function, convexity) and it is important for any optimization method to take advantage of the underlying structure. It turns out that gradient-based algorithms can really benefit from the structure of the optimization models arising in these recent applications Fercoq & Richtárik (2015); Nesterov (2012).

†Corresponding author. Email: ion.necoara@acse.pub.ro
‡Email: Takac.MT@gmail.com

Why random sketch descent methods? The optimization problem we consider in this paper has the following features: *the size of data is very large* so that usual methods based on whole gradient/Hessian computations are prohibitive; moreover *the constraints are coupled linear equalities*. In this case, an appropriate way to approach these problems is through sketch descent methods due to their low memory requirements and low per-iteration computational cost. Sketching is a very general framework that covers as a particular case the (block) coordinate descent methods Luo & Tseng (1993) when the sketch matrix is defined by sampling columns of the identity matrix. Sketching was used, with a big success, to either decrease the computation burden when evaluating the gradient in first order methods Nesterov (2012) or to avoid solving the full Newton direction in second order methods Pilanci & Wainwright (2017). Another crucial advantage of sketching is that for structured problems it keeps the computation cost low, while preserving the amount of data brought from RAM to CPU as for full gradient or Newton methods, and consequently allows for better CPUs utilization on modern multi-core machines. Moreover, in many situations general sketching keeps the per-iteration running-time almost unchanged when compared to the particular sketching of the identity matrix (i.e. comparable to coordinate descent). This, however, leads to a smaller number of iterations needed to achieve the desired quality of the solution, as observed e.g., in Qu *et al.* (2016).

In second order methods sketching was used to decrease the computation cost when evaluating the full Hessian or to avoid solving the full Newton direction. In Pilanci & Wainwright (2017); Berahas *et al.* (2017) a Newton sketch algorithm was proposed for unconstrained self-concordant minimization, which performs an approximate Newton step, wherein each iteration only a sub-sampled Hessian is used. This procedure significantly reduces the computation cost, and still guarantees superlinear convergence for self-concordant functions. In Qu *et al.* (2016), a random sketch method was used to minimize a smooth function which admits a non-separable quadratic upper-bound. In each iteration a block of coordinates was chosen and a subproblem involving a random principal submatrix of the Hessian of the quadratic approximation was solved to obtain an improving direction.

In first order methods particular sketching was used, by choosing as sketch matrix (block) columns of the identity matrix, in order to avoid computation of the full gradient, leading to coordinate descent framework. The main differences in all variants of coordinate descent methods consist in the criterion of choosing at each iteration the coordinate over which we minimize the objective function and the complexity of this choice. Two classical criteria are the cyclic and the greedy coordinate search, which significantly differs by the amount of computations required to choose the appropriate index. For cyclic coordinate search estimates on the rate of convergence were given recently in Beck & Tetruashvili (2013); Gurbuzbalaban *et al.* (2017); Sun & Ye (2016), while for the greedy coordinate search (e.g. Gauss-Southwell rule) the convergence rates were given in Tseng & Yun (2009); Luo & Tseng (1993). Another approach is based on random choice rule, where the coordinate search is random. Complexity results on random coordinate descent methods for smooth convex objective functions were obtained in Nesterov (2012); Necoara (2013). The extension to composite convex objective functions was given e.g. in Richtárik & Takáč (2014); Necoara & Clipici (2016); Lu & Xiao (2015); Necoara & Patrascu (2014). Recently, accelerated Fercoq & Richtárik (2015), parallel Necoara & Clipici (2016); Richtárik & Takáč (2016), asynchronous Liu & Wright (2015) and distributed implementations Takáč *et al.* (2019) of coordinate descent methods were also analyzed. Let us note that the idea of sketching was also successfully applied in various other settings, including Wang *et al.* (2017); Richtárik & Takáč (2020).

Related work. However, most of the aforementioned sketch descent methods assume essentially unconstrained problems, which at best allow separable constraints. In contrast, in this paper we consider sketch descent methods for general smooth problems with linear coupled constraints. Particular sketching-based algorithms, such as greedy coordinate descent, for solving linearly constrained optimization problems were investigated in Tseng & Yun (2009); Luo & Tseng (1993), while more recently in Beck (2014) a greedy coordinate descent method is developed for minimizing a smooth function subject to a single linear equality and additional bound constraints on variables. Random coordinate descent methods that choose at least 2 coordinates at each iteration have been also proposed recently for solving convex problems with a single linear coupled constraint in Necoara (2013); Necoara & Patrascu (2014); Necoara *et al.* (2017).

In all these papers, detailed convergence analysis is provided for both, convex and non-convex settings. Motivated by the work in Necoara *et al.* (2017) several recent papers have tried to extended the random coordinate descent settings to multiple linear coupled constraints Frongillo & Reid (2015); Necoara & Patrascu (2014); Reddi *et al.* (2015). In particular, in Reddi *et al.* (2015) an extension of the 2-random coordinate descent method from Necoara *et al.* (2017) has been analyzed, however under conservative assumptions, such as full rank condition on each block of the matrix describing the constraints. In Frongillo & Reid (2015) a particular sketch descent method is proposed, where the sketch matrices specify arbitrary subspaces that need to generate the kernel of the matrix describing the coupled constraints. However, in the large-scale context and for general linear constraints it is difficult to generate such sketch matrices. Other primal methods for solving linearly constrained convex problems are e.g., center-free gradient methods Xiao & Boyd (2006) or Newton-based methods Wei *et al.* (2013). Another strand of the literature on (linearly) constrained problems develops dual methods, such as augmented Lagrangian method (ALM) Nedelcu *et al.* (2014) or alternating direction method of multipliers (ADMM) Hong & Luo (2017), which embeds a coordinate descent strategy (Gaussian-Seidel decomposition) into each iteration of the ALM. ADMM was originally proposed for problems with a two-block structure and recently extended to the multi-block case, e.g. in Deng *et al.* (2017); Lin *et al.* (2016). Although, in general, ADMM is able to solve more general structured problems, sketch descent methods have the advantage of keeping feasibility throughout the iterations, while ADMM achieves feasibility only at the solution.

Our approach and contribution. Our approach introduces general sketch descent algorithms for solving large-scale smooth optimization problems with multiple linear coupled constraints. Since we have non-separable constraints in the problem formulation, a random sketch descent scheme needs to consider new sampling rules for choosing the coordinates. We first investigate conditions on the sketching of the coordinates over which we minimize at each iteration in order to have well-defined algorithms. Based on these conditions we develop new random sketch descent methods for solving our linearly constrained optimization problem, in particular, random sketch descent and accelerated random sketch descent algorithms. Both methods start from a feasible initial point. However, unlike existing methods, such as coordinate descent, our algorithms are capable of utilizing curvature information, which leads to striking improvements in both theory and practice.

Our contribution. To this end, our main contribution can be summarized as follows:

(*i*) Since we deal with optimization problems having non-separable constraints we need to design sketch descent schemes based on new sampling rules for choosing the sketch matrix. We derive necessary and sufficient conditions on the sketching of the coordinates over which we minimize at each iteration in order to have well-defined algorithms. To our knowledge, the algebraic conditions on the probability distribution defining the sketch matrices from Section 2.3 are new for this class of optimization problems.

(*ii*) Furthermore, from our best knowledge, this paper is the first complete work on the convergence analysis of random sketch descent type algorithms for problems with more than one linear constraint. Our theoretical results consist of new optimization algorithms, accompanied with global convergence guarantees to solve a wide class of non-separable optimization problems.

(*iii*) In particular, we propose a random sketch descent (RSD) algorithm for solving such general optimization problems with multiple coupling constraints. For the general case, when the objective function is smooth and non-convex, we prove sublinear rate in expectation for an appropriate optimality measure. In the smooth convex case we obtain in expectation an ε-accurate solution in at most $\mathcal{O}(1/\varepsilon)$ iterations, while for strongly convex functions the method converges linearly.

(*iv*) We also propose an accelerated random sketch descent (A-RSD) algorithm. From our knowledge, this is the first convergence analysis of an accelerated variant for optimization problems with non-separable linear constraints. In the smooth convex case we obtain in expectation an ε-accurate solution in at most $\mathcal{O}(1/\sqrt{\varepsilon})$ iterations. For strongly convex functions the new accelerated random sketch descent method converges linearly.

Let us emphasize the following points of our contribution. First, our sampling strategies are for multiple linear constraints and thus very different from the existing methods designed only for one linear constraint

(see Section 2.3). Second, we provide new convergence proofs for the random sketch descent method (RSD) that differ from those given already for random coordinate descent (see Theorem 3.2 and Remark 3.1). Third, our accelerated random sketch descent method (A-RSD) is the first designed for this class of problems and requires new proof techniques for deriving the convergence rates compared to standard convergence proofs for accelerated gradient methods (see Theorems 4.2, 4.3, and Remark 4.1). Fourth, our non-accelerated sketch descent algorithm covers as special cases some methods designed for problems with a single linear constraint and coordinate sketch. In these special cases, where convergence bounds are known, our theory recovers the best known bounds or leads to better bounds (see Remark 3.2). We also illustrate, that for some problems, random sketching of the coordinates produces better results than deterministic selection of them. Finally, our framework can be extend to other types of constraints, not necessarily linear, as long as we are able to find an initial feasible point and to solve efficiently the corresponding subproblem from each iteration (which, usually, it will not have a closed form solution as it is the case for the linear constraints) or used to further develop other methods such as Newton-type schemes.

Content. The paper is organized as follows. Section 2 presents necessary and sufficient conditions for the sampling of the sketch matrix. Sections 3 provides a full convergence analysis of the random sketch descent method, while Section 4 derives the convergence analysis for an accelerated variant. Finally, in Section 5 we present some numerical examples to illustrate the performances of our new algorithms.

1.1 *Problem formulation*

We consider the following general smooth optimization problem with multiple linear coupled constraints:

$$f^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad Ax = b, \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a general differentiable function and $A \in \mathbb{R}^{m \times n}$, with $m \ll n$, is such that the feasible set is nonempty. The last condition is satisfied if e.g. A has full row rank. Note that large n and small m are the typical settings for sketching (coordinate descent) methods, see e.g. Beck (2014); Necoara *et al.* (2017); Nesterov (2012). The simplest case is when n is large and $m = 1$, that is we have a single linear constraint $a^T x = b$, as considered in Beck (2014); Necoara (2013); Necoara & Patrascu (2014); Necoara *et al.* (2017). Note that we do not necessarily impose f to be a convex function. From the optimality conditions for our optimization problem (1.1) we have that $x^* \in \mathbb{R}^n$ is a stationary point if there exists some $\lambda^* \in \mathbb{R}^m$ such that:

$$\nabla f(x^*) + A^T \lambda^* = 0 \quad \text{and} \quad Ax^* = b.$$

However, if f is convex, then any x^* satisfying the previous optimality conditions is a global optimum for optimization problem (1.1). Let us define X^* the set of these points. Therefore, $x^* \in X^*$ is a stationary (optimal) point if it is feasible and satisfies the condition:

$$\nabla f(x^*) \in \text{range}(A^T).$$

1.2 *Motivation*

We present below several important applications from which the interest for problems of type (1.1) stems.

1.2.1 *Page ranking.* This problem has many applications in Google ranking, network control, data analysis Ishii *et al.* (2012); Nesterov (2012); Necoara (2013). For a given graph $\mathscr{G}$ let $\bar{E} \in \mathbb{R}^{n \times n}$ be its incidence matrix, which is sparse. Define $E = \bar{E}\ \text{diag}(\bar{E}^T e)^{-1}$, where $e \in \mathbb{R}^n$ denotes the vector with all entries equal to 1. Since $E^T e = e$, i.e. the matrix E is column stochastic, the goal is to determine a vector $x^* \geqslant 0$ such that: $Ex^* = x^*$ and $e^T x^* = 1$. This problem can be written directly in optimization form:

$$\min_{x \in \mathbb{R}^n} f(x) \quad \left(:= \frac{1}{2}\|Ex - x\|^2 \right) \quad \text{s.t.} \quad e^T x = 1,$$

which is a particular case of our optimization problem (1.1) with $m = 1$ and E sparse matrix.

1.2.2 *Machine learning.* Consider the optimization problem associated with the loss minimization of linear predictors without regularization for a training data set containing n observations $a_i \in \mathbb{R}^m$ Shalev-Shwartz & Zhang (2013):

$$\min_{w\in\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^{n} \phi_i(w^T a_i).$$

Here ϕ_i is some loss function, e.g. SVM $\phi_i(z) = \max\{0, 1 - y_i z\}$, logistic regression $\phi_i(z) = \log(1 + \exp(-y_i z))$, ridge regression $\phi_i(z) = (z - y_i)^2$, regression with the absolute value $\phi_i(z) = |z - y_i|$ and support vector regression $\phi_i(z) = \max\{0, |z - y_i| - v\}$ for some predefined insensitivity parameter $v > 0$. Moreover, in classification the labels $y_i \in \{-1, 1\}$, while in regression $y_i \in \mathbb{R}$. Further, let ϕ_i^* denote the Fenchel conjugate of ϕ_i. Then the dual of this problem becomes:

$$\min_{x\in\mathbb{R}^n} f(x) \quad \left(= \frac{1}{n}\sum_{i=1}^{n} \phi_i^*(x_i)\right) \quad \text{s.t.} \quad Ax = 0,$$

where $A = [a_1 \cdots a_n] \in \mathbb{R}^{m\times n}$. Clearly, this problem fits into our model (1.1), with m representing the number of features, n the number of training data, and the objective function f is separable.

1.2.3 *Portfolio optimization.* In the basic Markowitz portfolio selection model Markowitz (1952), see also Frongillo & Reid (2015) for related formulations, one assumes a set of n assets, each with expected returns μ_i, and a covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$, where $\Sigma_{(i,j)}$ is the covariance between returns of assets i and j. The goal is to allocate a portion of the budget into different assets, i.e. $x_i \in \mathbb{R}$ represents a portion of the wealth to be invested into asset i, leading to the first constraint: $\sum_{i=1}^n x_i = 1$. Then, the expected return (profit) is $r = \sum_{i=1}^n \mu_i x_i$ and the variance of the portfolio can be computed as $\sum_{i,j} x_i x_j \Sigma_{(i,j)}$. The investor seeks to minimize risk (variance) and maximize the expected return, which is usually formulates as maximizing profit while limiting the risk or minimizing risk while requiring given expected return. The later formulation can be written as:

$$\min_{x\in\mathbb{R}^n} x^T \Sigma x \quad \text{s.t.} \quad \sum_{i=1}^{n} \mu_i x_i = r, \ \sum_{i=1}^{n} x_i = 1,$$

which clearly fits again into our optimization model (1.1) with $m = 2$. We can further assume that each asset belongs exactly to one class $c \in [C]$, e.g. financials, health care, industrials, etc. The investor would like to diversify its portfolio in such a way that the net allocation in class c is a_c: $\sum_{i=1}^n x_i \mathbf{1}_c(i) = a_c$ for all $c \in [C]$, where $\mathbf{1}_c(i) = 1$ if asset i is in class c and $\mathbf{1}_c(i) = 0$ otherwise. One can observer that in this case we get a similar problem as above, but with C additional linear constraints ($m = C + 2$).

