SIAM J. SCI. COMPUT.
Vol. 42, No. 6, pp. A3979–A4016

CONVERGENCE ANALYSIS OF INEXACT RANDOMIZED ITERATIVE METHODS*

NICOLAS LOIZOU† AND PETER RICHTÁRIK‡

Abstract. In this paper we present a convergence rate analysis of inexact variants of several randomized iterative methods for solving three closely related problems: a convex stochastic quadratic optimization problem, a best approximation problem, and its dual, a concave quadratic maximization problem. Among the methods studied are stochastic gradient descent, stochastic Newton, stochastic proximal point, and stochastic subspace ascent. A common feature of these methods is that in their update rule a certain subproblem needs to be solved exactly. We relax this requirement by allowing for the subproblem to be solved inexactly. We provide iteration complexity results under several assumptions on the inexactness error. Inexact variants of many popular and some more exotic methods, including randomized block Kaczmarz, Gaussian block Kaczmarz, and randomized block coordinate descent, can be cast as special cases. Numerical experiments demonstrate the benefits of allowing inexactness.

Key words. inexact methods, iteration complexity, linear systems, randomized block coordinate descent, randomized block Kaczmarz, stochastic gradient descent, stochastic Newton method, quadratic optimization, convex optimization

AMS subject classifications. 68Q25, 68W20, 68W40, 65Y20, 90C15, 90C20, 90C25, 15A06, 15B52, 65F10

DOI. 10.1137/19M125248X

1. Introduction.

In the era of big data where datasets become continuously larger, randomized iterative methods have become very popular, and they are now playing a major role in areas like numerical linear algebra, scientific computing, and optimization. They are preferred mainly because of their cheap per iteration cost which leads to the improvement in terms of complexity upon classical results by orders of magnitude and to the fact that they can easily scale to extreme dimensions. However, a common feature of these methods is that in their update rule a particular subproblem needs to be solved exactly. In the case that the size of this problem is large, this step can be computationally very expensive. The purpose of this work is to reduce the cost of this step by incorporating inexact updates into the stochastic methods under study.

1.1. The setting.

In this paper we are interested in solving three closely related problems:

(i) Stochastic quadratic optimization problem,
(ii) Best approximation problem,
(iii) Concave quadratic maximization problem.

We start by presenting the main connections and key relationships between these problems as well as popular randomized iterative methods (with exact updates) for

*Submitted to the journal's Methods and Algorithms for Scientific Computing section March 25, 2019; accepted for publication (in revised form) August 19, 2020; published electronically December 15, 2020. Most of the work was done when both authors were associated with School of Mathematics, University of Edinburgh, UK.

https://doi.org/10.1137/19M125248X

†Mila and DIRO, Université de Montréal, Montreal, Quebec, H2S 3H1, Canada (nicolasloizou1@gmail.com).

‡Computer Science, King Abdullah University of Science and Technology (KAUST), Thuwal, 23955, Kingdom of Saudi Arabia (peter.richtarik@kaust.edu.sa).

solving each one of them.

Stochastic quadratic optimization problem. We study the stochastic quadratic optimization problem

$$\min_{x\in\mathbb{R}^n} f(x) := \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}[f_{\mathbf{S}}(x)], \tag{1.1}$$

first proposed in [54], for reformulating *consistent* linear systems

$$\mathbf{A}x = b. \tag{1.2}$$

In particular, problem (1.1) is defined by setting

$$f_{\mathbf{S}}(x) := \tfrac{1}{2}\|\mathbf{A}x - b\|^2_{\mathbf{H}} = \tfrac{1}{2}(\mathbf{A}x-b)^\top \mathbf{H}(\mathbf{A}x-b), \tag{1.3}$$

where $\mathbf{H} := \mathbf{S}(\mathbf{S}^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S})^\dagger\mathbf{S}^\top$ is a random symmetric positive semidefinite matrix that depends on three different matrices: the data matrix $\mathbf{A}\in\mathbb{R}^{m\times n}$ of the linear system (1.2), a random matrix $\mathbf{S}\in\mathbb{R}^{m\times q}\sim\mathcal{D}$, and an $n\times n$ positive definite matrix $\mathbf{B}$ which defines the geometry of the space.[1] Throughout the paper, $\mathbf{B}$ is used to define a $\mathbf{B}$-inner product in $\mathbb{R}^n$ via $\langle x,z\rangle_{\mathbf{B}} := \langle \mathbf{B}x, z\rangle$ and an induced $\mathbf{B}$-norm, $\|x\|_{\mathbf{B}} := (x^\top\mathbf{B}x)^{1/2}$. By $\dagger$ we denote the Moore–Penrose pseudoinverse.

The expectation in (1.1) is over random matrices $\mathbf{S}$ with m rows (and an arbitrary number of columns q, e.g., $q=1$) drawn from an arbitrary (user-defined) distribution $\mathcal{D}$. The authors of [54] give necessary and sufficient conditions that distribution $\mathcal{D}$ needs to satisfy for the set of solutions of (1.1) to be equal to the set of solutions of the linear system (1.2)—a property for which the term exactness was coined (see section 3 for more details on exactness).

In [54], problem (1.1) was solved via stochastic gradient descent (SGD)[2]:

$$x_{k+1} = x_k - \omega\nabla f_{\mathbf{S}_k}(x_k), \tag{1.4}$$

and a linear rate of convergence was proved despite the facts that f is not necessarily strongly convex, (1.1) is not a finite-sum problem, and a fixed stepsize $\omega>0$ is used.

The stochastic optimization problem (1.1) has many unique characteristics. For example, it holds that $f_{\mathbf{S}}(x) = \frac{1}{2}\|\nabla f_{\mathbf{S}}(x)\|^2_{\mathbf{B}}$, and it can be proved that all eigenvalues of the Hessian matrix $\nabla^2 f(x)$ are upper bounded by 1. Due to these specific characteristics, the update rules of seemingly different randomized iterative methods are identical. In particular the following methods for solving (1.1) have exactly the same behavior with SGD [54]:

- *Stochastic Newton method (SNM)*[3]:

$$x_{k+1} = x_k - \omega(\nabla^2 f_{\mathbf{S}_k}(x_k))^{\dagger_{\mathbf{B}}}\nabla f_{\mathbf{S}_k}(x_k). \tag{1.5}$$

[1] During the first reading of the paper, the reader might find it convenient to assume that $\mathbf{B}=\mathbf{I}$ throughout the paper. Very little will be lost this way, and all results of our paper can be appreciated even with this special choice of $\mathbf{B}$. Still, we prefer to cast all problems for general positive definite matrix $\mathbf{B}$. From a practical perspective, $\mathbf{B}$ should be seen as a *parameter* which we are free to choose in any way in order to achieve certain goals. First and foremost, the choice of $\mathbf{B}$ will *not* affect the solution set of the stochastic optimization problem. However, it will affect the solution set of the best approximation problem and its dual, which we shall discuss next. Likewise, the choice of $\mathbf{B}$ acts as a *selector* of the particular solution of the stochastic optimization problem our methods will converge to.

[2] The gradient is computed with respect to the inner product $\langle \mathbf{B}x, y\rangle$. This is obtained by premultiplying the standard gradient by $\mathbf{B}^{-1}$. Indeed, if $g(x)$ is the "standard" gradient, then $\langle g(x), y\rangle = \langle \mathbf{B}^{-1}g(x), y\rangle_{\mathbf{B}}$ for all x, y.

[3] In this method we take the $\mathbf{B}$-pseudoinverse of the Hessian of $f_{\mathbf{S}_k}$ instead of the classical inverse, as the inverse does not exist. The $\mathbf{B}$ pseudoinverse of matrix $\mathbf{U}$ is defined to be $\mathbf{U}^{\dagger_{\mathbf{B}}} = \mathbf{B}^{-1}\mathbf{U}^\top(\mathbf{U}\mathbf{B}^{-1}\mathbf{U}^\top)^\dagger$. When $\mathbf{B}=\mathbf{I}$, the $\mathbf{B}$-pseudoinverse specializes to the standard Moore–Penrose pseudoinverse. See also Table 4, and for properties see [54].

- *Stochastic proximal point method (SPPM)*[4]:

$$x_{k+1} = \arg\min_{x\in\mathbb{R}^n} \left\{ f_{\mathbf{S}_k}(x) + \tfrac{1-\omega}{2\omega}\|x - x_k\|^2_{\mathbf{B}} \right\}. \tag{1.6}$$

In all methods $\omega > 0$ is a fixed stepsize and $\mathbf{S}_k$ is sampled afresh in each iteration from distribution $\mathcal{D}$. See [54] for more insights into the reformulation (1.1), its properties, and other equivalent reformulation (e.g., stochastic fixed point problem, probabilistic intersection problem, and stochastic linear system).

Best approximation problem and sketch and project method. In [54, 35], it has been shown that for the case of consistent linear systems with multiple solutions, SGD (and as a result SNM (1.5) and SPPM (1.6)) converges linearly to one particular minimizer of function f, the projection of the initial iterate x_0 onto the solution set of the linear system (1.2). This naturally leads to the *best approximation problem*

$$\min_{x\in\mathbb{R}^n} P(x) := \tfrac{1}{2}\|x - x_0\|^2_{\mathbf{B}} \quad \text{subject to} \quad \mathbf{A}x = b. \tag{1.7}$$

Unlike the linear system (1.2), which is allowed to have multiple solutions, the best approximation problem always has (from its construction) a unique solution. For solving problem (1.7), the *sketch and project method* (SPM),

$$x_{k+1} = \omega\Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k) + (1-\omega)x_k, \tag{1.8}$$

was analyzed in [23, 54]. Here, $\Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k)$ denotes the projection of point x_k onto $\mathcal{L}_{\mathbf{S}_k} = \{x \in \mathbb{R}^n \ : \ \mathbf{S}_k^\top \mathbf{A}x = \mathbf{S}_k^\top b\}$ in the $\mathbf{B}$-norm. In the special case of unit stepsize ($\omega = 1$) algorithm (1.8) simplifies to

$$x_{k+1} = \Pi_{\mathcal{L}_{\mathbf{S}},\mathbf{B}}(x_k), \tag{1.9}$$

which was first proposed in [23]. The name *sketch and project method* is justified by the iteration structure which follows two steps: (i) Choose the *sketched* system $\mathcal{L}_{\mathbf{S}_k} := \{x \ : \ \mathbf{S}_k^\top \mathbf{A}x = \mathbf{S}_k^\top b\}$; (ii) *project* the last iterate x_k onto $\mathcal{L}_{\mathbf{S}_k}$. The sketch and project viewpoint will be useful later in explaining the natural interpretation of the proposed inexact update rules (see section 4.2).

Dual problem and stochastic dual subspace ascent. The Fenchel dual of (1.7) is the (bounded) unconstrained concave quadratic maximization problem

$$\max_{y\in\mathbb{R}^m} D(y) := (b - \mathbf{A}x_0)^\top y - \tfrac{1}{2}\|\mathbf{A}^\top y\|^2_{\mathbf{B}^{-1}}. \tag{1.10}$$

Boundedness follows from consistency. It turns out that by varying $\mathbf{A}$, $\mathbf{B}$, and b (but keeping consistency of the linear system), the dual problem in fact captures *all* bounded unconstrained concave quadratic maximization problems [35].

A direct dual method for solving problem (1.10) was first proposed in [24]. The dual method, *stochastic dual subspace ascent* (SDSA), updates the dual vectors y_k as follows:

$$y_{k+1} = y_k + \omega\mathbf{S}_k\lambda_k, \tag{1.11}$$

where the random matrix $\mathbf{S}_k$ is sampled afresh in each iteration from distribution $\mathcal{D}$, and λ_k is chosen in such a way to maximize the dual objective D. That is,

[4] In this case, the equivalence only works for $0 < \omega \leq 1$.

$\lambda_k \in \arg\max_\lambda D(y_k + \mathbf{S}_k\lambda)$. More specifically, SDSA is defined by picking the λ_k with the smallest (standard Euclidean) norm. This leads to the formula

$$\lambda_k = \left(\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top \left(b - \mathbf{A}(x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k)\right). \tag{1.12}$$

It can be proved [24, 35] that the iterates $\{x_k\}_{k\geq 0}$ of the SPM (1.8) arise as affine images of the iterates $\{y_k\}_{k\geq 0}$ of the dual method (1.11) as follows:

$$x_k = x(y_k) = x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k. \tag{1.13}$$

In [24] the dual method was analyzed for the case of unit stepsize ($\omega = 1$). Later in [35] the analysis extended to capturing the cases of $\omega \in (0, 2)$. Momentum variants of the dual method that provide further speed-up have been also studied in [35].

An interesting property that holds between the suboptimalities of the SPM and SDSA is that the dual suboptimality of y in terms of the dual function values is equal to the primal suboptimality of $x(y)$ in terms of distance [24, 35]. That is,

$$D(y_*) - D(y) = \tfrac{1}{2}\|x(y_*) - x(y)\|_{\mathbf{B}}^2. \tag{1.14}$$

This simple-to-derive result (by combining the expression of the dual function $D(y)$ (1.10) and (1.13)) gives for free the convergence analysis of SDSA, in terms of dual function suboptimality, once the analysis of SPM is available (see section 5).

1.2. Contributions. In this work we propose and analyze *inexact* variants of all previously mentioned randomized iterative algorithms for solving the stochastic optimization problem, the best approximation problem, and the dual problem. In all of these methods, a certain potentially expensive calculation/operation needs to be performed in each step; it is this operation that we propose be performed inexactly. For instance, in the case of SGD, it is the computation of the stochastic gradient $\nabla f_{\mathbf{S}_k}(x_k)$, in the case of SPM it is the computation of the projection $\Pi_{\mathcal{L}_{\mathbf{S}},\mathbf{B}}(x_k)$, and in the case of SDSA it is the computation of the dual update $\mathbf{S}_k\lambda_k$.

We perform an iteration complexity analysis under an abstract notion of inexactness and also under a more structured form of inexactness appearing in practical scenarios. An inexact solution of these subproblems can be obtained much more quickly than the exact solution. Since in practical applications the savings thus obtained are larger than the increase in the number of iterations needed for convergence, our inexact methods can be dramatically faster.

Let us now briefly outline the rest of the paper.

In section 2 we describe the subproblems and introduce two notions of inexactness (abstract and structured) that will be used in the rest of the paper. The inexact basic method (iBasic) is also presented. iBasic is a method that simultaneously captures inexact variants of the algorithms (1.4), (1.5), (1.6) for solving the stochastic optimization problem (1.1) and algorithm (1.8) for solving the best approximation problem (1.7). It is an inexact variant of the *basic method*, first presented in [54], where the inexactness is introduced by the addition of an inexactness error ϵ_k in the original update rule. We illustrate the generality of iBasic by presenting popular algorithms that can be cast as special cases.

In section 3 we establish convergence results of iBasic under general assumptions on the inexactness error ϵ_k of its update rule (see Algorithm 2.1). In this part we do not focus on any specific mechanisms which lead to inexactness; we treat the problem abstractly. However, such errors appear often in practical scenarios and

TABLE 1

Summary of the iteration complexity results obtained in this paper. ω denotes the stepsize (relaxation parameter) of the method. In all cases, $x_* = \Pi_{\mathcal{L},\mathbf{B}}(x_0)$ and $\rho = 1-\omega(2-\omega)\lambda_{\min}^{+} \in (0,1)$ are the quantities appearing in the convergence results (here $\lambda_{\min}^{+}$ denotes the minimum nonzero eigenvalue of matrix $\mathbf{W}$; see (1.19)). Inexactness parameter q is always chosen in such a way to obtain linear convergence and it can be seen as the quantity that controls the inexactness. In all theorems the quantity of convergence is $\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}^2]$ (except in Theorem 3.6, where we analyze $\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}]$). As we show in section 5, under similar assumptions, iSDSA has exactly the same convergence with iBasic, but the upper bounds of the third column are related to the dual function values $\mathbb{E}\left[D(y_*) - D(y_0)\right]$.