2. Random sketching

It is important to note that stochasticity enters in our algorithmic framework through a user-defined distribution $\mathscr{S}$ describing an ensemble of random matrices $S \in \mathbb{R}^{n\times p}$ (also called *sketch* matrices). We assume that $p \ll n$, in fact we usually require $p \sim \mathscr{O}(m)$ and note that p can also be random (i.e. the $\mathscr{S}$ can return matrices with different p). Our schemes and the underlying convergence theory support virtually all thinkable distributions. The choice of the distribution should ideally depend on the problem itself, as it will affect the convergence speed. However, for now we leave such considerations aside. The basic idea of our algorithmic framework consists of a given feasible x, a sample sketch matrix $S \sim \mathscr{S}$ and a basic update of the form:

$$x^+ = x + Sd \quad \text{such that} \quad ASd = 0, \tag{2.1}$$

where the requirement $ASd = 0$ ensures that the new point x^+ will remain feasible. Clearly, one can choose a distribution $\mathscr{S}$ which will not guarantee convergence to stationary/optimal point. Therefore, we need to impose some minimal necessary conditions for such a scheme to be well-defined. In particular, in order to

avoid trivial updates, we need to choose $S \sim \mathscr{S}$ such that the homogeneous linear system $ASd = 0$ admits also nontrivial solutions, that is we require:

$$\text{range}(S) \cap \ker(A) \neq 0. \tag{2.2}$$

Moreover, since for any feasible x^0 an optimal solution satisfies $x^* \in x^0 + \ker(A)$, it is necessary to require that with our distribution $\mathscr{S}$ we can generate $\ker(A)$:

$$\ker(A) = \text{Span}\left(\cup_{S\sim\mathscr{S}} \left(\text{range}(S) \cap \ker(A)\right)\right). \tag{2.3}$$

Note that the geometric conditions (2.2)-(2.3) are only necessary for a sketch descent type scheme to be well-defined. However, for a discrete probability distribution, having e.g. the property that $\mathbf{P}(S) > 0$ for all $S \sim \mathscr{S}$, condition (2.3) is also sufficient. In Section 2.3 (see Assumption 2.2) we will provide sufficient conditions for a general probability distribution $\mathscr{S}$ in order to obtain well-defined algorithms based on such sketching. Below we provide several examples of probability distributions satisfying our geometric conditions (2.2)-(2.3).

2.1 *Example 1 (finite case)*

Let us consider a finite (or even countable) probability distribution $\mathscr{S}$. Further, let x^0 be a particular solution of the linear system $Ax = b$. For example, if $A^\dagger$ denotes the pseudo-inverse of the matrix A, then we can take $x^0 = A^\dagger b$. Moreover, by the properties of the pseudo-inverse, $I_n - A^\dagger A$ is a projection matrix onto $\ker(A)$, that is $\text{range}(I_n - A^\dagger A) = \ker(A)$. Therefore, any solution of the linear system $Ax = b$ is:

$$x = A^\dagger b + (I_n - A^\dagger A)y,$$

for any $y \in \mathbb{R}^n$. Thus, we may consider a finite (the extension to countable case is straightforward) set of matrices $\Omega = \{S_i \in \mathbb{R}^{n\times p} : \ i = 1:N\}$ endowed with a probability distribution $P_i = \mathbf{P}(S = S_i)$ for all $i \in [N]$ and condition (2.3) requires that the span of the image spaces of $\{S_i\}_{i=1}^N$ contains or is equal to $\text{range}(I_n - A^\dagger A)$:

$$\ker(A) = \text{range}(I_n - A^\dagger A) = \text{Span}\left(\cup_{i:P_i>0}(\text{range}(S_i) \cap \ker(A))\right). \tag{2.4}$$

In particular, we have several choices for the sampling for a finite distribution:

Kernel sketching: If one can compute a basis for $\ker(A)$, then we can take as random sketch matrix $S_i \in \mathbb{R}^{n\times p}$ any block of p elements of this basis endowed with some probability $P_i = \mathbf{P}(S = S_i) > 0$ (for the case $p = 1$ the matrix S_i represents a single element of this basis generating $\ker(A)$). This sampling was also considered in Frongillo & Reid (2015). Clearly, in this particular case condition (2.2) and condition (2.3) or equivalently (2.4) hold since $\ker(A) = \text{Span}\left(\cup_{i=1}^N \text{range}(S_i)\right)$.

Coordinate sketching: However, for a general matrix A it is difficult to compute a basis of $\ker(A)$. A simple alternative is to consider then any p-tuple $\mathscr{N} = (i_1 \cdots i_p) \in 2^{[n]}$, with $p > m$, and the corresponding random sketch matrix $S_\mathscr{N} = [e_{i_1} \cdots e_{i_p}]$, where e_i denotes the ith column of the identity matrix I_n, with some probability distribution $P_\mathscr{N}$ over the set of p-tuples in $2^{[n]}$. It is clear that for this choice condition (2.2) and condition (2.3) or equivalently (2.4) also hold. For the particular case when we have a single linear coupled constraint, i.e. $a^T x = b$, we can take random matrices $S_{(ij)} = [e_i \ e_j]$ also considered e.g. in Necoara (2013). This particular sketch matrix based on sampling columns of the identity matrix leads to coordinate descent framework. However, the other examples (including those from Section 2.2) show that *our sketching framework is more general than coordinate descent*.

General sketching: Instead of working with the matrix I_n, as considered previously, we can take any orthogonal or full rank matrix $\mathscr{I} \in \mathbb{R}^{n\times n}$ having the columns $\mathscr{I}_i$ and thus forming a basis of $\mathbb{R}^n$. Then, we can consider p tuples $\mathscr{N} = (i_1, \cdots, i_p) \in 2^{[n]}$, with $p > m$, and the corresponding random sketch matrix $S_\mathscr{N} = [\mathscr{I}_{i_1} \cdots \mathscr{I}_{i_p}]$, with some probability distribution $P_\mathscr{N}$ over the set of p-tuples in $2^{[n]}$. Clearly, for this choice of the random sketch matrices S the conditions (2.2) and (2.3) or equivalently (2.4) still hold.

2.2 *Example 2 (infinite case)*

Let us now consider a continuous (uncountable) probability distribution $\mathscr{S}$. We can consider in this case two simple sampling strategies:

Kernel sketching: If one can sample easily a random matrix B such that $\text{range}(B) = \ker(A)$, then choose one or several columns from this matrix as a sketch matrix S. In this case $p \geqslant 1$.

General sketching: Alternatively, we can sample random full rank matrices in $\mathbb{R}^{n\times n}$ and then define S to be random $p > m$ columns. Furthermore, since it is known that random Gaussian matrices are full rank almost surely, then we can define $S \sim \mathscr{N}^{n\times p}$ to be a random Gaussian matrix. Similarly, we can consider random uniform matrices and define e.g. $S \sim \text{Unif}(-1,1)^{n\times p}$.

A sufficient condition for a well-defined sampling in the infinite case is to ensure that in expectation one can move in any direction in $\ker(A)$. Considering the general update rule (2.1), we see that if we sample $S \in \mathbb{R}^{n\times p}$, then our update can be only Sd for some $d \in \mathbb{R}^p$. Now, we also have a condition, that we want to stay in the $\ker(A)$, and therefore d cannot be anything, but has to be chosen such that $Sd \in \ker(A)$. Now, this restricts the set of possible d's to be such that:

$$ASd = 0 \qquad \Rightarrow \qquad d = (I_p - (AS)^\dagger(AS))t$$

for some $t \in \mathbb{R}^p$. Recall, that we allow p to be also random, hence to derive the sufficient condition we need to have some quantity with dimension independent on p. Note that each $t \in \mathbb{R}^p$ can be represented as $t = S^T t' + t''$ for some $t' \in \mathbb{R}^n$ and $t'' \in \ker(S)$. Hence, we see that if S is a general matrix and $t \in \mathbb{R}^p$, then we can move in the direction:

$$Sd = S(I - (AS)^\dagger(AS))t = S(I - (AS)^\dagger(AS))(S^T t' + t'') = S(I - (AS)^\dagger(AS))S^T t',$$

since $St'' = 0$. Hence, we have the ability to move in range $\left(S(I - (AS)^\dagger(AS))S^T\right)$. Now, the condition to be able to move in $\ker(A)$ can be expressed as requiring that on expectation we can move anywhere in $\ker(A)$:

$$\text{range}\left(\mathbf{E}\left[S(I - (AS)^\dagger(AS))S^T\right]\right) = \ker(A), \tag{2.5}$$

provided that the expectation exists and is finite. Note, that this condition must hold also for a discrete probability distribution, however the condition (2.3) is more intuitive in the discrete case. In the next section we provide algebraic sufficient conditions on the sampling for a general probability distribution $\mathscr{S}$ in order to obtain well-defined algorithms.

2.3 *Sufficient conditions for sketching*

It is well known that in order to derive any reasonable convergence guarantees for a minimization scheme we need to impose some smoothness property on the objective function. Therefore, throughout the paper we consider the following blanket assumption on the smoothness of f:

ASSUMPTION 2.1 For any feasible x^0 there exists a positive semidefinite matrix M such that M is positive definite on $\ker(A)$ and the following inequality holds:

$$f(y) \leqslant f(x) + \langle \nabla f(x), y - x\rangle + \frac{1}{2}(y-x)^T M(y-x), \quad \forall x, y \in x^0 + \ker(A). \tag{2.6}$$

Note that for a general (possibly non-convex) differentiable function f the smoothness inequality (2.6) does not imply that the objective function f has Lipschitz continuous gradient, so our assumption is less conservative than requiring Lipchitz gradient assumption. Note that when f is convex the condition (2.6) is equivalent with Lipschitz continuity of the gradient of f on $x^0 + \ker(A)$, see Nesterov (2013). In particular, if $M = L \cdot I_n$ for some Lipschitz constant $L > 0$ we recover the usual definition of Lipschitz continuity of the gradient for the class of convex functions. However, (2.6) allows for our methods to take into account more structure on the objective function, when this is available. For example, if the objective function

is quadratic we can choose M as the Hessian, and our methods can be interpreted as novel extensions to more general optimization models of the recently introduced iterative Hessian sketch method for minimizing self-concordant objective functions Pilanci & Wainwright (2017). If the objective function is separable, i.e. $f(x) = \sum_i f_i(x_i)$ as in the dual SVM application from Section 1.2.2, and there exist constants $M_i > 0$ such that $f_i''(x_i) \leqslant M_i$, we can choose $M = \mathrm{diag}(M_i, i \in [n])$. Finally, there are many application having the objective function of the form $f(x) = g(\bar{M}x)$ for some appropriate matrix $\bar{M}$ and some smooth convex function g, as in page ranking (see Section 1.2.1) or portfolio optimization (see Section 1.2.3). For this case, if L_g is an upper bound on the Hessian of g on the $\ker(A)$, then we can easily compute an M satisfying Assumption 2.1 as $M = L_g \bar{M}^T \bar{M}$, whereas obtaining the classical bound LI_n is a computationally more intensive task and would lead to worse theoretical convergence speed. *In conclusion, our sketching methods derived below are based on* (2.6) *and therefore they have the capacity to utilize curvature information, this leading usually to striking improvements in theory and practice*.

The reader should also note that we can further relax the condition (2.6) and require smoothness of f with respect to any image space generated by the random matrix S. More precisely, it is sufficient to assume that for any sample $S \sim \mathscr{S}$ there exists a positive semidefinite matrix M_S such that M_S is positive definite on $\ker(A) \cap \mathrm{range}(S)$ and the following inequality holds:

$$f(y) \leqslant f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2}(y-x)^T M_S (y-x) \quad \forall x, y \in x^0 + \ker(A) \ \wedge \ x - y \in \mathrm{range}(S).$$

Note that if $M_S = M$ for all S we recover the relation (2.6). For simplicity of the exposition in the sequel we assume (2.6) to be valid, although all our convergence results can be also extended under previous smoothness condition given in terms of M_S.

From the above discussion it is clear that the direction d in our basic update (2.1) needs to be in the kernel of matrix AS. However, it is well known that the projection onto $\ker(AS)$ is given by the projection matrix:

$$P_S = I_p - (AS)^\dagger (AS).$$

Clearly, we have $\ker(AS) = \mathrm{range}(P_S)$. Let us further define the symmetric matrix:

$$Z_S = SP_S (P_S^T S^T M S P_S)^\dagger P_S^T S^T \in \mathbb{R}^{n \times n}. \tag{2.7}$$

The matrix Z_S plays a key role in the accelerated and non-accelerated random sketch algorithms we propose in the sequel. For example, both algorithms have an update step of the form $x^{k+1} = x^k - Z_S \nabla f(x^k)$ (see Sections 3 and 4). Hence, we derive below some important properties for the matrix Z_S, since they are useful for algorithm development. First we observe that:

LEMMA 2.1 For any probability distribution $\mathscr{S}$ the matrix Z_S is symmetric ($Z_S = Z_S^T$), positive semidefinite ($Z_S \succeq 0$), and for any $u \in \mathrm{range}(A^T)$ we have $Z_S u = 0$, that is $\mathrm{range}(A^T) \subseteq \ker(Z_S)$. Moreover, the following identity holds $Z_S M Z_S = Z_S$.

Proof. It is clear that Z_S is positive semidefinite matrix since M is assumed positive semidefinite. It is well-known that for any given matrix B its pseudo-inverse satisfies $BB^\dagger B = B$ and $B^\dagger B B^\dagger = B^\dagger$. Now, for the first statement given the expression of Z_S it is sufficient to prove that $P_S^T S^T u = 0$ for $u \in \mathrm{range}(A^T)$. However, if $u \in \mathrm{range}(A^T)$ then there exists y such that $u = A^T y$ and consequently we have:

$$\begin{aligned} P_S^T S^T u = P_S^T S^T A^T y &= (I - (AS)^\dagger (AS))^T (AS)^T y \\ &= [(AS)(I - (AS)^\dagger (AS))]^T y \\ &= ((AS) - (AS)(AS)^\dagger (AS))^T y = 0, \end{aligned}$$

where in the last equality we used the first property of pseudo-inverse $(AS)(AS)^\dagger (AS) = AS$. For the second part of the lemma we use the expression of Z_S and the second property of the pseudo-inverse applied to the matrix $(P_S^T S^T M S P_S)^\dagger$, that is:

$$Z_S M Z_S = [SP_S (P_S^T S^T M S P_S)^\dagger P_S^T S^T] M [SP_S (P_S^T S^T M S P_S)^\dagger P_S^T S^T] = Z_S,$$

which concludes our statements. □

Now, since the random matrix Z_S is positive semidefinite, then we can define its expected value, which is also a symmetric positive semidefinite matrix:

$$Z = \mathbf{E}_S[Z_S]. \tag{2.8}$$

In the sequel we also consider the following assumption on the expectation matrix Z:

ASSUMPTION 2.2 We assume that the distribution $\mathscr{S}$ is chosen such that Z_S has a finite mean, that is the matrix Z is well defined, and positive definite (notation $Z \succ 0$) on $\ker(A)$.

It is straightforward to see that this assumption holds automatically for all the examples from Section 2.1, provided that M is positive definite matrix. As we will see below, Assumption 2.2 is a sufficient condition on the probability distribution $\mathscr{S}$ in order to ensure convergence of our algorithms that will be defined in the sequel. *To our knowledge our algebraic characterization of the probability distribution defining the sketch matrices S for problems with multiple non-separable linear constraints seems to be new*.

Note that the necessary condition (2.2) holds provided that $Z_S \neq 0$. Indeed, from Lemma 2.1 we have $\text{range}(A^T) \subseteq \ker(Z_S)$ for all $S \sim \mathscr{S}$ and $\ker(Z_S) \perp \text{range}(Z_S)$. Therefore, we get that $\text{range}(A^T) \perp \text{range}(Z_S)$ and we know that $\text{range}(A^T) \perp \ker(A)$. Let $z \in \text{range}(Z_S) \subseteq \mathbb{R}^n$, $z \neq 0$, then there exists unique $z_1 \in \text{range}(A^T)$ and $z_2 \in \ker(A)$ such that $z = z_1 + z_2$. Moreover, we have $z \perp \text{range}(A^T)$, i.e. $z \perp z_1$, which implies that $\langle z_1 + z_2, z_1 \rangle = \|z_1\|^2 + 0 = 0$. Thus, $z_1 = 0$ and $z \in \ker(A)$. From the last relation, we get:

$$\text{range}(Z_S) \subseteq \ker(A).$$

Moreover, from the definition of the symmetric matrix Z_S we have $\text{range}(Z_S) \subseteq \text{range}(S)$, which combined with the previous relation leads to:

$$\text{range}(Z_S) \subseteq \ker(A) \cap \text{range}(S),$$

and consequently proving that the condition (2.2) holds provided that $Z_S \neq 0$. Moreover, we can show that the necessary condition (2.3) holds if Z satisfies Assumption 2.2.

LEMMA 2.2 If Assumption 2.2 holds, then the following identity takes place

$$\text{range}(A^T) = \ker(Z)$$

and consequently $Z^\dagger Z$ is a projection matrix onto $\ker(A)$, where $Z^\dagger$ denotes the pseudo-inverse of the matrix Z. Moreover, the necessary condition (2.3) is also valid.

Proof. First we note that $Z \succ 0$ on $\ker(A)$, if and only if $Z \succ 0$ on $\mathbb{R}^n \setminus \text{range}(A^T)$. Indeed, since $\ker(A) \subseteq (\mathbb{R}^n \setminus \text{range}(A^T)) \cup \{0\}$, one direction of the equivalence is straightforward. In order to prove the other direction of the equivalence, we first note from Lemma 2.1 that $Zu = 0$ for any $u \in \text{range}(A^T)$. Now, any $x \in \mathbb{R}^n \setminus \text{range}(A^T)$ can be written as $x = A^T\lambda + x_\perp$, where $\lambda \in \mathbb{R}^m$ and $x_\perp \in \ker(A) \setminus \{0\}$, and consequently $x^T Z x = (A^T\lambda + x_\perp)^T Z (A^T\lambda + x_\perp) = x_\perp^T Z x_\perp > 0$, which proves the equivalence.

Furthermore, for proving the first part of the lemma we use that $\text{range}(A^T) \subseteq \ker(Z_S)$ for all $S \sim \mathscr{S}$ (see Lemma 2.1). This means that $\text{range}(A^T) \subseteq \cap_{S\sim\mathscr{S}} \ker(Z_S) \subseteq \ker(Z)$. The other inclusion follows by contradiction. Indeed, let us assume that there exists $u \notin \text{range}(A^T)$ such that $Zu = 0$, or equivalently $Zu = 0$ for some $u \in \mathbb{R}^n \setminus \text{range}(A^T)$. However, note that $Z \succ 0$ on $\ker(A)$ if and only if $Z \succ 0$ on $\mathbb{R}^n \setminus \text{range}(A^T)$, which contradicts our assumption. Moreover, it is well-known that $Z^\dagger Z$ is an orthogonal projector onto $\text{range}(Z)$ and the rest follows from standard algebraic arguments. In conclusion, the first statement holds.

For the second part of the lemma we observe that for any non-zero $u \in \ker(A)$, we have $Zu \neq 0$, that is $u \notin \ker(Z)$. In conclusion, we get $\ker(A) \subseteq (\mathbb{R}^n \setminus \ker(Z)) \cup \{0\}$. But, $\text{range}(Z) \subseteq \text{Span}(\cup_{S\sim\mathscr{S}} \text{range}(Z_S))$, from which we can conclude $\ker(A) \subseteq \text{Span}(\cup_{S\sim\mathscr{S}} \text{range}(Z_S))$ and consequently condition (2.3) holds. □

The primal-dual "norms". Since the matrix Z_S is positive semidefinite, matrix Z is also positive semidefinite. Moreover, from Lemma 2.1 we conclude that $\text{range}(A^T) \subseteq \ker(Z)$. In the sequel we assume that

$S \sim \mathscr{S}$ such that Z is a positive definite matrix on $\ker(A)$ and consequently on $\mathbb{R}^n \setminus \text{range}(A^T)$ (see Assumption 2.2). Then, we can define a norm induced by the matrix Z on $\ker(A)$ or even $\mathbb{R}^n \setminus \text{range}(A^T)$. This norm will be used subsequently for measuring distances in the subspace $\ker(A)$. More precisely, we define the *primal norm* induced by the positive semidefinite matrix Z as:

$$\|u\|_Z = \sqrt{u^T Z u} \quad \forall u \in \mathbb{R}^n.$$

Note that $\|u\|_Z = 0$ for all $u \in \text{range}(A^T)$ (see Lemma 2.1) and $\|u\|_Z > 0$ for all $u \in \mathbb{R}^n \setminus \text{range}(A^T)$. On the subspace $\ker(A)$ we introduce the extended dual norm:

$$\|x\|_Z^* = \max_{u \in \mathbb{R}^n : \|u\|_Z \leqslant 1} \langle x, u \rangle \quad \forall x \in \ker(A).$$

Using the definition of conjugate norms, the Cauchy-Schwartz inequality holds:

$$\langle u, x \rangle \leqslant \|u\|_Z \cdot \|x\|_Z^* \quad \forall x \in \ker(A),\ u \in \mathbb{R}^N. \tag{2.9}$$

LEMMA 2.3 Under Assumption 2.2 the primal and dual norms have the following expressions:

$$\|u\|_Z = \sqrt{u^T Z u}, \quad \|x\|_Z^* = \sqrt{x^T Z^\dagger x} \quad \forall u \in \mathbb{R}^n, \quad \forall x \in \ker(A). \tag{2.10}$$

Proof. Let us consider any $\hat{u} \in \text{range}(A^T)$. Then, the dual norm can be computed for any $x \in \ker(A)$ as:

$$\begin{aligned}
\|x\|_Z^* &= \max_{u \in \mathbb{R}^n :\ \langle Zu, u \rangle \leqslant 1} \langle x, u \rangle = \max_{u : \langle Z(u-\hat{u}), u-\hat{u} \rangle \leqslant 1} \langle x, u - \hat{u} \rangle \\
&= \max_{u : \langle Zu, u \rangle \leqslant 1, u \in \ker(A)} \langle x, u \rangle = \max_{u : \langle Zu, u \rangle \leqslant 1, Au = 0} \langle x, u \rangle = \max_{u : \langle Zu, u \rangle \leqslant 1, u^T A^T A u \leqslant 0} \langle x, u \rangle \\
&= \min_{\nu, \mu \geqslant 0} \max_{u \in \mathbb{R}^n} [\langle x, u \rangle + \mu (1 - \langle Zu, u \rangle) - \nu \langle A^T A u, u \rangle] \\
&= \min_{\nu, \mu \geqslant 0} \mu + \langle (\mu Z + \nu A^T A)^{-1} x, x \rangle = \min_{\nu \geqslant 0} \min_{\mu \geqslant 0} [\mu + \frac{1}{\mu} \langle (Z + \frac{\nu}{\mu} A^T A)^{-1} x, x \rangle] \\
&= \min_{\zeta \geqslant 0} \sqrt{\langle (Z + \zeta A^T A)^{-1} x, x \rangle}.
\end{aligned}$$

We obtain an extended dual norm that is well defined on the subspace $\ker(A)$:

$$\|x\|_Z^* = \min_{\zeta \geqslant 0} \sqrt{\langle (Z + \zeta A^T A)^{-1} x, x \rangle} \quad \forall x \in \ker(A). \tag{2.11}$$

The eigenvalue decomposition of the positive semidefinite matrix Z can be written in the following form $Z = U \text{diag}(\lambda_1, \cdots, \lambda_r, 0, \cdots, 0) U^T$, where λ_i are its positive eigenvalues and the columns of orthogonal matrix $U = [U_{ker}\ U_{range}]$ are the corresponding eigenvectors, U_{ker} generating $\ker(A)$ and U_{range} generating $\text{range}(A^T)$. Then, we have:

$$(Z + \zeta A^T A)^{-1} = U \text{diag}(\lambda_1, \cdots, \lambda_r, \zeta \lambda_{r+1}, \cdots, \zeta \lambda_n)^{-1} U^T,$$

where $\lambda_{r+1}, \cdots, \lambda_n$ are the nonzero eigenvalues of symmetric matrix $A^T A$. From (2.11) it follows that our newly defined dual norm has the following closed form:

$$\|x\|_Z^* = \sqrt{x^T Z^\dagger x} \quad \forall x \in \ker(A),$$

where $Z^\dagger$ denotes the pseudoinverse of matrix Z. □

The following example shows that the 2-coordinate sampling proposed in Necoara *et al.* (2017) (in the presence of a single linear constraint $m = 1$) is just a special case of the sketching analyzed in this paper:

EXAMPLE 2.3 Let us consider the following optimization problem:

$$f(x)=\sum_{i=1}^{n} f_i(x_i) \quad \text{subject to} \quad \sum_{i=1}^{n} x_i = b.$$

In this case, assuming that each scalar function f_i has L_i Lipschitz continuous gradient, then we consider $M=\mathrm{diag}(L_1,\cdots,L_n)$. Moreover, we can take any random pair of coordinates (i,j) with $i,j=1:n$, $i<j$ and consider the particular sketch matrix $S_{(ij)}=[e_i\, e_j]$. Note that, for simplicity, we focus here on Lipschitz dependent probabilities for choosing the pair (i,j), that is $P_{(i,j)}=(L_i+L_j)/((n-1)L)$ with $L=\sum_{i=1}^{n} L_i$. Following basic derivations we get:

$$\begin{aligned} Z_{(ij)} &= \frac{1}{L_i+L_j} S_{(ij)} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} S_{(ij)}^T = \frac{1}{L_i+L_j}(e_i-e_j)(e_i-e_j)^T, \\ Z &= \frac{n}{(n-1)L}\left(I_n-\frac{1}{n}ee^T\right), \qquad Z^\dagger = \frac{(n-1)L}{n}\left(I_n-\frac{1}{n}ee^T\right). \end{aligned} \tag{2.12}$$

Clearly, $Z \succ 0$ on $\ker(A)$ and thus Assumption 2.2 holds. Similarly, we can compute explicitly Z and $Z^\dagger$ for the fixed selection of the pair of coordinates $(i,i+1)$ with $i=1:n-1$.

3. Random Sketch Descent (RSD) algorithm

For the large-scale optimization problem (1.1) methods which scale cubically, or even quadratically, with the problem size n is already out of the question; instead, linear scaling of the computational costs per-iteration is desired. Clearly, optimization problem (1.1) can be solved using projected first order methods, such as gradient or accelerated gradient, both algorithms having comparable cost per iteration Nesterov (2013). In particular, both methods require the computation of the full gradient $\nabla f(x)$ and finding the optimal solution of a subproblem with quadratic objective over the subspace $ker(A)\subset\mathbb{R}^n$:

$$\min_{d\in\mathbb{R}^n: Ad=0} f(x)+\langle \nabla f(x), d\rangle + \frac{1}{2}\, d^T M d. \tag{3.1}$$

For example, for the projected gradient method since we assume M positive definite on $\ker(A)$ (see Assumption 2.1), then the previous subproblem has a unique solution leading to the following gradient iteration:

$$x_G^{k+1} = x_G^k - Z_{I_n}\nabla f(x_G^k), \tag{3.2}$$

where $Z_{I_n}\in\mathbb{R}^{n\times n}$ is obtained by replacing $S=I_n$ in the definition of the matrix Z_S. However, for very large n even the first iteration is not computable, since the cost of computing Z_{I_n} is cubic in the problem dimension (i.e. of order $\mathscr{O}(n^3)$ operations) for general matrix M. Moreover, since usually Z_{I_n} is a dense matrix regardless of the matrix M being dense or sparse, the cost of the subsequent iterations is quadratic in the problem size n (i.e. $\mathscr{O}(n^2)$). Hence, full gradient type methods are prohibitive when n is large and m is small. Therefore, the development of new optimization algorithms that target linear cost per iteration and nearly dimension-independent convergence rate is needed. These properties can be achieved using the sketch descent framework. In particular, let us assume that the initial iterate x^0 is a feasible point, i.e. $Ax^0=b$. Then, the first algorithm we propose, Random Sketch Descent (RSD) algorithm, chooses at each iteration a random sketch matrix $S\in\mathbb{R}^{n\times p}$ according to the probability distribution $\mathscr{S}$ and find a new direction solving a simple subproblem (see Algorithm 1 below). Let us explain the update rule of our algorithm RSD. Note that the new direction in the update $x^{k+1}=x^k+Sd^k$ of RSD is computed from

Algorithm 1 Algorithm RSD

choose $x^0 \in \mathbb{R}^n$ such that $Ax^0 = b$
for $k \geqslant 0$ **do**
  Sample $S \sim \mathscr{S}$ and perform the update:
  $x^{k+1} = x^k - Z_S \nabla f(x^k)$.
end for

a subproblem with quadratic objective over the subspace $\ker(AS) \subset \mathbb{R}^p$ that it is simpler than subproblem (3.1) corresponding to the full gradient:

$$d^k = \arg\min_{d \in \mathbb{R}^p : ASd = 0} f(x^k) + \langle \nabla f(x^k), Sd \rangle + \frac{1}{2}\, d^T S^T M S d. \tag{3.3}$$

We observe that from the feasibility condition $ASd = 0$ we can compute d as:

$$d = P_S t \quad \left(:= (I_p - (AS)^\dagger (AS))t\right),$$

for some t. Then, the constrains will not be violated. Now, let's plug this into the objective function of the subproblem, to obtain an unconstrained problem in t:

$$t^k = \arg\min_{t \in \mathbb{R}^p} \langle \nabla f(x^k), S((I_p - (AS)^\dagger (AS))t) \rangle + \frac{1}{2} \|S(I_p - (AS)^\dagger (AS))t\|_M^2.$$

Then, from the first order optimality conditions we obtain that:

$$P_S^T S^T M S P_S t^k = -P_S^T S^T \nabla f(x^k),$$

and hence we can define t^k as

$$t^k = -(P_S^T S^T M S P_S)^\dagger P_S^T S^T \nabla f(x^k).$$

In conclusion we obtain the following update rule for our RSD algorithm:

$$x^{k+1} = x^k - \underbrace{S P_S (P_S^T S^T M S P_S)^\dagger P_S^T S^T}_{=Z_S} \nabla f(x^k) = x^k - Z_S \nabla f(x^k). \tag{3.4}$$

After k iterations of the RSD algorithm, we generate a random output $(x^k, f(x^k))$, which depends on the observed implementation of the random variable:

$$\mathscr{F}_k = (S_0, \cdots, S_{k-1}).$$

Let us define the expected value of the objective function w.r.t. $\mathscr{F}_k$:

$$\phi_k = \mathbf{E}\left[f(x^k) \right].$$

Next, we compute the decrease of the objective function after one random step:

$$\begin{aligned} f(x^{k+1}) &= f(x^k + S P_S t^k) = f(x^k - Z_S \nabla f(x^k)) \\ &\overset{(2.6)}{\leqslant} f(x^k) - \langle \nabla f(x^k), Z_S \nabla f(x^k) \rangle + \frac{1}{2} \|Z_S \nabla f(x^k)\|_M^2 \\ &= f(x^k) - \langle \nabla f(x^k), Z_S \nabla f(x^k) \rangle + \frac{1}{2} \nabla f(x^k)^T Z_S M Z_S \nabla f(x^k) \\ &= f(x^k) - \langle \nabla f(x^k), Z_S \nabla f(x^k) \rangle + \frac{1}{2} \nabla f(x^k)^T Z_S \nabla f(x^k) \\ &= f(x^k) - \frac{1}{2} \langle \nabla f(x^k), Z_S \nabla f(x^k) \rangle. \end{aligned} \tag{3.5}$$

Then, we obtain the following strict decrease for the objective function in the conditional expectation:

$$\mathbf{E}[f(x^{k+1})|\mathscr{F}_k] \leqslant f(x^k) - \frac{1}{2}\|\nabla f(x^k)\|_Z^2, \tag{3.6}$$

provided that x^k is not optimal. This holds since we assume that $Z \succ 0$ on $\mathbb{R}^n \setminus \text{range}(A^T)$ and since any feasible x satisfying $\nabla f(x) \in \text{range}(A^T)$ is optimal for the original problem. Therefore, RSD algorithm belongs to the class of descent methods.