Assumption on the inexactness error ϵ_k	$\omega \in$	Upper bounds	Theorem
Assumption 3.2	$(0,2)$	$\rho^{k/2}\|x_0 - x_*\|_{\mathbf{B}} + \sum_{i=0}^{k-1} \rho^{\frac{k-1-i}{2}} \sigma_i$	3.6
Assumption 3.3	$(0,2)$	$\left(\sqrt{\rho} + q\right)^{2k} \|x_0 - x_*\|_{\mathbf{B}}^2$	3.8
Assumptions 3.1 + 3.5	$(0,2)$	$\rho^k \|x_0 - x_*\|_{\mathbf{B}}^2 + \sum_{i=0}^{k-1} \rho^{k-1-i} \bar{\sigma}_i^2$	3.9(i)
Assumptions 3.3 + 3.5	$(0,2)$	$\left(\rho + q^2\right)^k \|x_0 - x_*\|_{\mathbf{B}}^2$	3.9(ii)
Assumptions 3.4 + 3.5	$(0,2)$	$\left(\rho + q^2\lambda_{\min}^{+}\right)^k \|x_0 - x_*\|_{\mathbf{B}}^2$	3.9(iii)

can be associated with inaccurate numerical solvers, quantization, sparsification, and compression mechanisms. In particular, we introduce several abstract assumptions on the inexactness level and describe our generic convergence results. For all assumptions we establish a linear rate of decay of the quantity $\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}^2]$ (i.e., L2 convergence).[5]

Subsequently, in section 4 we apply our general convergence results to a more structured notion of inexactness error and propose a concrete mechanism leading to such errors. We provide theoretical guarantees for this method in situations when a linearly convergent iterative method (e.g., conjugate gradient (CG)) is used to solve the subproblem inexactly. We also highlight the importance of the dual viewpoint through a sketch and project interpretation.

In section 5 we study an inexact variant of SDSA, which we called iSDSA, for directly solving the dual problem (1.10). We provide a correspondence between iBasic and iSDSA, and we show that the random iterates of iBasic arise as affine images of iSDSA. We consider both abstract and structured inexactness errors and provide linearly convergent rates in terms of the dual function suboptimality $\mathbb{E}\left[D(y_*) - D(y_0)\right]$.

Finally, in section 6 we evaluate the performance of the proposed inexact methods through numerical experiments and show the benefits of our approach on both synthetic and real datasets. Concluding remarks are given in section 7.

A summary of the convergence results of iBasic under several assumptions on the inexactness error with pointers to the relevant theorems is available in Table 1. We highlight that similar convergence results can be also obtained for iSDSA in terms of the dual function suboptimality $\mathbb{E}\left[D(y_*) - D(y_0)\right]$ (check section 5 for more details on iSDSA).

1.3. Notation. For convenience, a table of the most frequently used notation is included in Appendix C. In particular, with boldface uppercase letters we denote matrices, and $\mathbf{I}$ is the identity matrix. By $\mathcal{L}$ we denote the solution set of the linear system $\mathbf{A}x = b$. By $\mathcal{L}_{\mathbf{S}}$, where $\mathbf{S}$ is a random matrix, we denote the solution set of the *sketched* linear system $\mathbf{S}^\top\mathbf{A}x = \mathbf{S}^\top b$. In general, we use $\cdot^*$ to express the exact solution of a subproblem and $\cdot^{\approx}$ to indicate its inexact variant. Unless stated

[5] As we explain later, a convergence of the expected function values of problem (1.1) can be easily obtained as a corollary of L2 convergence.

otherwise, throughout the paper, x_* is the projection of x_0 onto $\mathcal{L}$ in the $\mathbf{B}$-norm: $x_* = \Pi_{\mathcal{L},\mathbf{B}}(x_0)$. An explicit formula for the projection of point x onto set $\mathcal{L}$ is given by

$$\Pi_{\mathcal{L},\mathbf{B}}(x) := \arg\min_{x' \in \mathcal{L}} \|x' - x\|_{\mathbf{B}} = x - \mathbf{B}^{-1}\mathbf{A}^\top(\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top)^\dagger(\mathbf{A}x - b). \tag{1.15}$$

A formula for the projection onto $\mathcal{L}_{\mathbf{S}} = \{x \in \mathbb{R}^n \ : \ \mathbf{S}^\top\mathbf{A}x = \mathbf{S}^\top b\}$ is obtained by replacing $\mathbf{A}$ and b with $\mathbf{S}^\top\mathbf{A}$ and $\mathbf{S}^\top b$, respectively, in the above equation. We denote this projection by $\Pi_{\mathcal{L}_{\mathbf{S}},\mathbf{B}}(x)$. We also write $[n] := \{1, 2, \ldots, n\}$.

In order to keep the expression brief throughout the paper we define[6]

$$\mathbf{Z} := \mathbf{A}^\top\mathbf{H}\mathbf{A} = \mathbf{A}^\top\mathbf{S}(\mathbf{S}^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S})^\dagger\mathbf{S}^\top\mathbf{A}. \tag{1.16}$$

Using this matrix, we can easily express important quantities related to the problems under study. For example, the stochastic functions $f_{\mathbf{S}}$ of problem (1.1) can be expressed as

$$f_{\mathbf{S}}(x) = \tfrac{1}{2}(\mathbf{A}x - b)^\top\mathbf{H}(\mathbf{A}x - b) = \tfrac{1}{2}(x - x_*)^\top\mathbf{Z}(x - x_*). \tag{1.17}$$

In addition, the gradient and the Hessian of $f_{\mathbf{S}}$ with respect to the $\mathbf{B}$-inner product are equal to

$$\nabla f_{\mathbf{S}}(x) \overset{(1.3)}{=} \mathbf{B}^{-1}\mathbf{A}^\top\mathbf{H}(\mathbf{A}x - b) = \mathbf{B}^{-1}\mathbf{A}^\top\mathbf{H}\mathbf{A}(x - x_*) = \mathbf{B}^{-1}\mathbf{Z}(x - x_*) \tag{1.18}$$

and $\nabla^2 f_{\mathbf{S}}(x) = \mathbf{B}^{-1}\mathbf{Z}$ [54]. Similarly, the gradient and Hessian of the objective function f of problem (1.1) are $\nabla f(x) = \mathbf{B}^{-1}\mathbb{E}[\mathbf{Z}](x - x_*)$ and $\nabla^2 f(x) = \mathbf{B}^{-1}\mathbb{E}[\mathbf{Z}]$, respectively.

A key matrix in our analysis is

$$\mathbf{W} := \mathbf{B}^{-\frac{1}{2}}\mathbb{E}[\mathbf{Z}]\mathbf{B}^{-\frac{1}{2}}, \tag{1.19}$$

which has the same spectrum as the matrix $\nabla^2 f(x)$ but at the same time is symmetric and positive semidefinite.[7] We denote by $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ the n eigenvalues of $\mathbf{W}$. By $\lambda_{\min}^+$ we indicate the smallest nonzero eigenvalue and by $\lambda_{\max} = \lambda_n$ the largest eigenvalue. It was shown in [54] that $0 \leq \lambda_i \leq 1$ for all $i \in [n]$.

2. Inexact update rules. In this section we start by explaining the key subproblems that need to be solved exactly in the update rules of the previously described methods. We present iBasic, a method that solves problems (1.1) and (1.7), and we show how by varying the main parameters of the method we recover inexact variants of popular algorithms as special cases. Finally, closely related work on inexact algorithms for solving different problems is also presented.

2.1. Expensive subproblems in update rules. Let us devote this subsection to explaining how the inexactness can be introduced in the current exact update rules of SGD[8] (1.4), SPM (1.8), and SDSA (1.11) for solving the stochastic optimization,

[6] In the k^{th} iteration the expression becomes $\mathbf{Z}_k := \mathbf{A}^\top\mathbf{S}_k(\mathbf{S}_k^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k)^\dagger\mathbf{S}_k^\top\mathbf{A}$.

[7] Note that matrix $\nabla^2 f(x)$ is not symmetric, but it is self-adjoint with respect to the $\mathbf{B}$-inner product. That is, $\langle\nabla^2 f(x)u, v\rangle_{\mathbf{B}} = \langle u, \nabla^2 f(x)v\rangle_{\mathbf{B}}$ for all u, v. In the special case when $\mathbf{B} = \mathbf{I}$, self-adjointness reduces to symmetry.

[8] Note that SGD has identical updates to the SNM and the SPPM. Thus the inexactness can be added to these updates in a similar way.

best approximation, and the dual problem, respectively. As we have shown, these methods solve closely related problems, and the key subproblems in their update rule are similar. However, the introduction of inexactness into the update rule of each one of them can have different interpretation.