3.1 *Computation cost per-iteration for RSD*

It is easy to observe that if the cost of updating the gradient ∇f is negligible, then the cost per iteration in RSD is given by the computational effort of finding the solution of the subproblem. The sketch sampling $\mathscr{S}$ can be completely dense (e.g. Gaussian random matrix) or can be extremely sparse (e.g. a few columns of the identity matrix).

Case 1: dense sketch matrix S. In this case, since we assume $p \ll n$ (in fact we usually choose p of order $\mathscr{O}(m)$ or even smaller), then the computational cost per-iteration in the update (3.4) is linear in n (more precisely of order $\mathscr{O}(pmn)$) plus the cost of computing the matrix $S^TMS \in \mathbb{R}^{p\times p}$. Clearly, if M is also a dense matrix, then the cost of computing the matrix S^TMS is quadratic in n. However, it can be reduced substantially, that is the cost of computing this matrix depends linearly on n, when e.g. we have available a decomposition of the matrix M as $M = \bar{M}^T\bar{M}$, with $\bar{M} \in \mathbb{R}^{\bar{p}\times n}$ and $\bar{p} \ll n$, or M is sparse.

Case 2: sparse sketch matrix S. For simplicity, we can assume that S is chosen as few columns of the identity matrix and thus obtaining a coordinate descent type method. In this case, the cost per-iteration of RSD is independent of the problem size n. For example, the cost of computing $(AS)^\dagger$ is $\mathscr{O}(m^2p)$, while the cost of computing $(P_S^TS^TMSP_S)^\dagger$ is $\mathscr{O}(p^3)$.

In conclusion, in all situations the iteration (3.4) of RSD is much computationally cheaper (at least one order of magnitude) than the iteration (3.2) corresponding to the full gradient. Based on the decrease of the objective function (3.6) we can derive different convergence rates for our algorithm RSD depending on the assumptions imposed on the objective function f.

3.2 *Convergence rate: smooth case*

We derive in this section the convergence rate of the sequence generated by the RSD algorithm when the objective function is only smooth (Assumption 2.1). Recall that in the non-convex settings a feasible x^* is a stationary point for optimization problem (1.1) if $\nabla f(x^*) \in \text{range}(A^T)$. On the other hand, for any feasible x we have the unique decomposition of $\nabla f(x) \in \mathbb{R}^n$:

$$\nabla f(x) = A^T\lambda + \nabla f(x)_\perp, \quad \text{where} \quad \lambda \in \mathbb{R}^m,\ \nabla f(x)_\perp \in \ker(A).$$

It is clear that if a feasible x satisfies $\nabla f(x)_\perp = 0$, then such an x is a stationary point for (1.1). In conclusion, a good measure of optimality for a feasible x is described in terms of $\|\nabla f(x)_\perp\|$. The theorem below provides a convergence rate for the sequence generated by RSD in terms of this optimality measure:

THEOREM 3.1 Let f be bounded from below, i.e. there exists $\bar{f} > -\infty$ such that $\min_{x\in x^0+\ker(A)} f(x) \geqslant \bar{f}$ and Assumptions 2.1 and 2.2 hold. Then, the iterates of RSD have the following sublinear convergence rate in expectation:

$$\min_{0\leqslant l\leqslant k-1} \mathbf{E}[\|\nabla f(x^l)_\perp\|_Z^2] \leqslant \frac{2(f(x^0)-\bar{f})}{k}. \tag{3.7}$$

Proof. Taking expectation over the entire history $\mathscr{F}_k$ in (3.6) we get:

$$\phi_{k+1} \leqslant \phi_k - \frac{1}{2}\mathbf{E}[\|\nabla f(x^k)\|_Z^2]. \tag{3.8}$$

Summing the previous relation and using that f is bounded from below we further get:

$$\sum_{l=0}^{k-1} \mathbf{E}[\|\nabla f(x^l)\|_Z^2] \leqslant 2(\phi_0 - \phi_k) \leqslant 2(\phi_0 - \bar{f}).$$

Using the unique decomposition $\nabla f(x^l) = A^T \lambda^l + \nabla f(x^l)_\perp$ for all l and since $\ker(Z) = \text{range}(A^T)$, then we obtain $\|\nabla f(x^l)\|_Z^2 = \|\nabla f(x^l)_\perp\|_Z^2$. Therefore, taking the limit as $k \to \infty$ we obtain the asymptotic convergence $\lim_{k\to\infty} \mathbf{E}[\|\nabla f(x^k)_\perp\|_Z^2] = 0$. Moreover, since $Z \succ 0$ on $\ker(A)$ and $\nabla f(x^l)_\perp \in \ker(A)$ we also get:

$$\min_{0\leqslant l\leqslant k-1} \mathbf{E}[\|\nabla f(x^l)_\perp\|_Z^2] \leqslant \frac{2(f(x^0) - \bar{f})}{k},$$

which concludes our statement. □

3.3 *Convergence rate: smooth convex case*

In the next theorem we prove sublinear convergence for RSD in expected value of the objective function in the smooth convex case:

THEOREM 3.2 Let the objective function f be convex and Assumptions 2.1 and 2.2 hold. Let also define $c = \lambda_{\max}(M^{-1/2} Z^\dagger M^{-1/2}) < \infty$. Then, the iterates generated by RSD have the following sublinear convergence rate in the expected value of the objective function:

$$\phi_k - f^* \leqslant \frac{1}{k+c} \left(\frac{1}{2} \min_{x^* \in X^*} \|x^0 - x^*\|_{Z^\dagger}^2 + c(f(x^0) - f^*) \right) \tag{3.9}$$

Proof. Recall that all our iterates are feasible, i.e. $x^k \in x^0 + \ker(A)$. Let us define $r_k^2 = (\|x^k - x^*\|_Z^*)^2 = \|x^k - x^*\|_{Z^\dagger}^2$. Then, we have:

$$\begin{aligned} r_{k+1}^2 &= \|x^{k+1} - x^*\|_{Z^\dagger}^2 = \|x^k - Z_S \nabla f(x^k) - x^*\|_{Z^\dagger}^2 \\ &= r_k^2 - 2 \underbrace{(x^k - x^*)^T Z^\dagger Z_S \nabla f(x^k)}_{:=T_1} + \underbrace{\|Z_S \nabla f(x^k)\|_{Z^\dagger}^2}_{:=T_2}. \end{aligned} \tag{3.10}$$

In the following we find the appropriate bounds for the two terms T_1 and T_2, respectively. For the term T_1 by taking conditional expectation with respect to $\mathscr{F}_k$ we obtain:

$$\mathbf{E}[T_1|\mathscr{F}_k] = \mathbf{E}[(x^k - x^*)^T Z^\dagger Z_S \nabla f(x^k)|\mathscr{F}_k] = (x^k - x^*)^T Z^\dagger Z \, \nabla f(x^k).$$

Moreover, we have that $x^k - x^* \in \ker(A)$, or equivalently by Lemma 2.2, $x^k - x^* \in \text{range}(Z^T) = \text{range}(Z)$, since Z is symmetric matrix. Therefore, there exists some u^k such that $x^k - x^* = Zu^k$, and by utilizing the fact that $Z = ZZ^\dagger Z$, we can continue the above equality as:

$$\mathbf{E}[T_1|\mathscr{F}_k] = \left(ZZ^\dagger Zu^k\right)^T \nabla f(x^k) = (Zu^k)^T \nabla f(x^k) = (x^k - x^*)^T \nabla f(x^k). \tag{3.11}$$

Now, we also derive a bound for the second term T_2. From the definition of $c = \lambda_{\max}(M^{-1/2} Z^\dagger M^{-1/2})$, it follows that $Z^\dagger \preceq cM$. Then, we get:

$$\begin{aligned} T_2 &= \|Z_S \nabla f(x^k)\|_{Z^\dagger}^2 = \nabla f(x^k)^T Z_S^T Z^\dagger Z_S \nabla f(x^k) \\ &\leqslant \nabla f(x^k)^T Z_S^T (cM) Z_S \nabla f(x^k) = c \nabla f(x^k)^T Z_S \nabla f(x^k), \end{aligned}$$

where the last relation follows from Lemma 2.1. By taking now expectation with respect to $\mathscr{F}_k$ we obtain:

$$\mathbf{E}[T_2|\mathscr{F}_k] \leqslant c \nabla f(x^k)^T Z \, \nabla f(x^k) = c \|\nabla f(x^k)\|_Z^2. \tag{3.12}$$

In conclusion, taking expectation with respect to $\mathscr{F}_k$ in (3.10) we have:

$$\begin{aligned}\mathbf{E}[r_{k+1}^2|\mathscr{F}_k] &= r_k^2 - 2\,\mathbf{E}[T_1|\mathscr{F}_k] + \mathbf{E}[T_2|\mathscr{F}_k] \overset{(3.11)+(3.12)}{\leqslant} r_k^2 - 2(x^k - x^*)^T\nabla f(x^k) + c\|\nabla f(x^k)\|_Z^2 \\ &\overset{(3.6)}{\leqslant} r_k^2 + 2c\left(f(x^k) - \mathbf{E}[f(x^{k+1})|\mathscr{F}_k]\right) - 2(f(x^k) - f^*),\end{aligned}$$

where in the last inequality we also used convexity of f. Taking now expectation over the entire history, we have:

$$\begin{aligned}c\,\mathbf{E}[f(x^{k+1}) - f^*] &\leqslant \mathbf{E}[\frac{1}{2}r_{k+1}^2 + c(f(x^{k+1}) - f^*)] \leqslant \mathbf{E}[\frac{1}{2}r_k^2 + c(f(x^k) - f^*)] - \mathbf{E}[f(x^k) - f^*] \\ &\leqslant \frac{1}{2}r_0^2 + c(f(x^0) - f^*) - \sum_{l=0}^{k}\mathbf{E}[f(x^l) - f^*].\end{aligned}$$

By utilizing the monotonic decrease of the expected objective values, see (3.6), we get:

$$c\,\mathbf{E}[f(x^{k+1}) - f^*] \leqslant \frac{1}{2}r_0^2 + c(f(x^0) - f^*) - (k+1)\,\mathbf{E}[f(x^{k+1}) - f^*],$$

which leads to our convergence estimate (3.9). □

REMARK 3.1 Note that for coordinate descent methods the convergence proofs in the smooth convex case are usually given in terms of a constant $\mathscr{R}(x^0) = \max_{\{x\in x^0+\ker(A):f(x)\leqslant f(x^0)\}}\ \min_{x^*\in X^*}\|x - x^*\|_Z^*$, which it is assumed to be bounded. For example, in Nesterov (2012); Necoara & Patrascu (2014) convergence rates of order $\mathscr{O}(\mathscr{R}^2(x^0)/k)$ are derived. However, the new proof of Theorem 3.2 does not depend on such $\mathscr{R}(x^0)$, which can be arbitrarily large, but it is given in terms of the parameter c, which basically depends on the properties of the objective function (M) and probability distribution on the sketching matrix (Z).

3.4 *Convergence rate: smooth strongly convex case*

In addition to the smoothness assumption, we now assume that the function f is strongly convex with respect to the extended norm $\|\cdot\|_Z^*$ with strong convexity parameter $\sigma_Z > 0$ on the subspace $x^0 + \ker(A)$:

ASSUMPTION 3.3 We assume that the objective function f is strongly convex on the subspace $x^0 + \ker(A)$, that is there exists a parameter $\sigma_Z > 0$ satisfying the following inequality:

$$f(x) \geqslant f(y) + \langle\nabla f(y), x - y\rangle + \frac{\sigma_Z}{2}\left(\|x - y\|_Z^*\right)^2 \quad \forall x, y \in x^0 + \ker(A). \tag{3.13}$$

Note that if f is strongly convex function everywhere in $\mathbb{R}^n$, that is there exists a positive definite matrix G such that:

$$f(x) \geqslant f(y) + \langle\nabla f(y), x - y\rangle + \frac{1}{2}(y - x)^T G(y - x) \quad \forall x, y \in \mathbb{R}^n, \tag{3.14}$$

then using the definition of the dual norm $(\|x\|_Z^*)^2 = x^T Z^\dagger x$ (see Lemma 2.3) we have that (3.13) also holds for some σ_Z satisfying:

$$G \succeq \sigma_Z Z^\dagger \quad \text{on} \quad \ker(A)\ \ (\text{or equivalently on } \mathbb{R}^n \setminus \text{range}(A^T)).$$

Since $x^T Z x = 0$ for all $x \in \text{range}(A^T)$ (see Lemma 2.1), then also $x^T Z^\dagger x = 0$ for all $x \in \text{range}(A^T)$. In conclusion, the matrix inequality $G \succeq \sigma_Z Z^\dagger$ holds automatically on $\text{range}(A^T)$ for any constant σ_Z, and consequently we can define σ_Z as the largest positive constant satisfying everywhere on $\mathbb{R}^n$ the matrix inequality:

$$G \succeq \sigma_Z Z^\dagger.$$

This shows that Assumption 3.3 is less restrictive than requiring strong convexity for f everywhere in $\mathbb{R}^n$ as in (3.14). Next, we prove that the strong convexity parameter σ_Z is bounded from above:

LEMMA 3.1 Under Assumptions 2.1, 2.2 and 3.3 the strong convexity parameter σ_Z defined in (3.13) is bounded above by:

$$\sigma_Z \leqslant \lambda_{\max}(M^{1/2} Z M^{1/2}) \leqslant 1. \tag{3.15}$$

Proof. By the Lipschitz continuous gradient inequality (see Assumption 2.1) and the strong convexity inequality (see Assumptions 3.3) we have that $\sigma_Z Z^\dagger \preceq M$ on $\ker(A)$ (or equivalently on $\mathbb{R}^n \setminus \text{range}(A^T)$). Since $x^T Z x = 0$ for all $x \in \text{range}(A^T)$ (see Lemma 2.1), then also $x^T Z^\dagger x = 0$ for all $x \in \text{range}(A^T)$. In conclusion, the matrix inequality $\sigma_Z Z^\dagger \preceq M$ holds automatically on $\text{range}(A^T)$. Therefore, we get the following matrix inequality valid on $\mathbb{R}^n$:

$$\sigma_Z Z^\dagger \preceq M.$$

Pre- and post-multiplying the previous matrix inequality with $ZM^{1/2}$ leads to:

$$\sigma_Z M^{1/2} Z Z^\dagger Z M^{1/2} \preceq M^{1/2} Z M Z M^{1/2},$$

or equivalently

$$\sigma_Z M^{1/2} (Z Z^\dagger Z) M^{1/2} \preceq M^{1/2} Z (M^{1/2} M^{1/2}) Z M^{1/2}.$$

Using the basic properties of the pseudo-inverse we obtain:

$$\sigma_Z \, M^{1/2} Z M^{1/2} \preceq (M^{1/2} Z M^{1/2})(M^{1/2} Z M^{1/2}).$$