For example, in the case of SGD for solving the stochastic optimization problem (1.1), if we define $\lambda_k^* = (\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top (b - \mathbf{A}x_k)$, then the stochastic gradient of function f becomes $\nabla f_{\mathbf{S}_k}(x_k) \overset{(1.18)}{=} -\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_k^*$, and the update rule of SGD takes the form $x_{k+1} = x_k + \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_k^*$. Clearly, in this update the expensive part is the computation of the quantity λ_k^* that can be equivalently computed to be the least norm solution of the smaller (in comparison to $\mathbf{A}x = b$) linear system $\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda = \mathbf{S}_k^\top (b - \mathbf{A}x_k)$. In our work we are suggesting using an approximation $\lambda_k^{\approx}$ of the exact solution and this way avoid executing the possibly expensive step of the update rule. Thus the inexact update takes the following form:

$$x_{k+1} = x_k + \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_k^{\approx} = x_k - \omega \nabla f_{\mathbf{S}_k}(x_k) + \underbrace{\omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k (\lambda_k^{\approx} - \lambda_k^*)}_{\epsilon_k}.$$

Here ϵ_k denotes a more abstract notion of inexactness, and it is not necessary for it to always be equivalent to the quantity $\omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k (\lambda_k^{\approx} - \lambda_k^*)$. It can be interpreted as an expression that acts as a perturbation of the exact update. In the case that ϵ_k has the above form, we say that the notion of inexactness is structured. In our work we are interested in both the *abstract* and more *structured* notions of inexactness. We first present general convergence results where we require the error ϵ_k to satisfy general assumptions (without caring how this error is generated), and later we analyze the concept of structured inexactness by presenting algorithms where $\epsilon_k = \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k (\lambda_k^{\approx} - \lambda_k^*)$.

In a similar way, the expensive operation of SPM (1.8) is the exact computation of the projection $\Pi^*_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k)$. Thus we are suggesting replacing this step with an inexact variant and computing an approximation of this projection. The inexactness here can be also interpreted using both the abstract ϵ_k error and its more structured version $\epsilon_k = \omega(\Pi^{\approx}_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k) - \Pi^*_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k))$. At this point, observe that, by using the expression (1.15), the structure of the ϵ_k in SPM and SGD has the same form.

In SDSA the expensive subproblem in the update rule is the computation of the λ_k^*'s that satisfy $\lambda_k^* \in \arg\max_\lambda D(y_k + \mathbf{S}_k \lambda)$. Using the definition of the dual function (1.10) this value can be also computed by evaluating the least norm solution of the linear system $\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda = \mathbf{S}_k^\top \left(b - \mathbf{A}(x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k)\right)$. Later in section 5 we analyze both notions of inexactness (abstract and more structured) for inexact variants of SDSA.

Table 2 presents the key subproblem that needs to be solved in each algorithm as well as the part where the inexact error appears in the update rule.

2.2. The inexact basic method. In each iteration of the aforementioned exact methods, a sketch matrix $\mathbf{S} \sim \mathcal{D}$ is drawn from a given distribution, and then a certain subproblem is solved exactly to obtain the next iterate. The sketch matrix $\mathbf{S} \in \mathbb{R}^{m \times q}$ is required to have m rows, but no assumption on the number of columns is made, which means that the number of columns q is allowed to vary throughout the iterations, and it can be very large. The setting that we are interested in is precisely that of having such large random matrices $\mathbf{S}$. In these cases we expect that having approximate solutions of the subproblems will be beneficial.

Recently, randomized iterative algorithms that require solving large subproblems

TABLE 2

The exact algorithms under study with the potentially expensive-to-compute key subproblems of their update rule. The inexact update rules are presented in the last column for both notions of inexactness (abstract and structured). We use $\cdot^*$ to define the important quantity that needs to be computed exactly in the update rule of each method and $\cdot^{\approx}$ to indicate the proposed inexact variant. For more details on the notation used in the table, see section 4 and Table 4.

Exact algorithm	Key subproblem (problem that we solve inexactly)	Inexact update rule (abstract & structured inexactness error)
SGD (1.4)	Exact computation of $\lambda_k^* = \arg\min_{\lambda:\mathbf{M}_k\lambda=d_k} \|\lambda\|$ (Appears in the computation of $\nabla f_{\mathbf{S}_k}(x_k) = -\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k\lambda_k^*$)	$x_{k+1} = x_k + \omega\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k\lambda_k^{\approx}$ $= x_k - \omega\nabla f_{\mathbf{S}_k}(x_k) + \underbrace{\omega\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k(\lambda_k^{\approx} - \lambda_k^*)}_{\epsilon_k}$
SPM (1.8)	Exact computation of the projection $\Pi^*_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k) = \arg\min_{x'\in\mathcal{L}_{\mathbf{S}_k}} \|x' - x_k\|_{\mathbf{B}}$	$x_{k+1} = \omega\Pi^{\approx}_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k) + (1-\omega)x_k$ $= \omega\Pi^{\mathbf{B}}_{\mathcal{L}_{\mathbf{S}_k}}(x_k) + (1-\omega)x_k + \underbrace{\omega\left(\Pi^{\approx}_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k) - \Pi^*_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k)\right)}_{\epsilon_k}$
SDSA (1.11)	Exact computation of $\lambda_k^* \in \arg\max_\lambda D(y_k + \mathbf{S}_k\lambda)$	$y_{k+1} = y_k + \omega\mathbf{S}_k\lambda_k^{\approx}$ $= y_k + \omega\mathbf{S}_k\lambda_k^* + \underbrace{\omega\mathbf{S}_k(\lambda_k^{\approx} - \lambda_k^*)}_{\epsilon_k^d}$

in each iteration have been extensively studied and have been shown to be especially beneficial when compared to their single coordinate variants ($\mathbf{S} \in \mathbb{R}^{m\times 1}$) [42, 43, 52, 33]. However, in these cases the evaluation of an exact solution for the subproblem in the update rule can be computationally very expensive. In this work we propose and analyze inexact variants by allowing solving the subproblem that appears in the update rules of the stochastic methods inexactly. In particular, following the convention established in [54] of naming the main algorithm of the paper the *basic method*, we propose the *inexact basic method* (iBasic) (Algorithm 2.1).

Algorithm 2.1. Inexact basic method (iBasic).

Input: Distribution $\mathcal{D}$ from which we draw random matrices $\mathbf{S}$, positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$, stepsize $\omega > 0$.
Initialize: $x_0 \in \mathbb{R}^n$
for $k = 0, 1, 2, \ldots$ **do**
1: Generate a fresh sample $\mathbf{S}_k \sim \mathcal{D}$
2: Set $x_{k+1} = x_k - \omega\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k(\mathbf{S}_k^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k)^\dagger\mathbf{S}_k^\top(\mathbf{A}x_k - b) + \epsilon_k$
end for

The ϵ_k in the update rule of the method represents the abstract inexactness error described in subsection 2.1. Note that iBasic can have several equivalent interpretations. This allows us to study the methods (1.4), (1.5), (1.6) for solving the stochastic optimization problem and the SPM (1.8) for the best approximation problem in a single algorithm only. In particular, iBasic can be seen as inexact stochastic gradient descent (iSGD) with fixed stepsize applied to (1.1). From (1.17), $\nabla f_{\mathbf{S}_k}(x_k) = \mathbf{B}^{-1}\mathbf{A}^\top\mathbf{H}_k(\mathbf{A}x_k - b)$, and as a result the update rule of iBasic can be equivalently written as $x_{k+1} = x_k - \omega\nabla f_{\mathbf{S}_k}(x_k) + \epsilon_k$. In the case of the best approxi-

mation problem (1.7), iBasic can be interpreted as inexact SPM (iSPM) as follows:

$$\begin{aligned} x_{k+1} &= x_k - \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k (\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top (\mathbf{A}x_k - b) + \epsilon_k \\ &= \omega \left[x_k - \mathbf{B}^{-1}(\mathbf{S}_k^\top \mathbf{A})^\top (\mathbf{S}_k^\top \mathbf{A} B^{-1} (\mathbf{S}_k^\top \mathbf{A})^\top)^\dagger (\mathbf{S}_k^\top \mathbf{A}x - \mathbf{S}_k^\top b) \right] + (1-\omega)x_k + \epsilon_k \\ &\overset{(1.15)}{=} \omega \Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k) + (1-\omega)x_k + \epsilon_k. \end{aligned} \tag{2.1}$$

For the dual problem (1.10) we devote section 5 to presenting an inexact variant of the SDSA (iSDSA) and analyze its convergence using the rates obtained for iBasic in sections 3 and 4.

2.3. General framework and further special cases. The proposed inexact methods, iBasic (Algorithm 2.1) and iSDSA (section 5), belong in the general *sketch and project* framework, first proposed from Gower and Richtárik in [23] for solving consistent linear systems and where a unified analysis of several randomized methods was studied. This interpretation of the algorithms allows us to recover a comprehensive array of well-known methods as special cases by choosing carefully the combination of the main parameters of the algorithms.