Therefore, if we denote by $\Upsilon = M^{1/2} Z M^{1/2} \succeq 0$, then we get that $\Upsilon^2 - \sigma_Z \Upsilon \succeq 0$ and thus for any eigenvalue λ of Υ it holds that $\lambda^2 - \sigma_Z \lambda \geqslant 0$ or equivalently $\sigma_Z \leqslant \lambda$. It remains to show that $\lambda_{\max}(\Upsilon) \leqslant 1$. For this, we recall that according to Lemma 2.1 we know that $Z_S = Z_S M Z_S$. By utilizing the fact that M is symmetric and positive-definite, we can notice that

$$M^{1/2} Z_S M^{1/2} = M^{1/2} Z_S M Z_S M^{1/2} = (M^{1/2} Z_S M^{1/2})(M^{1/2} Z_S M^{1/2}).$$

Therefore, all the eigenvalues of $M^{1/2} Z_S M^{1/2}$ belongs to the set $\{0, 1\}$. Further, by the definition of Z in (2.8) and using the convexity of the function $\lambda_{\max}$ on the set of positive semidefinite matrices, we have:

$$\lambda_{\max}(M^{1/2} Z M^{1/2}) = \lambda_{\max}(\mathbf{E}_S[M^{1/2} Z_S M^{1/2}]) \leqslant \mathbf{E}_S[\lambda_{\max}(M^{1/2} Z_S M^{1/2})] \leqslant 1,$$

which completes our proof. □

We now derive a linear convergence estimate for our algorithm RSD under this additional strong convexity assumption on the subspace $x^0 + \ker(A)$:

THEOREM 3.4 Under Assumptions 2.1, 2.2 and 3.3 the sequence generated by RSD satisfies the following linear convergence rate for the expected value of the objective function:

$$\phi_k - f^* \leqslant (1 - \sigma_Z)^k \left(f(x^0) - f^* \right). \tag{3.16}$$

Proof. Given x^k, taking the conditional expectation in (3.6) over the random matrix S leads to the following inequality:

$$2 \left(f(x^k) - E\left[f(x^{k+1}) \mid x^k \right] \right) \geqslant \|\nabla f(x^k)\|_Z^2. \tag{3.17}$$

On the other hand, consider the minimization of the right hand side in (3.13) over $x \in x^0 + \ker(A)$, and denote $x(y)$ its optimal solution. Using the definition of the dual norm $\|\cdot\|_Z^*$ in the subspace $\ker(A)$, one can see that $x(y)$ satisfies the following optimality conditions:

$$\exists \mu \text{ s.t.: } \quad \nabla f(y) + \sigma_Z Z^\dagger (x(y) - y) + A^T \mu = 0 \quad \text{and} \quad x(y) \in x^0 + \ker(A).$$

Combining these optimality conditions with the well-known property of the pseudo-inverse, that is $Z^\dagger Z Z^\dagger = Z^\dagger$, we get that the optimal value of this minimization problem has the following expression:

$$f(y) - \frac{1}{2\sigma_Z} \|\nabla f(y)\|_Z^2.$$

Therefore, minimizing both sides of inequality (3.13) over $x \in x^0 + \ker(A)$, we have:

$$\|\nabla f(y)\|_Z^2 \geqslant 2\sigma_Z(f(y) - f^*) \quad \forall y \in x^0 + \ker(A) \tag{3.18}$$

and for $y = x^k$ we get:

$$\|\nabla f(x^k)\|_Z^2 \geqslant 2\sigma_Z \left(f(x^k) - f^*\right).$$

Combining the inequality (3.17) with the previous one, and taking expectation in $\mathscr{F}_{k-1}$ on both sides, we arrive at the statement of the theorem. □

From the proof of Theorem 3.4 it follows that we can further relax the strong convexity assumption, that is instead of (3.13) it is sufficient to require (3.18) to hold on $x^0 + \ker(A)$. The reader should note that an inequality of the form (3.18) is known in the optimization literature as the Polyak-Lojasiewicz (PL) condition (see e.g. Karimi *et al.* (2016) for a recent exposition), and the proof above shows that algorithm RSD converges linearly for smooth convex functions satisfying only the PL condition. Since functions satisfying the PL inequality need not be convex, linear convergence of RSD method to the global optimum extends beyond the realm of convex functions. More precisely, is is easy to see that the convergence result of Theorem 3.1 can be strengthen, that is we can prove linear convergence in the expected values of the objective function for the iterates of algorithm RSD provided that additionally the PL type condition (3.18) holds (we just need to combine the inequalities (3.8) and (3.18)).

REMARK 3.2 Note that in special cases, where complexity bounds are known for particular RSD, such as 2-coordinate descent for optimization problems with a single linear coupled constraint, our theory recovers the best known bounds or leads to better bounds (see e.g. the convergence analysis in Frongillo & Reid (2015); Necoara & Patrascu (2014); Necoara *et al.* (2017)). Note that the convergence rates of coordinate descent methods for the smooth convex case are usually given in terms of a constant defined as $\mathscr{R}(x^0) = \max_{\{x \in x^0 + \ker(A): f(x) \leqslant f(x^0)\}} \min_{x^* \in X^*} \|x - x^*\|_Z^*$, which is assumed to be bounded. For example, in Frongillo & Reid (2015); Nesterov (2012); Necoara & Patrascu (2014); Necoara *et al.* (2017) convergence rates of order $\mathscr{O}(\mathscr{R}^2(x^0)/k)$ are derived for coordinate descent type methods. Similarly, for the strongly convex case, choosing for the sketch matrix S any 2 columns of the identity matrix, then combining (2.12) with Theorem 3.4 we recover the convergence rate of 2-coordinate descent algorithm from (Necoara *et al.*, 2017, Theorem 4.2) for the problem with a separable objective function and a single linear constraint. *In conclusion, to our knowledge, this is the first complete convergence analysis of a general random sketch descent (RSD) algorithm, for which coordinate descent method is a particular case, for solving optimization problems with multiple linear coupled constraints.*

4. Accelerated random sketch descent algorithm

For the accelerated variant of Algorithm RSD let us first define the following constant:

$$\nu_{\max} = \max_{u \in \ker(A), u \neq 0} \frac{\mathbf{E}[(\|Z_S u\|_Z^*)^2]}{\|u\|_Z^2} = \max_{u \in \ker(A), u \neq 0} \frac{\mathbf{E}[\|Z_S u\|_{Z^\dagger}^2]}{\|u\|_Z^2}. \tag{4.1}$$

Let us now consider any constant parameter $\nu \geqslant \nu_{\max}$. The Accelerated Random Sketch Descent (A-RSD) scheme is depicted in Algorithm 2:

4.1 *Computation cost per-iteration for A-RSD*

It is easy to observe that the computational cost for updating the sequence x^k is comparable to the one corresponding to RSD algorithm. Therefore, the conclusions regarding the cost per-iteration from Section 3.1 corresponding to RSD are also valid here. Note that the accelerated variant also requires updating two additional sequences y^k and v^k, which requires computations with full vectors in $\mathbb{R}^n$. However, for structured optimization problems we can avoid the addition of full vectors in $\mathbb{R}^n$ and still keep the cost per-iteration of A-RSD comparable to that of RSD. More precisely, we can efficiently implement the updates

Algorithm 2 Algorithm A-RSD

Input: Positive sequences $\{\alpha_k\}_{k=0}^{\infty}, \{\beta_k\}_{k=0}^{\infty}, \{\gamma_k\}_{k=0}^{\infty}$
Choose $x^0 \in \mathbb{R}^n$ such that $Ax^0 = b$ and set $v^0 = x^0$
for $k \geqslant 0$ **do**
  sample $S \sim \mathscr{S}$ and perform the following updates:
  $y^k = \alpha_k v^k + (1-\alpha_k)x^k$
  $x^{k+1} = y^k - Z_S \nabla f(y^k)$
  $v^{k+1} = \beta_k v^k + (1-\beta_k)y^k - \gamma_k Z_S \nabla f(y^k)$
end for

of A-RSD algorithm without full-dimensional vector operations when the sketch matrix S is sparse and when we can efficiently compute:

$$\nabla f(\alpha v + \beta u) \qquad \forall \alpha, \beta \in \mathbb{R} \text{ and } v, u \in \mathbb{R}^n.$$

Note that gradient evaluation in such points is computationally easy when f has a special structure, e.g. of the form $f(x) = g(Ex)$, where E is a sparse matrix Fercoq & Richtárik (2015). Objective functions of this form includes many generalized linear models, such as logistic regression, least squares, etc. In Appendix we provide efficient implementations of the updates of A-RSD for these settings.

4.2 *Basic properties of A-RSD*

Before deriving convergence rates for A-RSD we analyze some basic properties of this algorithm. First, we prove that the newly introduced constant $\nu_{\max}$ is bounded, thus finite:

LEMMA 4.1 Under Assumptions 2.1, 2.2 and 3.3 we have:

$$\sigma_Z \leqslant \nu_{\max} \leqslant \lambda_{\max}(M^{-1/2} Z^\dagger M^{-1/2}) \leqslant \frac{1}{\sigma_Z}.$$

Proof. If we denote $c = \lambda_{\max}(M^{-1/2} Z^\dagger M^{-1/2})$, then it follows that $Z^\dagger \preceq cM$. Using this matrix inequality in the definition of $\nu_{\max}$ we have:

$$\frac{\mathbf{E}[\|Z_S u\|^2_{Z^\dagger}]}{\|u\|^2_Z} = \frac{\mathbf{E}[u^T Z_S Z^\dagger Z_S u]}{u^T Z u} \leqslant \frac{\mathbf{E}[c \cdot u^T Z_S M Z_S u]}{u^T Z u}$$
$$= c \frac{\mathbf{E}[u^T Z_S u]}{u^T Z u} = c \quad \forall u \in \ker(A), u \neq 0.$$

This proves that $\nu_{\max} \leqslant c < \infty$ provided that Assumptions 2.1 and 2.2 hold (there is no need to impose strong convexity for this upper bound). Now, we will show that $\sigma_Z \leqslant \nu_{\max}$ if additionally strong convexity (Assumption 3.3) holds. Indeed, from Jensen inequality we have:

$$\nu_{\max} = \max_{u \in \ker(A), u \neq 0} \frac{\mathbf{E}[\|Z_S u\|^2_{Z^\dagger}]}{\|u\|^2_Z} \geqslant \max_{u \in \ker(A), u \neq 0} \frac{\|\mathbf{E}[Z_S] u\|^2_{Z^\dagger}}{\|u\|^2_Z}$$
$$= \max_{u \in \ker(A), u \neq 0} \frac{\|Zu\|^2_{Z^\dagger}}{\|u\|^2_Z} = \max_{u \in \ker(A), u \neq 0} \frac{\|u\|^2_Z}{\|u\|^2_Z} = 1 \overset{(3.15)}{\geqslant} \sigma_Z,$$

proving the left-hand side inequality. Moreover, by the Lipschitz continuous gradient inequality and the strong convexity inequality we have $\sigma_Z Z^\dagger \preceq M$ and consequently $M^{-1/2} Z^\dagger M^{-1/2} \preceq \frac{1}{\sigma_Z} I_n$. Hence, $\lambda_{\max}(M^{-1/2} Z^\dagger M^{-1/2}) \leqslant \frac{1}{\sigma_Z}$, which concludes the proof. □

EXAMPLE 4.1 (Cont. of example 2.3.) For the optimization problem considered in Example 2.3 we can easily compute a good upper approximation for $\nu_{\max}$:

$$\nu_{\max} = \max_{u\in\ker(A),u\neq 0} \frac{\mathbf{E}[u^T Z_{(i,j)} Z^\dagger Z_{(i,j)} u]}{u^T Z u} = \max_{u\in\ker(A),u\neq 0} \frac{\mathbf{E}\left[\frac{2(n-1)L}{n(L_i+L_j)} u^T Z_{(i,j)} u\right]}{u^T Z u} \leqslant \max_{i<j} \frac{2(n-1)L}{n(L_i+L_j)}, \tag{4.2}$$

where we used $(e_i-e_j)^T(I_n - 1/n\ ee^T)(e_i-e_j) = 2$. This shows that $\nu_{\max} \leqslant \frac{\sum_i L_i}{\min_i L_i}$ $(:\backsimeq n)$ and consequently it is related to the dimension of the optimization problem (see also Nesterov (2012)).

For simplicity of the exposition let us also denote:

$$g^k = -Z_S \nabla f(y^k) \quad \left(= -SP_S(P_S^T S^T M S P_S)^\dagger P_S^T S^T \nabla f(y^k)\right).$$

From the updates of A-RSD we can also show a descent property for the conditional expectation $\mathbf{E}[f(x^{k+1})|\mathscr{F}_k]$. Indeed, from our updates and Assumption 2.1 we have:

$$f(x^{k+1}) = f(y^k + g_k) \leqslant f(y^k) + \langle \nabla f(y^k), g_k\rangle + \frac{1}{2}\|g_k\|_M^2.$$

Taking now the conditional expectation with respect to random choice S and using that $Z_S M Z_S = Z_S$ (see Lemma 2.1) we obtain:

$$\begin{aligned}
\mathbf{E}[f(x^{k+1})|\mathscr{F}_k] &\leqslant f(y_k) + \langle \nabla f(y_k), \mathbf{E}[g_k|\mathscr{F}_k]\rangle + \frac{1}{2}\mathbf{E}[\|g_k\|_M^2|\mathscr{F}_k] \\
&= f(y_k) + \langle \nabla f(y_k), \mathbf{E}[-Z_S\nabla f(y^k)|\mathscr{F}_k]\rangle + \frac{1}{2}\mathbf{E}[\|Z_S\nabla f(y^k)\|_M^2|\mathscr{F}_k] \\
&= f(y_k) - \|\nabla f(y_k)\|_Z^2 + \frac{1}{2}\|\nabla f(y^k)\|_Z^2 = f(y_k) - \frac{1}{2}\|\nabla f(y_k)\|_Z^2.
\end{aligned} \tag{4.3}$$

Moreover, the sequences x^k, y^k and v^k satisfies $x^k - x^* \in \ker(A)$, $y^k - x^* \in \ker(A)$ and $v^k - x^* \in \ker(A)$, and consequently also $x^k - y^k \in \ker(A)$. Moreover, since $\mathrm{range}(A^T) = \ker(Z)$ (see Lemma 2.2), then $Z^\dagger Z$ is a projection matrix onto $\ker(A)$, that is $Z^\dagger Z u = u$ for all $u \in \ker(A)$, and thus the following holds:

$$Z^\dagger Z(x^* - y^k) = x^* - y^k \quad \text{and} \quad Z^\dagger Z(x^k - y^k) = x^k - y^k. \tag{4.4}$$

For any optimal point x^* let us also define the sequence: $r_k^2 = \|v^k - x^*\|_{Z^\dagger}^2$. Based on the previous discussion, we can show the following descent property for the sequence r_k generated by Algorithm A-RSD that holds also for the case $\sigma_Z = 0$:

LEMMA 4.2 Under Assumptions 2.1, 2.2 and 3.3, for any choices of the sequences $\{\alpha_k\}_{k=0}^\infty \in (0,\ 1]$, $\{\beta_k\}_{k=0}^\infty \in (0,\ 1]$ and $\{\gamma_k\}_{k=0}^\infty \in (0,\ \infty)$, and any $\nu \geqslant \nu_{\max}$ the Algorithm A-RSD produces a sequence of points (x_k, y_k, v_k) such that the following descent inequality holds:

$$\begin{aligned}
\mathbf{E}[r_{k+1}^2 + 2\gamma_k^2\nu(f(x^{k+1}) - f^*)|\mathscr{F}_k] &\leqslant \beta_k\left(r_k^2 + 2\gamma_k\frac{1-\alpha_k}{\alpha_k}(f(x^k) - f^*)\right) \\
&+ (1-\beta_k - \gamma_k\sigma_Z)\|y^k - x^*\|_{Z^\dagger}^2 + \left(2\gamma_k^2\nu - 2\gamma_k - 2\gamma_k\beta_k\frac{1-\alpha_k}{\alpha_k}\right)(f(y^k) - f^*).
\end{aligned} \tag{4.5}$$

Proof. Using the definition of r_{k+1} we have:

$$\begin{aligned}
r_{k+1}^2 &= \|v^{k+1} - x^*\|_{Z^\dagger}^2 = \|\beta_k v^k + (1-\beta_k)y^k - x^* + \gamma_k g_k\|_{Z^\dagger}^2 \\
&= \|\beta_k v^k + (1-\beta_k)y^k - x^*\|_{Z^\dagger}^2 + \gamma_k^2\|g_k\|_{Z^\dagger}^2 + 2\gamma_k\left(\beta_k v^k + (1-\beta_k)y^k - x^*\right)^T Z^\dagger g_k \\
&\leqslant \beta_k\|v^k - x^*\|_{Z^\dagger}^2 + (1-\beta_k)\|y^k - x^*\|_{Z^\dagger}^2 + \gamma_k^2\|g_k\|_{Z^\dagger}^2 + 2\gamma_k\left(\beta_k v^k + (1-\beta_k)y^k - x^*\right)^T Z^\dagger g_k,
\end{aligned}$$

where in the last inequality we used the convexity of the norm and the fact that $\beta_k \in [0,\ 1]$. Taking now the conditional expectation with respect to $\mathscr{F}_k$ we get:

$$\begin{aligned}
\mathbf{E}[r_{k+1}^2|\mathscr{F}_k] &\leqslant \beta_k\|v^k - x^*\|_{Z^\dagger}^2 + (1-\beta_k)\|y^k - x^*\|_{Z^\dagger}^2 + \gamma_k^2\,\mathbf{E}[\|-Z_S\nabla f(y^k)\|_{Z^\dagger}^2|\mathscr{F}_k] \\
&\quad + 2\gamma_k\left(\beta_k v^k + (1-\beta_k)y^k - x^*\right)^T Z^\dagger(-Z\nabla f(y^k)) \\
&\overset{(4.1)}{\leqslant} \beta_k r_k^2 + (1-\beta_k)\|y^k - x^*\|_{Z^\dagger}^2 + \gamma_k^2\nu\|\nabla f(y^k)\|_Z^2 + 2\gamma_k\left(x^* - \beta_k v^k - (1-\beta_k)y^k\right)^T Z^\dagger Z\nabla f(y^k) \\
&= \beta_k r_k^2 + (1-\beta_k)\|y^k - x^*\|_{Z^\dagger}^2 + \gamma_k^2\nu\|\nabla f(y^k)\|_Z^2 \\
&\quad + 2\gamma_k\left(x^* - y^k\right)Z^\dagger Z\nabla f(y^k) - 2\gamma_k\beta_k\left(v^k - y^k\right)^T Z^\dagger Z\nabla f(y^k) \\
&= \beta_k r_k^2 + (1-\beta_k)\|y^k - x^*\|_{Z^\dagger}^2 + \gamma_k^2\nu\|\nabla f(y^k)\|_Z^2 \\
&\quad + 2\gamma_k\left(x^* - y^k\right)Z^\dagger Z\nabla f(y^k) - 2\gamma_k\beta_k\frac{1-\alpha_k}{\alpha_k}\left(y^k - x^k\right)^T Z^\dagger Z\nabla f(y^k) \\
&\overset{(4.3)}{\leqslant} \beta_k r_k^2 + (1-\beta_k)\|y^k - x^*\|_{Z^\dagger}^2 + 2\gamma_k^2\nu\left(f(y^k) - \mathbf{E}[f(x^{k+1})|x^k]\right) \\
&\quad + 2\gamma_k\left(x^* - y^k\right)Z^\dagger Z\nabla f(y^k) - 2\gamma_k\beta_k\frac{1-\alpha_k}{\alpha_k}\left(y^k - x^k\right)^T Z^\dagger Z\nabla f(y^k).
\end{aligned}$$

Rearranging the terms, we get:

$$\begin{aligned}
&\mathbf{E}[r_{k+1}^2 + 2\gamma_k^2\nu(f(x^{k+1}) - f^*)|x^k] \\
&\leqslant \beta_k r_k^2 + (1-\beta_k)\|y^k - x^*\|_{Z^\dagger}^2 + 2\gamma_k^2\nu(f(y^k) - f^*) \\
&\quad + 2\gamma_k\left(x^* - y^k\right)^T Z^\dagger Z\nabla f(y^k) + 2\gamma_k\beta_k\frac{1-\alpha_k}{\alpha_k}\left(x^k - y^k\right)^T Z^\dagger Z\nabla f(y^k) \\
&\overset{(4.4)}{\leqslant} \beta_k r_k^2 + (1-\beta_k)\|y^k - x^*\|_{Z^\dagger}^2 + 2\gamma_k^2\nu(f(y^k) - f^*) \\
&\quad + 2\gamma_k\left(x^* - y^k\right)^T \nabla f(y^k) + 2\gamma_k\beta_k\frac{1-\alpha_k}{\alpha_k}\left(x^k - y^k\right)^T \nabla f(y^k) \\
&\overset{(3.13)}{\leqslant} \beta_k r_k^2 + (1-\beta_k)\|y^k - x^*\|_{Z^\dagger}^2 + 2\gamma_k^2\nu(f(y^k) - f^*) \\
&\quad + 2\gamma_k\left(f^* - f(y^k) - \frac{\sigma_Z}{2}\|y^k - x^*\|_{Z^\dagger}^2\right) + 2\gamma_k\beta_k\frac{1-\alpha_k}{\alpha_k}\left(x^k - y^k\right)^T \nabla f(y^k).
\end{aligned}$$

Note that the previous derivations also hold without Assumption 3.3, that is we use the strong convexity inequality (3.13) with $\sigma_Z = 0$. Using now the convexity of function f and that $\alpha_k \in (0,\ 1]$, we further get:

$$\begin{aligned}
&\mathbf{E}[r_{k+1}^2 + 2\gamma_k^2\nu(f(x^{k+1}) - f^*)|x^k] \\
&\leqslant \beta_k r_k^2 + (1-\beta_k-\gamma_k\sigma_Z)\|y^k - x^*\|_{Z^\dagger}^2 + (2\gamma_k^2\nu - 2\gamma_k)(f(y^k) - f^*) + 2\gamma_k\beta_k\frac{1-\alpha_k}{\alpha_k}(f(x^k) - f(y^k)) \\
&= \beta_k\left(r_k^2 + 2\gamma_k\frac{1-\alpha_k}{\alpha_k}(f(x^k) - f^*)\right) + (1-\beta_k-\gamma_k\sigma_Z)\|y^k - x^*\|_{Z^\dagger}^2 \\
&\quad + \left(2\gamma_k^2\nu - 2\gamma_k - 2\gamma_k\beta_k\frac{1-\alpha_k}{\alpha_k}\right)(f(y^k) - f^*),
\end{aligned}$$

which concludes our statement. □

Based on the previous descent property we can derive different convergence rates for our algorithm A-RSD depending on the assumptions imposed on the objective function f.

4.3 *Convergence rate: smooth convex case*

In this section we prove the sublinear convergence rate for A-RSD (Algorithm 2) for some choices of the sequences $\{\alpha_k\}_{k=0}^{\infty}$, $\{\beta_k\}_{k=0}^{\infty}$ and $\{\gamma_k\}_{k=0}^{\infty}$. In particular, the next lemma shows the behavior of $\{\gamma_k\}_{k=0}^{\infty}$ defined as follows:

LEMMA 4.3 Let $\{\gamma_k\}_{k=0}^{\infty}$ be a sequence defined recursively as $\gamma_0 = \frac{1}{\nu}$ and γ_{k+1} be the largest solution of the second order equation:

$$\gamma_{k+1}^2 - \tfrac{1}{\nu}\gamma_{k+1} = \gamma_k^2. \tag{4.6}$$

Then, γ_k satisfies the following inequality:

$$\gamma_k \geqslant \frac{k+2}{2\nu}. \tag{4.7}$$

Proof. First, we observe that $\{\gamma_k\}_{k=0}^{\infty}$ is a non-decreasing sequence. Indeed, the largest root of (4.6) is:

$$\gamma_{k+1} = \frac{\frac{1}{\nu} + \sqrt{\frac{1}{\nu^2} + 4\gamma_k^2}}{2} \geqslant \frac{\sqrt{4\gamma_k^2}}{2} = \gamma_k. \tag{4.8}$$

Next, we have:

$$\tfrac{1}{\nu}\gamma_{k+1} \overset{(4.6)}{=} \gamma_{k+1}^2 - \gamma_k^2 = (\gamma_{k+1} - \gamma_k)(\gamma_{k+1} + \gamma_k) \overset{(4.8)}{\leqslant} 2\gamma_{k+1}(\gamma_{k+1} - \gamma_k), \tag{4.9}$$

which implies that

$$\gamma_k + \tfrac{1}{2\nu} \leqslant \gamma_{k+1} \quad \Rightarrow \quad \gamma_k \geqslant \gamma_0 + k\frac{1}{2\nu} = \frac{2+k}{2\nu}.$$

This concludes our proof. □

From (4.7) it follows $\gamma_k \nu \geqslant 1, \forall k \geqslant 0$. Now, we are ready to prove the sublinear convergence of A-RSD:

THEOREM 4.2 Under Assumptions 2.1 and 2.2 the sequences generated by Algorithm A-RSD with $\alpha_k = \frac{1}{\gamma_k \nu} \in (0,\ 1]$, $\beta_k = 1$, $\gamma_0 = \frac{1}{\nu}$ and γ_k be the largest solution defined by recursion (4.6), satisfy the following sublinear convergence rate in expectation:

$$\mathbf{E}\left[f(x^k) - f^*\right] \leqslant \frac{2\nu}{(k+1)^2} \min_{x^* \in X^*} \|x^0 - x^*\|_{Z^\dagger}^2 \quad \forall k \geqslant 1.$$

Proof. In the smooth convex case we can use Lemma 4.2 (i.e. descent relation (4.5)) by setting $\sigma_Z = 0$:

$$\begin{aligned}\mathbf{E}[r_{k+1}^2 + 2\gamma_k^2 \nu (f(x^{k+1}) - f^*) | \mathscr{F}_k] \overset{(4.5)}{\leqslant} &\left(r_k^2 + 2\gamma_k \frac{1-\alpha_k}{\alpha_k}(f(x^k) - f^*)\right) \\ &+ \left(2\gamma_k^2 \nu - 2\gamma_k \frac{1}{\alpha_k}\right)(f(y^k) - f^*).\end{aligned} \tag{4.10}$$

Note that $\alpha_k = \frac{1}{\gamma_k \nu}$ and hence the last term in (4.10) vanishes. Thus, we further obtain:

$$\mathbf{E}[r_{k+1}^2 + 2\gamma_k^2 \nu (f(x^{k+1}) - f^*) | \mathscr{F}_k] \overset{(4.10)}{\leqslant} r_k^2 + 2\gamma_k \frac{1-\alpha_k}{\alpha_k}(f(x^k) - f^*). \tag{4.11}$$

Moreover, since $\alpha_k = \frac{1}{\gamma_k \nu}$, then:

$$2\gamma_k \frac{1-\alpha_k}{\alpha_k} = 2\gamma_k^2 \nu \left(1 - \frac{1}{\gamma_k \nu}\right) = 2\gamma_k^2 \nu - 2\gamma_k. \tag{4.12}$$

Plugging (4.12) into (4.11) and dividing both sides by 2ν we obtain:

$$\mathbf{E}[\tfrac{1}{2\nu}r_{k+1}^2+\gamma_k^2(f(x^{k+1})-f^*)|\mathscr{F}_k]\leqslant\left(\tfrac{1}{2\nu}r_k^2+(\gamma_k^2-\tfrac{1}{\nu}\gamma_k)(f(x^k)-f^*)\right). \tag{4.13}$$

Now, it reminds to note that γ_{k+1} satisfy (4.6) and consequently:

$$\mathbf{E}[\tfrac{1}{2\nu}r_{k+1}^2+(\gamma_{k+1}^2-\tfrac{1}{\nu}\gamma_{k+1})(f(x^{k+1})-f^*)|\mathscr{F}_k]\leqslant\left(\tfrac{1}{2\nu}r_k^2+(\gamma_k^2-\tfrac{1}{\nu}\gamma_k)(f(x^k)-f^*)\right).$$

Taking now the expectation over the entire history in the previous recursion and unrolling it, we get:

$$\begin{aligned}\mathbf{E}[(\gamma_k^2-\tfrac{1}{\nu}\gamma_k)(f(x^k)-f^*)]&\leqslant\mathbf{E}[\tfrac{1}{2\nu}r_k^2+(\gamma_k^2-\tfrac{1}{\nu}\gamma_k)(f(x^k)-f^*)]\\&\leqslant\left(\tfrac{1}{2\nu}r_0^2+(\gamma_0^2-\tfrac{1}{\nu}\gamma_0)(f(x^0)-f^*)\right).\end{aligned}$$

Since the second order equation $\gamma_k^2-\frac{1}{\nu}\gamma_k=\gamma_{k-1}^2\overset{(4.7)}{\geqslant}\left(\frac{k+1}{2\nu}\right)^2$ for all $k\geqslant 1$, we get our statement. □

4.4 *Convergence rate: smooth strongly convex case*

We are now ready to state the linear convergence rate for A-RSD (Algorithm 2).

THEOREM 4.3 Under Assumptions 2.1, 2.2 and 3.3 the sequences generated by Algorithm A-RSD with $\alpha_k=\frac{\gamma_k\sigma_Z}{1+\gamma_k\sigma_Z}\in(0,\,1]$, $\beta_k=1-\gamma_k\sigma_Z\in[0,\,1]$ and $\gamma_k=\frac{1}{\sqrt{\sigma_Z\nu}}\leqslant\frac{1}{\sigma_Z}$ satisfy the following linear convergence rate in expectation:

$$\mathbf{E}\left[r_k^2+\frac{2}{\sigma_Z}(f(x^k)-f^*)\right]\leqslant\left(1-\sqrt{\frac{\sigma_Z}{\nu}}\right)^k\left(r_0^2+\frac{2}{\sigma_Z}(f(x^0)-f^*)\right).$$

Proof. Note that the choices of α_k,β_k and γ_k from the theorem guarantee that:

$$2\gamma_k^2\nu=2\gamma_k\frac{1-\alpha_k}{\alpha_k},\quad 1-\beta_k-\gamma_k\sigma_Z\leqslant 0,\quad 2\gamma_k^2\nu-2\gamma_k-2\gamma_k\beta_k\frac{1-\alpha_k}{\alpha_k}=0.$$

Using these relations in Lemma 4.2 (i.e. descent relation (4.5)), we get:

$$\mathbf{E}[r_{k+1}^2+2\gamma_k^2\nu(f(x^{k+1})-f^*)|\mathscr{F}_k]\overset{(4.5)}{\leqslant}\beta_k\left(r_k^2+2\gamma_k\frac{1-\alpha_k}{\alpha_k}(f(x^k)-f^*)\right).$$

After plugging $\alpha_k=\frac{\gamma_k\sigma_Z}{1+\gamma_k\sigma_Z}$ and $\gamma_k=\frac{1}{\sqrt{\sigma_Z\nu}}$ we further obtain:

$$\mathbf{E}[r_{k+1}^2+2/\sigma_Z(f(x^{k+1})-f^*)|\mathscr{F}_k]\leqslant\beta_k\left(r_k^2+2/\sigma_Z(f(x^k)-f^*)\right),$$

and taking now the expectation over the entire history we get:

$$\mathbf{E}[r_k^2+2/\sigma_Z(f(x^k)-f^*)]\leqslant\left(\prod_{j=0}^{k-1}\beta_j\right)\left(r_0^2+2/\sigma_Z(f(x^0)-f^*)\right),$$

which leads to the statement of our theorem. □

REMARK 4.1 Note that the presence of coupling constraints requires new proof techniques for A-RSD algorithm, different from the standard convergence proofs of accelerated coordinate descent methods from the literature, e.g. the parameters α_k,β_k and γ_k are defined differently from those in Nesterov (2012). Consequently the recurrence relation from Lemma 4.2. is also different from the one in Nesterov (2012). *It is also important to note that in this work we designed for the first time an accelerated random sketch descent algorithm (A-RSD) for optimization problems with multiple non-separable linear constraints accompanied by a full convergence analysis*.