In particular, iBasic has two main parameters (besides the stepsize $\omega > 0$ of the update rule). These are the distribution $\mathcal{D}$ from which we draw random matrices $\mathbf{S}$ and the positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$. By choosing carefully combinations of the parameters $\mathcal{D}$ and $\mathbf{B}$ we can recover several existing popular algorithms as special cases of the general method. For example, special cases of the exact basic method are randomized Kaczmarz, Gaussian Kaczmarz,[9] randomized coordinate descent, and their block variants. For more details about the generality of the sketch and project framework and further algorithms that can be cast as special cases of the analysis we refer the interested reader to section 3 of [23] and section 7 of [35]. Here we present only the inexact update rules of two special cases that we will later use in the numerical evaluation.

Special cases. Let $\mathbf{I}_{:C}$ denote the column concatenation of the $m \times m$ identity matrix indexed by a random subset C of $[m]$.

- *Inexact randomized block Kaczmarz (iRBK)*: Let $\mathbf{B} = \mathbf{I}$ and let us pick in each iteration the random matrix $\mathbf{S} = \mathbf{I}_{:C} \sim \mathcal{D}$. In this setup the update rule of iBasic simplifies to

$$x_{k+1} = x_k - \omega \mathbf{A}_{C:}^\top (\mathbf{A}_{C:}\mathbf{A}_{C:}^\top)^\dagger (\mathbf{A}_{C:}x_k - b_C) + \epsilon_k. \tag{2.2}$$

- *Inexact randomized block coordinate descent (iRBCD)*[10]: If the matrix $\mathbf{A}$ of the linear system is positive definite, then we can choose $\mathbf{B} = \mathbf{A}$. Let us also pick in each iteration the random matrix $\mathbf{S} = \mathbf{I}_{:C} \sim \mathcal{D}$. In this setup the update rule of iBasic simplifies to

$$x_{k+1} = x_k - \omega \mathbf{I}_{:C} (\mathbf{I}_{:C}^\top \mathbf{A} \mathbf{I}_{:C})^\dagger \mathbf{I}_{:C}^\top (\mathbf{A}x_k - b) + \epsilon_k. \tag{2.3}$$

For more papers related to the Kaczmarz method (randomized, greedy, cyclic update rules) we refer the interested reader to [28, 34, 46, 9, 45, 47, 13, 40, 42, 18, 38, 69, 43, 55, 59]. For the coordinate descent method (a.k.a. Gauss–Seidel for linear systems) and its block variant, randomized block coordinate descent, we suggest [30, 44, 52, 53, 48, 49, 51, 11, 29, 19, 1, 63].

[9] Special case of iBasic when the random matrix $\mathbf{S}$ is chosen to be a Gaussian vector with mean $0 \in R^m$ and a positive definite covariance matrix $\Sigma \in \mathbb{R}^{m\times m}$. That is, $\mathbf{S} \sim N(0, \Sigma)$ [23, 35].

[10] In the setting of solving linear systems randomized coordinate descent is known also as the Gauss–Seidel method. Its block variant can be also interpreted as the randomized coordinate Newton method (see [50]).

2.4. Other related work on inexact methods. One of the current trends in large scale optimization problems is the introduction of inexactness into the update rules of popular deterministic and stochastic methods. The rationale behind this is that an approximate/inexact step can often computed very efficiently and can have significant computational gains compared to its exact variants.

In the area of deterministic algorithms, the inexact variant of the full gradient descent method, $x_{k+1} = x_k - \omega_k[\nabla f(x_k) + \epsilon_k]$, has received a lot of attention [58, 15, 60, 21, 39]. It has been analyzed for the cases of convex and strongly convex functions under several meaningful assumptions on the inexactness error ϵ_k, and its practical benefit compared to exact gradient descent is apparent. For further deterministic inexact methods, check [14] for inexact Newton methods, [61, 56] for inexact proximal point methods, and [4] for inexact fixed point methods.

In recent years, with the explosion that is happening in areas like machine learning and data science, inexactness has also become part of the updating rules of several stochastic optimization algorithms, and many new methods have been proposed and analyzed.

In the large scale setting, stochastic optimization methods are preferred mainly because of their cheap per iteration cost (compared to their deterministic variants), their property to scale to extreme dimensions, and their improved theoretical complexity bounds. In areas like machine learning and data science, where the datasets become larger rapidly, the development of faster and efficient stochastic algorithms is crucial. For this reason, inexactness has recently been introduced into the update rules of several stochastic optimization algorithms, and new methods have been proposed and analyzed. One of the most interesting works on inexact stochastic algorithms appears in the area of second order methods—in particular, inexact variants of the sketch Newton method and subsampled Newton method for minimizing convex and nonconvex functions [57, 3, 7, 65, 66, 67]. Note that our results are also related to this literature since our algorithm can be seen as an inexact stochastic Newton method (see (1.5)). To the best of our knowledge, our work is the first to provide convergence analysis of inexact stochastic proximal point methods (see (1.6)) in any setting. From a numerical linear algebra viewpoint, inexact SPMs for solving the best approximation problem and its dual problem were also never analyzed before.

As we already mentioned, our framework is quite general, and many algorithms like iRBK (2.2) and iRBCD (2.3) can be cast as special cases. As a result, our general convergence analysis includes the analysis of inexact variants of all of these more specific algorithms as special cases. In [42] an analysis of the exact randomized block Kacmzarz method has been proposed, and in the experiments an inexact variant was used to speed up the method. However, no iteration complexity results were presented for the inexact variant, and both the analysis and the numerical evaluation were done for linear systems with full rank matrices that come with a natural partition of the rows (this is a much more restricted case than the one analyzed in our setting). For inexact variants of the randomized block coordinate descent algorithm in different settings than ours, we suggest consulting [62, 20, 10, 17].

Finally, an analysis of approximate stochastic gradient descent for solving the empirical risk minimization problem using quadratic constraints and sequential semidefinite programs has been presented in [27].

3. Convergence results under general assumptions. In this section we consider scenarios in which the inexactness error ϵ_k can be controlled, by specifying a per iteration bound σ_k on the norm of the error. In particular, by making different

assumptions on the bound σ_k we derive general convergence rate results. Our focus is on the abstract notion of inexactness described in section 2.1, and we make no assumptions on how this error is generated.

An important assumption that needs to hold in all of our results is exactness. A formal presentation is presented below. We state it here, and we highlight that it is a requirement for all of our convergence results (exactness is also required in the analysis of the exact variants [54]).

Exactness. Note that $f_{\mathbf{S}}$ is a convex quadratic and that $f_{\mathbf{S}}(x) = 0$ whenever $x \in \mathcal{L} := \{x \ : \ \mathbf{A}x = b\}$. However, $f_{\mathbf{S}}$ can be zero also for points x outside of $\mathcal{L}$. Clearly, $f(x)$ is nonnegative, and $f(x) = 0$ for $x \in \mathcal{L}$. However, without further assumptions, the set of minimizers of f can be larger than $\mathcal{L}$. The exactness assumption ensures that this does not happen. For necessary and sufficient conditions for exactness, we refer the reader to [54]. Here it suffices to remark that a sufficient condition for exactness is to require $\mathbb{E}[\mathbf{H}]$ to be positive definite. This is easy to see by observing that $f(x) = \mathbb{E}[f_{\mathbf{S}}(x)] = \frac{1}{2}\|\mathbf{A}x - b\|^2_{\mathbb{E}[\mathbf{H}]}$. In other words, if $\mathcal{X} = \operatorname{argmin} f(x)$ is the solution set of the stochastic optimization problem (1.1) and $\mathcal{L} = \{x : \mathbf{A}x = b\}$ is the solution set of the linear system (1.2), then the notion of exactness is captured by $\mathcal{X} = \mathcal{L}$.

3.1. Assumptions on inexactness error. In the convergence analysis of iBasic the following assumptions on the inexactness error are used. We note that Assumptions 3.2, 3.3, and 3.4 are special cases of Assumption 3.1. Moreover, Assumption 3.5 is algorithmic dependent and can hold in addition to any of the other four assumptions. In our analysis, depending on the result we aim at, we will require one of the first four assumptions either to hold by itself or to hold together with Assumption 3.5. We will always assume exactness.

In all assumptions the expectation on the norm of error ($\|\epsilon_k\|^2$) is conditioned on the value of the current iterate x_k and the random matrix $\mathbf{S}_k$. Moreover, it is worth mentioning that for the convergence analysis we never assume that the inexactness error has zero mean, that is, $\mathbb{E}[\epsilon_k] = 0$.