Table 1: Comparison of convergence rates for RSD and A-RSD algorithms.

	RSD	A-RSD
smooth convex	$\frac{\|x^0-x^*\|^2_{Z^\dagger}}{k}$	$\frac{\nu\|x^0-x^*\|^2_{Z^\dagger}}{k^2}$
smooth strong convex	$(1-\sigma_Z)^k$	$\left(1-\sqrt{\frac{\sigma_Z}{\nu}}\right)^k$

Table 1 summarizes the convergence rates in $\mathbf{E}[f(x^k)]-f^*$ of RSD and A-RSD algorithms for smooth (strongly) convex objective functions (we assume, for simplicity, $\|x^0-x^*\|^2_{Z^\dagger} \backsimeq c(f(x^0)-f^*)$). We observe from this table that we have obtained the typical convergence rates for these two methods, in particular, A-RSD achieves accelerated converges rates, see Nesterov (2012). Let us write the convergence rates of the algorithms RSD and A-RSD for the particular choice of sketching matrix $S_{(ij)} = [e_i\, e_j]$ for solving the optimization problem with a single linear constraint from Example 2.3. In this case, according to (4.2), we have $\nu \leqslant \frac{\sum_i L_i}{\min_i L_i}$. Hence, we get:

$$\frac{\|x^0-x^*\|^2_{Z^\dagger}}{k} \quad \text{vrs} \quad \frac{\sum_i L_i}{\min_i L_i}\cdot\frac{\|x^0-x^*\|^2_{Z^\dagger}}{k^2} \qquad \text{and} \qquad (1-\sigma_Z)^k \quad \text{vrs} \quad \left(1-\sqrt{\sigma_Z\cdot\frac{\min_i L_i}{\sum_i L_i}}\right)^k.$$

Clearly, in the smooth convex case A-RSD is superior to RSD provided that the number of iterations k is larger than $\frac{\sum_i L_i}{\min_i L_i}$. Similarly, in the strongly convex case A-RSD is superior to RSD for $\sigma_Z \leqslant \frac{\min_i L_i}{\sum_i L_i}$.

5. Illustrative numerical experiments

In this section we provide several numerical examples showing the benefits of random sketching and the performances of our new algorithms.

Experiment #1: A pre-fixed coordinate sampling can be very slow. Recently, in Tu *et al.* (2017) it has been shown for linear systems that Gauss-Seidel algorithm with randomly sampled coordinates substantially outperforms Gauss-Seidel with any fixed partitioning of the coordinates that are chosen ahead of time. Motivated by this finding, we also analyze the behavior of RSD and A-RSD algorithms for three different choices for $S \in \mathbb{R}^{n\times 2}$, fixed coordinates sketch, random coordinates sketch and Gaussian sketch:

$$\begin{aligned}\text{fixed partition of coordinates}: &\ S_{(i,i+1)} = [e_i\, e_{i+1}] \quad \forall i = 1:n-1\\ \text{random partition of coordinates}: &\ S_{(i,j)} = [e_i\, e_j] \quad \forall i<j\\ \text{Gaussian sketch}: &\ S = [\mathcal{N}(0,1)]^{n\times 2},\end{aligned}$$

where we recall that e_i denotes the ith column of the identity matrix I_n and $\mathcal{N}(0,1)$ is normally distributed random variable with mean 0 and variance 1. Here, $p=2$ and note that the index i is chosen random when using the fixed partition of coordinates matrix $S_{(i,i+1)}$. We build three challenging problems. The first problem is to minimize the following convex optimization problem parameterized by $\delta \in [0,1]$:

$$\min_{x\in\mathbb{R}^n} x^T\left(I_n+(1-\delta)(e_1e_n^T+e_ne_1^T)\right)x \qquad \text{s.t.} \quad e^Tx=0. \tag{5.1}$$

The initial iterate in this case was set $x^0=[1\ -1\cdots]^T$. For the second problem we consider:

$$\min_{x\in\mathbb{R}^n} x^TMx \qquad \text{s.t.} \quad Ax=0,$$

where $M=M_0+\delta I_n$ and $M_0 \succeq 0$ is a rank deficient random matrix. In this case we denote $\{v_k\}_{k=0}^n$ to be a set of orthogonal eigenvectors of M, such that v_1 corresponds to the largest eigenvalue and v_n is the

eigenvector which corresponds to the smallest eigenvalue. We have chosen $x^0 = v_1$ and $A = v_2^T$. The third problem is as follows:

$$\min_{x\in\mathbb{R}^n} (x_1 - x_2)^2 + (x_2 - x_3)^2 + \cdots + (x_{n-1} - x_n)^2 \qquad \text{s.t.} \quad \sum_i \frac{1}{i^2} x_i = 0.$$

The initial iterate in this case was set $x^0 = e_n$. The optimal solution for all three problems is $x^* = \mathbf{0}$ with $f(x^*) = 0$. In Figure 1 we show the results for the three problems (each column corresponds to one

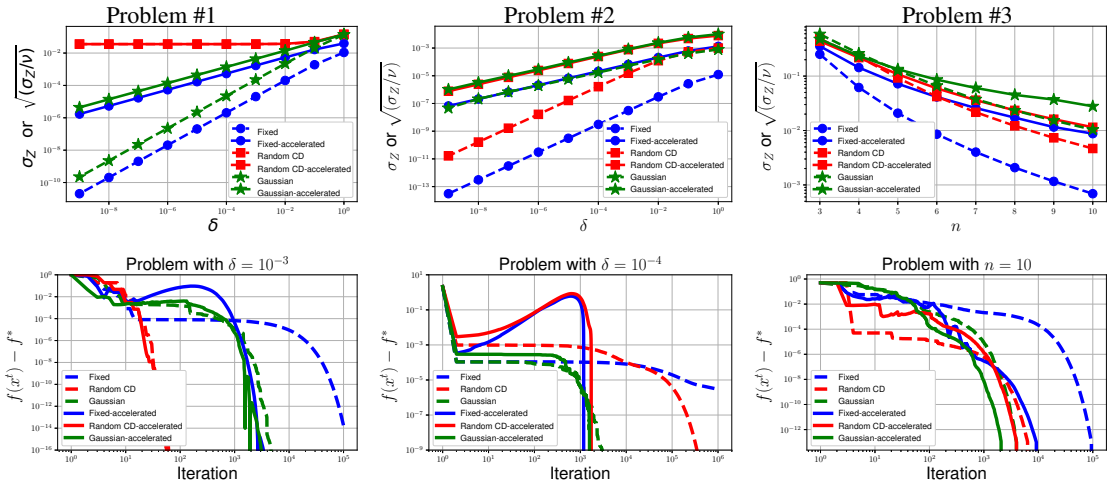

FIG. 1: Behavior of RSD and A-RSD for 3 problems and 3 different random sketch samplings.

problem). The first row shows the important quantities σ_Z or $\sqrt{\sigma_Z/\nu}$ which characterize the convergence rates of the two algorithms in the strongly convex case (see Table 1). In the bottom row we show the typical evolution of $f(x^k) - f^*$. One can observe that for the first problem, the random coordinate sampling is the best, whereas the other two samplings are suffering. The main reason is that for this problem, the most important sketch matrix S is $S = [e_1\ e_n]$ which is selected more often by the random coordinate sketching than the other two sketching strategies. For the second test problem the best results were obtained for random coordinate and Gaussian sketch. For the third problem, we can see that the Gaussian sketching is the best choice leading to the smallest number of iterations. Therefore, empirically these experiments show that both algorithms, RSD and A-RSD, based on random sketching provides speedups compared to the fixed partitioning of the coordinates.

Experiment #2: The effect of a quadratic upper-bound in convergence speed. In this experiment, we investigate the benefit of using the full matrix M in (2.6) as compared to just using a scaled diagonal upper-bound as considered e.g. in Frongillo & Reid (2015); Necoara (2013). Consider the following convex optimization problem parameterized by $\delta \in [0,1]$:

$$\min_{x\in\mathbb{R}^n} x^T \underbrace{(\delta I_n + (1-\delta) e e^T)}_{B \succeq 0} x \quad \text{s.t.} \quad e^T x = 0. \tag{5.2}$$

We compare in Figure 2 the speed of Algorithm RSD when the random matrix S is chosen uniformly at random as p columns of the identity matrix and consider three choices for the matrix M: $M = B$, $M = \lambda_{\max}(B) I_n$ and $M_S = \lambda_{\max}(S^T B S) S S^T$. We also implement RSD for the Gaussian sketch and $M = B$. From Figure 2 one can observe that if we set $M = B$, then increasing p will decrease the number of iterations needed to achieve the desired accuracy with the best rate. We can also observe that in this scenario the Gaussian sketch and random coordinate sketch have a similar behavior for $M = B$.

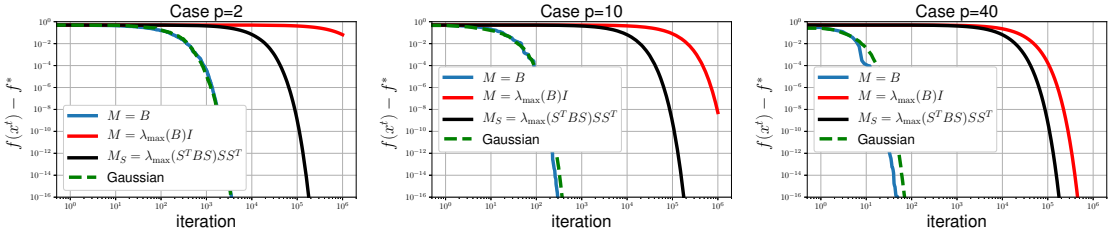

FIG. 2: Comparison of speed of Algorithm RSD for various choices of M and p.

Experiment #3: Portfolio optimization with specified industry allocation. In Section 1.2.3 we have described the basic Markowitz portfolio selection model Markowitz (1952). We have also described a variant of the basic model which assumes that investor also decide how much net wealth would be allocated in different asset classes (e.g. Financials, Health Care, Industrials, etc). When we have C asset classes, then the problem of minimizing the risk with all the desired constraints will lead to $C+2$ linear equality constraints. In Figure 3 we compare the performance of the RSD algorithm with the sketch matrix S chosen random Gaussian with the coordinate descent sketch. We consider real data from the index S&P500 which contains 500 assets split across $C = 11$ asset classes. The μ_is and Σ were estimated from the historical data. In the left plot we show the evolution of error $f(x^k) - f^*$ for various sizes of block size p as a function of iterations and in the middle plot the total computational time in seconds. We can observe that increasing p leads to a significant decrease of the number of iterations and also a faster convergence in terms of wall-clock time for the RSD with Gaussian sketch than for the coordinate descent variant. For the coordinate descent algorithm increasing p has a very slight benefit for decreasing the number of iterations, and, in terms of time, after $p = 50$ one observe a slow-down. Note also that as p is becoming larger, the per-iteration computational cost increases moderately (see the right plot in Figure 3) for both RSD with Gaussian sketch and for coordinate descent. Note that for coordinate descent implementation, for small p, each iteration is faster than the Gaussian sketching counterpart because it can benefit from the sparsity of sketch matrix S and thus many steps can be implemented efficiently. But as p is becoming larger, the computation of pseudoinverse becomes the dominant cost for both implementations and consequently the per-iteration cost is identical for the two schemes. However, the RSD with Gaussian sketch requires significantly less number of iterations.

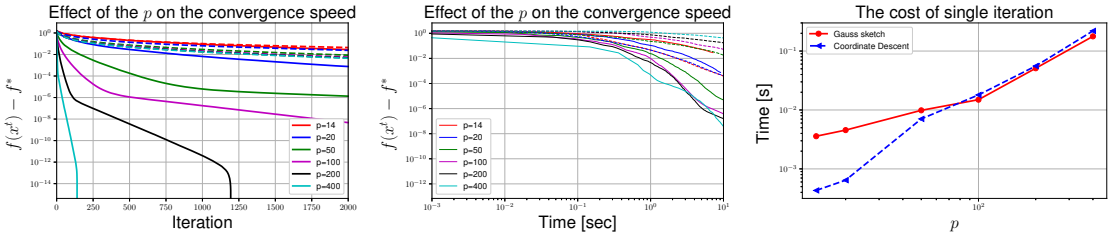

FIG. 3: Algorithm RSD on Markovitz portfolio optimization problem with 13 linear constraints: solid line is for RSD with Gaussian sketch and dashed line is for coordinate descent variant.

Experiment #4: Very large- and small-scale page-rank problem. In this experiment we consider page rank problem (see Section 1.2.1) with dimensions ranging from $n = 10^3$ to $n = 10^6$.

Page rank on random networks. In Figure 4 we compare the speed of Algorithm RSD using a random coordinate sketch for various values of sketching size p. We build random graphs with dimensions ranging from $n = 10^3$ to $n = 10^6$ such that each column has on average 10 or 40 non-zero elements arranged on random places. One can observe that the algorithm has no problem to achieve a very high accuracy in

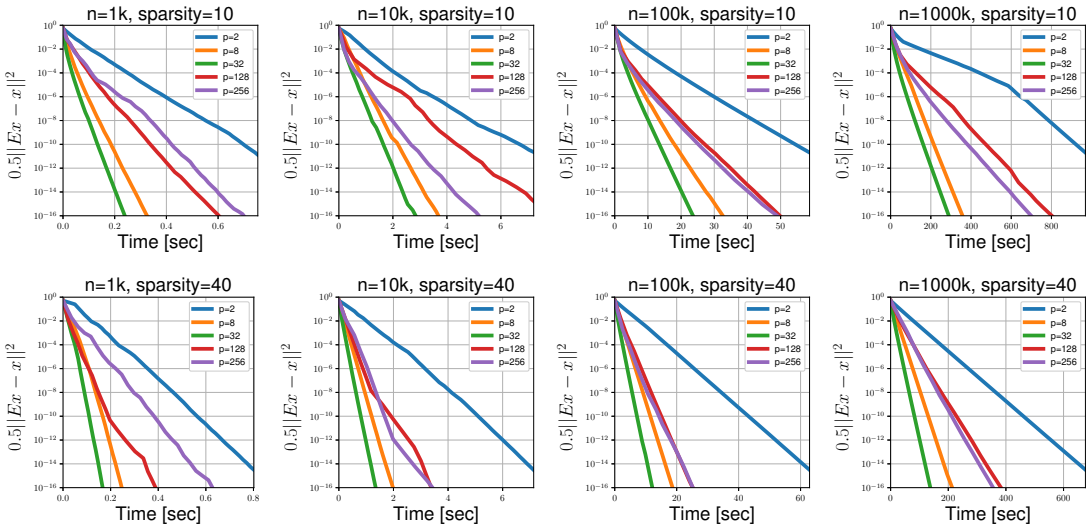

FIG. 4: Performance of Algorithm RSD on randomly generated networks.

relatively short time. Moreover, choosing around $p = 32$ coordinates leads to the fastest convergence (in terms of running time).

Page rank on Wikipedia network. Computation of page-rank on a random network is relatively easy, as the

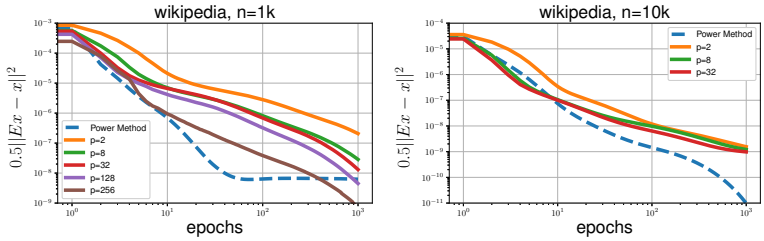

FIG. 5: Performance of Algorithm RSD on a subset of Wikipedia network.

Hessian is well-conditioned. This motivated us to also test RSD on a real network data. In this experiment we consider the Wikipedia dump and extract a link structure between pages. We have restricted ourselves on a subset of pages written in English language. We have chosen the top 10^3 or 10^4 pages (top in a sense of number of outgoing links from given page). In Figure 5 we compare the Algorithm RSD with a random coordinate sketch for different values of sampling size p and the Power method. Observe, that this problem is much harder when compared to random networks. Moreover, for all sketching sizes p chosen, our algorithm RSD is performing better than Power method in terms of number of full iterations (epochs). Also, more coordinates we sample, less epochs are needed to achieve given accuracy to solution.

Conclusions. In this paper we have designed novel sketch descent methods (random sketch descent and accelerated random sketch descent) for solving general smooth linearly constrained problems. From our knowledge, this is the first complete convergence analysis of random sketch descent algorithms for optimization problems with multiple non-separable linear constraints. In special cases, where complexity bounds are known for some particular sketching algorithms, such as coordinate descent methods for optimization problems with a single linear coupled constraint, our convergence rates recover the best known bounds. The numerical examples also illustrate the performances of our new algorithms.

Acknowledgements

The work of Ion Necoara was supported by the Executive Agency for Higher Education, Research and Innovation Funding (UEFISCDI), Romania, under PNIII-P4-PCE-2016-0731, project ScaleFreeNet, no. 39/2017. The work of Martin Takáč was partially supported by the U.S. National Science Foundation, under award numbers NSF:CCF:1618717, NSF:CMMI:1663256 and NSF:CCF:1740796.

REFERENCES

BECK, A. (2014) The 2-coordinate descent method for solving double-sided simplex constrained minimization problems. *Journal of Optimization Theory and Applications*, **162**, 892–919.

BECK, A. & TETRUASHVILI, L. (2013) On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, **23**, 2037–2060.

BERAHAS, A. S., BOLLAPRAGADA, R. & NOCEDAL, J. (2017) An investigation of newton-sketch and subsampled newton methods. *arXiv:1705.06211*.

DENG, W., LAI, M.-J., PENG, Z. & YIN, W. (2017) Parallel multi-block admm with $o(1/k)$ convergence. *Journal of Scientific Computing*, **71**, 712–736.

FERCOQ, O. & RICHTÁRIK, P. (2015) Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, **25**, 1997–2023.

FRONGILLO, R. & REID, M. D. (2015) Convergence analysis of prediction markets via randomized subspace descent. *Advances in Neural Information Processing Systems*, 3034–3042.

GURBUZBALABAN, M., OZDAGLAR, A., PARRILO, P. A. & VANLI, N. (2017) When cyclic coordinate descent outperforms randomized coordinate descent. *Advances in Neural Information Processing Systems*, 6999–7007.

HONG, M. & LUO, Z.-Q. (2017) On the linear convergence of the alternating direction method of multipliers. *Mathematical Programming*, **162**, 165–199.

ISHII, H., TEMPO, R. & BAI, E.-W. (2012) A web aggregation approach for distributed randomized pagerank algorithms. *IEEE Transactions on Automatic Control*, **57**, 2703–2717.

KARIMI, H., NUTINI, J. & SCHMIDT, M. (2016) Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, Springer, pp. 795–811.

LEE, Y. T. & SIDFORD, A. (2013) Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. *IEEE Symposium on Fondations of Computer Science (arXiv:1305.1922)*.

LIN, T., MA, S. & ZHANG, S. (2016) Iteration complexity analysis of multi-block admm for a family of convex minimization without strong convexity. *Journal of Scientific Computing*, **69**, 52–81.

LIU, J. & WRIGHT, S. J. (2015) Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization*, **25**, 351–376.

LU, Z. & XIAO, L. (2015) On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming*, **152**, 615–642.

LUO, Z.-Q. & TSENG, P. (1993) Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, **46**, 157–178.

MARKOWITZ, H. (1952) Portfolio selection. *The Journal of Finance*, **7**, 77–91.

NECOARA, I. (2013) Random coordinate descent algorithms for multi-agent convex optimization over networks. *IEEE Transactions on Automatic Control*, **58**, 2001–2012.

NECOARA, I., NESTEROV, Y. & GLINEUR, F. (2017) Random block coordinate descent methods for linearly constrained optimization over networks. *Journal of Optimization Theory and Applications*, **173**, 227–254.

NECOARA, I. & CLIPICI, D. (2016) Parallel random coordinate descent method for composite minimization: Convergence analysis and error bounds. *SIAM Journal on Optimization*, **26**, 197–226.

NECOARA, I. & PATRASCU, A. (2014) A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints. *Computational Optimization and Applications*, **57**, 307–337.

NEDELCU, V., NECOARA, I. & TRAN DINH, Q. (2014) Computational complexity of inexact gradient augmented lagrangian methods: Application to constrained mpc. *SIAM Journal of Control and Optimization*, **52**, 3109–3134.

NESTEROV, Y. (2012) Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, **22**, 341–362.

NESTEROV, Y. (2013) *Introductory lectures on convex optimization: A basic course*, vol. 87. Springer Science & Business Media.

PILANCI, M. & WAINWRIGHT, M. (2017) Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence. *SIAM Journal on Optimization*, **27**, 205–245.

QU, Z., RICHTÁRIK, P., TAKÁČ, M. & FERCOQ, O. (2016) SDNA: stochastic dual newton ascent for empirical risk minimization. *International Conference on Machine Learning*, 1823–1832.

REDDI, S., HEFNY, A., DOWNEY, C., DUBEY, A. & SRA, S. (2015) Large-scale randomized-coordinate descent methods with non-separable linear constraints. *Conference on Uncertainty in Artificial Intelligence (arXiv:1409.2617)*, 762–771.

RICHTÁRIK, P. & TAKÁČ, M. (2014) Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, **144**, 1–38.

RICHTÁRIK, P. & TAKÁČ, M. (2016) Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, **156**, 433–484.

RICHTÁRIK, P. & TAKÁČ, M. (2020) Stochastic reformulations of linear systems: algorithms and convergence theory. *SIAM Journal on Matrix Analysis and Applications (arXiv:1706.01108)*, **to appear**.

SHALEV-SHWARTZ, S. & ZHANG, T. (2013) Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, **14**, 567–599.

SUN, R. & YE, Y. (2016) Worst-case complexity of cyclic coordinate descent: $o(n^2)$ gap with randomized version. *Mathematical Programming (arXiv:1604.07130)*, **to appear**.

TAKÁČ, M., RICHTÁRIK, P. & SREBRO, N. (2019) Distributed mini-batch SDCA. *Journal of Machine Learning Research (arXiv:1507.08322)*, **to appear**.

TSENG, P. & YUN, S. (2009) Block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization. *Journal of Optimization Theory and Applications*, **140**, 513.

TU, S., VENKATARAMAN, S., WILSON, A. C., GITTENS, A., JORDAN, M. I. & RECHT, B. (2017) Breaking locality accelerates block gauss-seidel. *International Conference on Machine Learning (arXiv:1701.03863)*, 3482–3491.

WANG, J., LEE, J. D., MAHDAVI, M., KOLAR, M. & SREBRO, N. (2017) Sketching meets random projection in the dual: A provable recovery algorithm for big and high-dimensional data. *Electronic Journal of Statistics*, **11**, 4896–4944.

WEI, E., OZDAGLAR, A. & JADBABAIE, A. (2013) A distributed newton method for network utility maximization–i: Algorithm. *IEEE Transactions on Automatic Control*, **58**, 2162–2175.

WRIGHT, S. J. (2012) Accelerated block-coordinate relaxation for regularized optimization. *SIAM Journal on Optimization*, **22**, 159–186.

XIAO, L. & BOYD, S. (2006) Optimal scaling of a gradient method for distributed resource allocation. *Journal of Optimization Theory and Applications*, **129**, 469–488.

Appendix. Efficient implementation of A-RSD

Algorithm 3 Efficient implementation of A-RSD for sparse sketching: strongly convex case

Input: Positive sequences $\{\alpha_k\}_{k=0}^{\infty}, \{\beta_k\}_{k=0}^{\infty}, \{\gamma_k\}_{k=0}^{\infty}$

choose $x^0 \in \mathbb{R}^n$ such that $Ax^0 = b$ and set $u^0 = w^0 = x^0$, $B_0 = I_2$

for $k \geqslant 0$ **do**

sample $S \sim \mathscr{S}$ and compute $g = Z_S \nabla f \left(\underbrace{B_k^{11} u^k + B_k^{12} w^k}_{y^k} \right)$

update $B_{k+1} = A_k B_k$

$$\begin{pmatrix} u^{k+1} \\ w^{k+1} \end{pmatrix} = \begin{pmatrix} u^k \\ w^k \end{pmatrix} - B_{k+1}^{-1} \begin{pmatrix} (1 - \alpha_{k+1}(1-\gamma_k))g \\ \gamma_k g \end{pmatrix}$$

end for

In this appendix we discuss how to implement the A-RSD updates without full-dimensional vector operations. Recall that we assume the following settings: the sketch matrix S is sparse and we can efficiently evaluate $\nabla f(\alpha v+\beta u)$. First we derive an efficient implementation of A-RSD iterations for strongly convex objective functions and then a simplified implementation for the convex case. Following a similar approach as in the coordinate descent work proposed in Lee & Sidford (2013) for solving linear systems and further extended in Fercoq & Richtárik (2015) for accelerated coordinate descent method with separable composite problems we note that:

$$\begin{aligned} y^{k+1} &= \alpha_{k+1}v^{k+1} + (1-\alpha_{k+1})x^{k+1} \\ &= (1-\alpha_{k+1}\beta_k)y^k + \alpha_{k+1}\beta_k v^k - (1-\alpha_{k+1}(1-\gamma_k))Z_S\nabla f(y^k). \end{aligned}$$

Hence, we obtain the following recursion:

$$\begin{pmatrix} y^{k+1} \\ v^{k+1} \end{pmatrix} = A_k \begin{pmatrix} y^k \\ v^k \end{pmatrix} - s_k, \tag{A.1}$$

with

$$A_k = \begin{pmatrix} 1-\alpha_{k+1}\beta_k & \alpha_{k+1}\beta_k \\ 1-\beta_k & \beta_k \end{pmatrix}, \qquad s_k = \begin{pmatrix} (1-\alpha_{k+1}(1-\gamma_k))Z_S\nabla f(y^k) \\ \gamma_k Z_S\nabla f(y^k) \end{pmatrix}.$$

Now, our goal is to maintain two sequences $\{u^k\}_k, \{w^k\}_k$ such that: $\begin{pmatrix} y^k \\ v^k \end{pmatrix} = B_k \begin{pmatrix} u^k \\ w^k \end{pmatrix}$. Therefore, it has to hold that

$$B_{k+1}\begin{pmatrix} u^{k+1} \\ w^{k+1} \end{pmatrix} = \begin{pmatrix} y^{k+1} \\ v^{k+1} \end{pmatrix} \overset{(A.1)}{=} A_k B_k \begin{pmatrix} u^k \\ w^k \end{pmatrix} - s_k,$$

and therefore we require

$$\begin{pmatrix} u^{k+1} \\ w^{k+1} \end{pmatrix} = B_{k+1}^{-1}A_k B_k \begin{pmatrix} u^k \\ w^k \end{pmatrix} - B_{k+1}^{-1}s_k.$$

In order to make this computationally efficient, it is sufficient to define B_k recursively as: $B_0 = I_2$, $B_{k+1} = A_k B_k$, $u^0 = y^0$ and $w^0 = v^0$, to obtain the following update rule

$$\begin{pmatrix} u^{k+1} \\ w^{k+1} \end{pmatrix} = \begin{pmatrix} u^k \\ w^k \end{pmatrix} - B_{k+1}^{-1}s_k,$$

which is a sparse update provided that s_k is a sparse vector. However, when the sketch matrix S is sparse the vector $Z_S\nabla f(y^k)$ is sparse as well and consequently s_k is also a sparse vector (see Example 2.3 where for $S_{(i,j)} = [e_i\ e_j]$ the corresponding vector $Z_{(i,j)}\nabla f(y)$ has only two non-zero entries). The final algorithm is depicted in Algorithm 3.

Simplified Convex Case.
In the case of non-strongly convex objective function, the implementation can be significantly simplified using the fact that $\beta_k = 1$ for all k. Then, we have:

$$v^{k+1} = v^k - \gamma_k Z_S\nabla f(y^k) \tag{A.2}$$

and

$$\begin{aligned} y^{k+1} - v^{k+1} &= \alpha_{k+1}v^{k+1} + (1-\alpha_{k+1})x^{k+1} - v^{k+1} \\ &= (1-\alpha_{k+1})(y^k - Z_S\nabla f(y^k) - v^k + \gamma_k Z_S\nabla f(y^k)) \\ &= (1-\alpha_{k+1})(y^k - v^k) - (1-\alpha_{k+1})(1-\gamma_k)Z_S\nabla f(y^k). \end{aligned}$$

Algorithm 4 Efficient implementation of A-RSD for sparse sketching: convex case

Input: Positive sequences $\{\alpha_k\}_{k=0}^{\infty}, \{\gamma_k\}_{k=0}^{\infty}$
choose $x^0 \in \mathbb{R}^n$ such that $Ax^0 = b$
set $v^0 = x^0$, $u^0 = \mathbf{0}$ and $b_0 = 1$
for $k \geqslant 0$ **do**
  sample $S \sim \mathscr{S}$ and compute $g = Z_S \nabla f \left(\underbrace{v^k + b_k u^k}_{y^k} \right)$
  update $v^{k+1} = v^k - \gamma_k g$
  $u^{k+1} = u^k - \frac{1-\gamma_k}{b_k} g$
  $b_{k+1} = (1-\alpha_{k+1}) b_k$
end for

Therefore, we obtain the following recursion:

$$\begin{pmatrix} y^{k+1} - v^{k+1} \\ v^{k+1} \end{pmatrix} = \tilde{A}_k \begin{pmatrix} y^k - v^k \\ v^k \end{pmatrix} - \tilde{s}_k, \tag{A.3}$$

with

$$\tilde{A}_k = \begin{pmatrix} 1-\alpha_{k+1} & 0 \\ 0 & 1 \end{pmatrix}, \qquad \tilde{s}_k = \begin{pmatrix} (1-\alpha_{k+1})(1-\gamma_k) Z_S \nabla f(y^k) \\ \gamma_k Z_S \nabla f(y^k) \end{pmatrix}.$$

Now, we see that the update of v^k given by (A.2) is sparse if $Z_S \nabla f(y^k)$ is sparse. Further, we want to express $y^{k+1} - v^{k+1} = b_{k+1} u^{k+1}$. Then, from (A.3) we have:

$$\begin{aligned} b_{k+1} u^{k+1} = y^{k+1} - v^{k+1} &= (1-\alpha_{k+1})(y^k - v^k) - (1-\alpha_{k+1})(1-\gamma_k) Z_S \nabla f(y^k) \\ &= (1-\alpha_{k+1}) b_k u^k - (1-\alpha_{k+1})(1-\gamma_k) Z_S \nabla f(y^k). \end{aligned}$$

Therefore, if we define $b_{k+1} = (1-\alpha_{k+1}) b_k$, this will simplify to:

$$u^{k+1} = u^k - \frac{(1-\alpha_{k+1})(1-\gamma_k)}{b_{k+1}} Z_S \nabla f(y^k) = u^k - \frac{1-\gamma_k}{b_k} Z_S \nabla f(y^k).$$

It follows that the update of u^k is also sparse if $Z_S \nabla f(y^k)$ is sparse. Next, we can easily compute $y^k = v^k + b_k u^k$ (however, this shouldn't be formed during the run of the algorithm). Finally, it is sufficient to note that $v^0 = x^0$, $u^0 = 0$ and we can choose $b_0 = 1$. The final algorithm is given in Algorithm 4.