ASSUMPTION 3.1.

$$\mathbb{E}[\|\epsilon_k\|^2_{\mathbf{B}} \mid x_k, \mathbf{S}_k] \leq \sigma_k^2, \tag{3.1}$$

where the upper bound σ_k is a sequence of random variables (that can possibly depend on both the value of the current iterate x_k and the choice of the random $\mathbf{S}_k$ at the k^{th} iteration).

The following three assumptions on the sequence of upper bounds are more restricted, however, as we will later see, allowing us to obtain stronger and more controlled results.

ASSUMPTION 3.2.

$$\mathbb{E}\left[\|\epsilon_k\|^2_{\mathbf{B}} \mid x_k, \mathbf{S}_k\right] \leq \sigma_k^2, \tag{3.2}$$

where the upper bound $\sigma_k \in \mathbb{R}$ is a sequence of real numbers.

ASSUMPTION 3.3.

$$\mathbb{E}\left[\|\epsilon_k\|^2_{\mathbf{B}} \mid x_k, \mathbf{S}_k\right] \leq \sigma_k^2 = q^2\|x_k - x_*\|^2_{\mathbf{B}}, \tag{3.3}$$

where the upper bound is a special sequence that depends on a nonnegative inexactness parameter q and the distance to the optimal value $\|x_k - x_*\|^2_{\mathbf{B}}$.

ASSUMPTION 3.4.

$$\mathbb{E}\left[\|\epsilon_k\|_{\mathbf{B}}^2 \mid x_k, \mathbf{S}_k\right] \leq \sigma_k^2 = 2q^2 f_{\mathbf{S}_k}(x_k), \tag{3.4}$$

where the upper bound is a special sequence that depends on a nonnegative inexactness parameter q and the value of the stochastic function $f_{\mathbf{S}_k}$ computed at the iterate x_k.

Finally, the next assumption on the relationship between the inexactness error ϵ_k and a certain random vector arising from the iterative process allows us to obtain fast rates. In particular, we assume that the inexactness error is orthogonal with respect to the $\mathbf{B}$-inner product to the vector $\Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k) - x_* = (\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k - x_*)$.

ASSUMPTION 3.5.

$$\mathbb{E}\left[\left\langle (\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k - x_*), \epsilon_k \right\rangle_{\mathbf{B}}\right] = 0. \tag{3.5}$$

This assumption clearly holds if there is no error (exact case) or if the error is independent of the x_k and $\mathbf{Z}_k$ in expectation (inexactness resulting from independent noise). However, this assumption also holds naturally when inexactness arises by applying an arbitrary iterative technique to a certain auxiliary subproblem; see Algorithm 4.1. This is the main reason why this assumption makes sense.

3.2. Convergence results. In this section we present the analysis of the convergence rates of iBasic by assuming several combinations of the previous presented assumptions.

All convergence results are described only in terms of convergence of the iterates x_k, that is, $\|x_k - x_*\|_{\mathbf{B}}^2$, and not the objective function values $f(x_k)$. This is sufficient, because by $f(x) \leq \frac{\lambda_{\max}}{2}\|x - x_*\|_{\mathbf{B}}^2$ (see Lemma A.1) we can directly deduce a convergence rate for the function values.

The exact basic method (Algorithm 2.1 with $\epsilon_k = 0$), has been analyzed in [54], and it was shown to converge with $\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}^2] \leq \rho^k \|x_0 - x_*\|_{\mathbf{B}}^2$ where $\rho = 1 - \omega(2-\omega)\lambda_{\min}^+$. Our analysis of iBasic is more general and includes the convergence of the exact basic method as a special case when we assume that the upper bound is $\sigma_k = 0$ for all $k \geq 0$. For brevity, in the convergence analysis results of this manuscript we also use

$$\rho := 1 - \omega(2-\omega)\lambda_{\min}^+.$$

Let us start by presenting the convergence of iBasic when only Assumption 3.2 holds for the inexactness error.

THEOREM 3.6. *Let us assume exactness and let $\{x_k\}_{k=0}^{\infty}$ be the iterates produced by iBasic with $\omega \in (0,2)$. Set $x_* = \Pi_{\mathcal{L},\mathbf{B}}(x_0)$ and consider the error ϵ_k to be such that it satisfies Assumption* 3.2. *Then*

$$\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}] \leq \rho^{k/2}\|x_0 - x_*\|_{\mathbf{B}} + \sum_{i=0}^{k-1} \rho^{\frac{k-1-i}{2}} \sigma_i. \tag{3.6}$$

Proof. See Appendix B.1. ∎

COROLLARY 3.7. *In the special case that the upper bound σ_k in Assumption* 3.2 *is fixed, that is, $\sigma_k = \sigma$ for all $k > 0$, inequality* (3.6) *of Theorem* 3.6 *takes the following form:*

$$\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}] \leq \rho^{k/2}\|x_0 - x_*\|_{\mathbf{B}} + \sigma\frac{\rho^{1/2}}{1-\rho}. \tag{3.7}$$

This means that we obtain a linear convergence rate up to a solution level that is proportional to the upper bound σ.[11]

Proof. See Appendix B.2. ∎

Inspired by [21], let us now analyze iBasic using the sequence of upper bounds that are described in Assumption 3.3. This construction of the upper bounds allows us to obtain stronger and more controlled results. In particular, using the upper bound of Assumption 3.3 the sequence of expected errors converges linearly to the exact x_* (not in a potential neighborhood like the previous result). In addition, Assumption 3.3 guarantees that the distance to the optimal solution reduces with the increase of the number of iterations. However, for this stronger convergence a bound for $\lambda_{\min}^+$ is required, a quantity that in many problems is unknown to the user or intractable to compute. Nevertheless, there are cases in which this value has a closed form expression and can be computed beforehand without any further cost. See, for example, [33, 36, 32, 26], where methods for solving the average consensus were presented and the value of $\lambda_{\min}^+$ corresponds to the algebraic connectivity of the network under study.

THEOREM 3.8. *Assume exactness. Let $\{x_k\}_{k=0}^{\infty}$ be the iterates produced by iBasic with $\omega \in (0,2)$. Set $x_* = \Pi_{\mathcal{L},\mathbf{B}}(x_0)$ and consider the inexactness error ϵ_k to be such that it satisfies Assumption* 3.3*, with $0 \leq q < 1 - \sqrt{\rho}$. Then*

$$\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}^2] \leq (\sqrt{\rho} + q)^{2k} \|x_0 - x_*\|_{\mathbf{B}}^2. \tag{3.8}$$

Proof. See Appendix B.3. ∎

According to Theorem 3.8, to guarantee linear convergence, the *inexact parameter* q should live in the interval $\left[0, 1 - \sqrt{\rho}\right)$. In particular, q is the parameter that controls the level of inexactness of Algorithm 2.1. Not surprisingly, the fastest convergence rate is obtained when $q = 0$; in such a case the method becomes equivalent to its exact variant, and the convergence rate simplifies to $\rho = 1 - \omega(2-\omega)\lambda_{\min}^+$. Note also that, similar to the exact case, the optimal convergence rate is obtained for $\omega = 1$ [54].

Moreover, the upper bound σ_k of Assumption 3.3 depends on two important quantities, the $\lambda_{\min}^+$ (through the upper bound of the inexactness parameter q) and the distance to the optimal solution $\|x_k - x_*\|_{\mathbf{B}}^2$. Thus, it can have a natural interpretation. In particular, the inexactness error is allowed to be large either when the current iterate is far from the optimal solution ($\|x_k - x_*\|_{\mathbf{B}}^2$ large) or when the problem is well conditioned and $\lambda_{\min}^+$ is large. In the opposite scenario, when we have an ill conditioned problem or we are already close enough to the optimum x_* we should be more careful and allow fewer errors in the updates of the method.

In the next theorem we provide the complexity results of iBasic in the case that Assumption 3.5 is satisfied combined with one of the previous assumptions.

THEOREM 3.9. *Let us assume exactness and let $\{x_k\}_{k=0}^{\infty}$ be the iterates produced by iBasic with $\omega \in (0,2)$. Set $x_* = \Pi_{\mathcal{L},\mathbf{B}}(x_0)$. Let us also assume that the inexactness error ϵ_k is such that it satisfies Assumption* 3.5*. Then the following hold:*

[11] Several similar, more specific assumptions can be made for the upper bound σ_k. For example, if the upper bound satisfies $\sigma_k = \sigma^k$ with $\sigma \in (0,1)$ for all $k > 0$, then it can be shown that $C \in (0,1)$ exist such that inequality (3.6) of Theorem 3.6 takes the form $\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}] \leq O(C^k)$ (see [60, 21] for similar results).

(i) *If Assumption* 3.1 *holds,*

$$\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}^2] \leq \rho^k \|x_0 - x_*\|_{\mathbf{B}}^2 + \sum_{i=0}^{k-1} \rho^{k-1-i} \bar{\sigma}_i^2, \tag{3.9}$$

where $\bar{\sigma}_i^2 = \mathbb{E}[\sigma_i^2] \, \forall i \in [k-1]$.

(ii) *If Assumption* 3.3 *holds with* $q \in \left(0, \sqrt{\rho}\right)$,

$$\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}^2] \leq (\rho + q^2)^k \|x_0 - x_*\|_{\mathbf{B}}^2. \tag{3.10}$$

(iii) *If Assumption* 3.4 *holds with* $q \in (0, \sqrt{\omega(2-\omega)})$,

$$\begin{aligned} \mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}^2] &\leq (1 - (\omega(2-\omega) - q^2)\lambda_{\min}^+)^k \|x_0 - x_*\|_{\mathbf{B}}^2 \\ &= (\rho + q^2 \lambda_{\min}^+)^k \|x_0 - x_*\|_{\mathbf{B}}^2. \end{aligned} \tag{3.11}$$

Proof. See Appendix B.4. ∎

Remark 3.10. In the case that Assumptions 3.2 and 3.5 hold simultaneously, the convergence of iBasic is similar to (3.9), but in this case $\bar{\sigma}_i^2 = \sigma_i^2 \, \forall i \in [k-1]$ (due to Assumption 3.2, $\sigma_k \in \mathbb{R}$ is a sequence of real numbers). In addition, note that for $q \in (0, \min\{\sqrt{\rho}, 1 - \sqrt{\rho}\})$ having Assumption 3.5 on top of Assumption 3.3 leads to improvement of the convergence rate. In particular, from Theorem 3.8, iBasic converges with rate $(\sqrt{\rho} + q)^2 = \rho + q^2 + 2\sqrt{\rho}q$, while having both assumptions this is simplified to the faster $\rho + q^2$ (3.10).

4. iBasic with structured inexactness error. Up to this point, the analysis of iBasic was focused on more general abstract cases where the inexactness error ϵ_k of the update rule satisfies several general assumptions. In this section we focus on a more structured form of inexactness error, and we provide convergence analysis in the case that a linearly convergent algorithm is used for the computation of the expensive key subproblem of the method.

4.1. Linear system in the update rule. As we already mentioned in section 2.1, the update rule of the exact basic method (Algorithm 2.1 with $\epsilon_k = 0$) can be expressed as $x_{k+1} = x_k + \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_k^*$, where $\lambda_k^* = (\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top (b - \mathbf{A}x_k)$.

Using this expression the exact basic method can be equivalently interpreted as the following two-step procedure:

1. Find the least norm solution[12] of $\underbrace{\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k}_{\mathbf{M}_k} \lambda = \underbrace{\mathbf{S}_k^\top (b - \mathbf{A}x_k)}_{d_k}$. That is, find $\lambda_k^* = \arg\min_{\lambda \in \mathcal{Q}_k} \|\lambda\|$, where $\mathcal{Q}_k = \{\lambda \in \mathbb{R}^q : \mathbf{M}_k \lambda = d_k\}$.
2. Compute the next iterate: $x_{k+1} = x_k + \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_k^*$.

In the case that the random matrix $\mathbf{S}_k$ is large (this is the case that we are interested in), solving exactly the linear system $\mathbf{M}_k \lambda = d_k$ in each step can be prohibitively expensive. To reduce this cost we allow the inner linear system $\mathbf{M}_k \lambda = d_k$ to be solved inexactly using an iterative method. In particular, we propose and analyze the following inexact algorithm.

[12] We are precisely looking for the least norm solution of the linear system $\mathbf{M}_k \lambda = d_k$ because this solution can be written down in a compact way using the Moore–Penrose pseudoinverse. This is equivalent with the expression that appears in our update $\lambda_k^* = (\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top (b - \mathbf{A}x_k) = \mathbf{M}_k^\dagger d_k$. However, it can be easily shown that the method will still converge with the same rate of convergence even if we choose any other solution of the linear system $\mathbf{M}_k \lambda = d_k$.

Algorithm 4.1. iBasic with structured inexactness error.

Input: Distribution $\mathcal{D}$ from which we draw random matrices $\mathbf{S}$, positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$, stepsize $\omega > 0$.
Initialize: $x_0 \in \mathbb{R}^n$
for $k = 0, 1, 2, \ldots$ **do**

1: Generate a fresh sample $\mathbf{S}_k \sim \mathcal{D}$
2: Using an iterative method compute an approximation $\lambda_k^{\approx}$ of the least norm solution of the linear system:

$$\underbrace{\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k}_{\mathbf{M}_k} \lambda = \underbrace{\mathbf{S}_k^\top (b - \mathbf{A}x_k)}_{d_k}. \tag{4.1}$$

3: Set $x_{k+1} = x_k + \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_k^{\approx}$.
end for

For the computation of the inexact solution of the linear system (4.1) any known iterative method for solving general linear systems can be used. In our analysis we focus on linearly convergent methods. For example, based on the properties of the linear system (4.1), conjugate gradient (CG) or the sketch and project method (SPM) can be used for the execution of step 2. In these cases, we name Algorithm 4.1 *InexactCG* and *InexactSP*, respectively.

It is known that the classical CG can solve linear systems with positive definite matrices. In our approach, matrix $\mathbf{M}_k$ is positive definite only when the original linear system $\mathbf{A}x = b$ has full rank matrix $\mathbf{A}$. On the other side, SPM can solve any consistent linear system and as a result can solve the inner linear system $\mathbf{M}_k\lambda_k = d_k$ without any further assumption on the original linear system. In this case, one should be careful because the system has no unique solution. We are interested in finding the least norm solution of $\mathbf{M}_k\lambda_k = d_k$, which means that the starting point of SPM at the k^{th} iteration should always be $\lambda_k^0 = 0$. Recall that any special case of SPM (section 2.3) solves the best approximation problem.

Note that several classical solvers can be used for solving the symmetric linear system (4.1). For example, popular solvers like MINRES, LSQR, and LSMR could also be good choices. For more details on the convergence properties of these methods, we refer the interested reader to [6]. Later, in section 6, we numerically evaluate the performance of Algorithm 4.1 when these methods are used for the computation of the approximate solution $\lambda_k^{\approx}$ of the linear system (4.1).

Let us now define λ_k^r to be the approximate solution $\lambda_k^{\approx}$ of the $q \times q$ linear system (4.1) obtained after r steps of the linearly convergent iterative method. Using this, the update rule of Algorithm 4.1 takes the form

$$x_{k+1} = x_k + \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_k^r. \tag{4.2}$$

Remark 4.1. The update rule (4.2) of Algorithm 4.1 is equivalent to the update rule of iBasic (Algorithm 2.1) when the error ϵ_k is chosen to be

$$\epsilon_k = \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k(\lambda_k^r - \lambda_k^*). \tag{4.3}$$

This is precisely the connection between the abstract and more concrete/structured notion of inexactness first presented in Table 2.

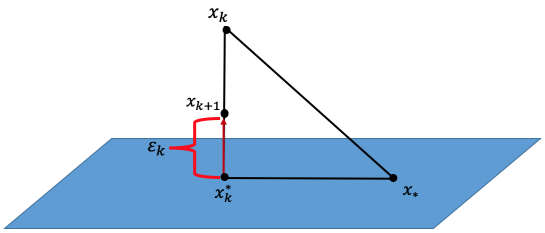


FIG. 1. *Graphical interpretation of orthogonality (justifies* (4.4)*). It shows that the two vectors,* $x_k^* - x_*$ *and* ϵ_k*, are orthogonal complements of each other with respect to the* $\mathbf{B}$*-inner product.* x_{k+1} *is the point that Algorithm* 4.1 *computes in each step. The colored region represents the* $Null(\mathbf{S}_k^\top \mathbf{A})$*.* $x_k^* = \Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k)$*,* $x_* = \Pi_{\mathcal{L},\mathbf{B}}(x_0)$ *and* ϵ_k *is the inexactness error.*

We now state a result that will be useful for the analysis in this section. The result says that Algorithm 4.1 with unit stepsize satisfies the general Assumption 3.5 presented in section 3.1.

LEMMA 4.2. *Let us denote by* $x_k^* = \Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k)$ *the projection of* x_k *onto* $\mathcal{L}_{\mathbf{S}_k}$ *in the* $\mathbf{B}$*-norm and* $x_* = \Pi_{\mathcal{L},\mathbf{B}}(x_0)$*. Let also assume that* $\omega = 1$ *(unit stepsize). Then for the updates of Algorithm* 4.1 *it holds that*

$$\langle x_k^* - x_*, \epsilon_k \rangle_{\mathbf{B}} = \left\langle (\mathbf{I} - \omega \mathbf{B}^{-1}\mathbf{Z}_k)(x_k - x_*), \epsilon_k \right\rangle_{\mathbf{B}} = 0 \quad \forall k \geq 0. \tag{4.4}$$

Proof. Note that $x_k^* - x_* = x_k - \nabla f_{\mathbf{S}_k}(x_k) - x_* \in Null(\mathbf{S}_k^\top \mathbf{A})$. Moreover, $\epsilon_k \overset{(4.3)}{=} \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k(\lambda_k^r - \lambda_k^*) \in Range(\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)$. From the knowledge that the null space of an arbitrary matrix is the orthogonal complement of the range space of its transpose, we have that $Null(\mathbf{S}_k^\top \mathbf{A})$ is orthogonal with respect to the $\mathbf{B}$-inner product to $Range(\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)$. This completes the proof (see Figure 1 for the graphical interpretation). ∎

4.2. Sketch and project interpretation. Let us now give a different interpretation of the inexact update rule of Algorithm 4.1 using the sketch and project approach. This will make us better appreciate the importance of the dual viewpoint and make clear the connection between the primal and dual methods.

Recall that in the special case of unit stepsize (see (1.9)) the exact SPM performs updates of the form

$$x_{k+1} = \arg\min_{x \in \mathbb{R}^n} \tfrac{1}{2}\|x - x_k\|_{\mathbf{B}}^2 \quad \text{subject to} \quad \mathbf{S}_k^\top \mathbf{A} x = \mathbf{S}_k^\top b. \tag{4.5}$$

That is, a *sketched* system $\mathbf{S}^\top \mathbf{A} x = \mathbf{S}^\top b$ is first chosen, and then the next iterate is computed by making a projection of the current iterate x_k onto this system.

In general, executing a projection step is one of the most common tasks in numerical linear algebra/optimization literature. However, in the large scale setting even this task can be prohibitively expensive, and it can be difficult to execute inexactly. For this reason, we suggest moving to the dual space, where the inexactness can be easily controlled.

Observe that the update rule of (4.5) has the same structure as the best approximation problem (1.7), where the linear system under study is the sketched system $\mathbf{S}_k^\top \mathbf{A} x = \mathbf{S}_k^\top b$ and the starting point is the current iterate x_k. Hence we can easily

compute its dual:

$$\max_{\lambda\in\mathbb{R}^q} D_k(\lambda) := (\mathbf{S}_k^\top b - \mathbf{S}_k^\top \mathbf{A}x_k)^\top \lambda - \tfrac{1}{2}\|\mathbf{A}^\top \mathbf{S}_k\lambda\|^2_{\mathbf{B}^{-1}}, \tag{4.6}$$

where $\lambda \in \mathbb{R}^q$ is the dual variable. The λ_k^* (possibly more than one) that solves the dual problem in each iteration k is the one that satisfies $\nabla D_k(\lambda_k^*) = 0$. By computing the derivative this is equivalent to finding the λ that satisfies the linear system $\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k\lambda = \mathbf{S}_k^\top(b - \mathbf{A}x_k)$. This is the same linear system we desire to solve inexactly in Algorithm 4.1. Thus, computing an inexact solution $\lambda_k^{\approx}$ of the linear system is equivalent to computing an inexact solution of the dual problem (4.6). Then by using the affine mapping (1.13) that connects the primal and the dual spaces we can also evaluate an inexact solution of the original primal problem (4.5).

The following result relates the inexact levels of these quantities. In particular, it shows that dual suboptimality of λ_k in terms of dual function values is equal to the distance of the dual values λ_k in the $\mathbf{M}_k$-norm.

LEMMA 4.3. *Let $\lambda_k^* \in \mathbb{R}^q$ be the exact solution of the linear system*

$$\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k\lambda = \mathbf{S}_k^\top(b - \mathbf{A}x_k)$$

or, equivalently, of the dual problem (4.6)*. Denoting by $\lambda_k^{\approx} \in \mathbb{R}^q$ the inexact solution, we have*

$$D_k(\lambda_k^*) - D_k(\lambda_k^{\approx}) = \tfrac{1}{2}\|\lambda_k^{\approx} - \lambda_k^*\|^2_{\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k}.$$

Proof.

$$\begin{aligned}
D_k(\lambda_k^*) - D_k(\lambda_k^{\approx}) &\overset{(4.6)}{=} [\mathbf{S}_k^\top b - \mathbf{S}_k^\top \mathbf{A}x_k]^\top[\lambda_k^* - \lambda_k^{\approx}] - \tfrac{1}{2}(\lambda_k^*)^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k\lambda_k^* \\
&\qquad + \tfrac{1}{2}(\lambda_k^{\approx})^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k\lambda_k^{\approx} \\
&\overset{(1.13)}{=} (\lambda_k^*)^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k[\lambda_k^* - \lambda_k^{\approx}] - \tfrac{1}{2}(\lambda_k^*)^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k\lambda_k^* \\
&\qquad + \tfrac{1}{2}(\lambda_k^{\approx})^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k\lambda_k^{\approx} \\
&= \tfrac{1}{2}(\lambda_k^{\approx} - \lambda_k^*)^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k(\lambda_k^{\approx} - \lambda_k^*) \\
&= \tfrac{1}{2}\|\lambda_k^{\approx} - \lambda_k^*\|^2_{\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k},
\end{aligned}$$

where in the second equality we use (1.13) to connect the optimal solutions of (4.5) and (4.6) and obtain $[\mathbf{S}_k^\top b - \mathbf{S}_k^\top \mathbf{A}x_k]^\top = (\lambda_k^*)^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k$. ∎

4.3. Complexity results. In this part we analyze the performance of Algorithm 4.1 when a linearly convergent iterative method is used for solving inexactly the linear system (4.1) in step 2 of Algorithm 4.1. We denote by λ_k^r the approximate solution of the linear system after we run the iterative method for r steps.

Before stating the main convergence result, let us present a lemma that summarizes some observations that are true in our setting.

LEMMA 4.4. *Let $\lambda_k^* = (\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top(b - \mathbf{A}x_k)$ be the exact solution and λ_k^r be the approximate solution of the linear system* (4.1)*. Then, $\|\lambda_k^*\|^2_{\mathbf{M}_k} = 2f_{\mathbf{S}_k}(x_k)$ and $\|\epsilon_k\|^2_{\mathbf{B}} = \|\lambda_k^r - \lambda_k^*\|^2_{\mathbf{M}_k}$.*

Proof. We have

$$\begin{aligned}\|\lambda_k^*\|^2_{\mathbf{M}_k} &= \|\mathbf{M}_k^\dagger \mathbf{S}_k^\top \mathbf{A}(x_* - x_k)\|^2_{\mathbf{M}_k} \\ &= (x_k - x_*)^\top \mathbf{A}^\top \mathbf{S}_k \underbrace{\mathbf{M}_k^\dagger \mathbf{M}_k \mathbf{M}_k^\dagger}_{\mathbf{M}_k^\dagger} \mathbf{S}_k^\top \mathbf{A}(x_k - x_*) \\ &\overset{(1.16)}{=} (x_k - x_*)^\top \mathbf{Z}_k (x_k - x_*) \overset{(1.17)}{=} 2 f_{\mathbf{S}_k}(x_k), \end{aligned} \tag{4.7}$$

and, moreover,

$$\begin{aligned}\|\epsilon_k\|^2_{\mathbf{B}} &\overset{\text{Remark 4.1}}{=} \|\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k(\lambda_k^r - \lambda_k^*)\|^2_{\mathbf{B}} \\ &= \|\lambda_k^r - \lambda_k^*\|^2_{\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k} = \|\lambda_k^r - \lambda_k^*\|^2_{\mathbf{M}_k}. \end{aligned} \tag{4.8}$$

∎

THEOREM 4.5. *Let us assume that for the computation of the inexact solution of the linear system* (4.1) *in step* 2 *of Algorithm* 4.1*, a linearly convergent iterative method is chosen such that*[13]

$$\mathbb{E}[\|\lambda_k^r - \lambda_k^*\|^2_{\mathbf{M}_k} \mid x_k, \mathbf{S}_k] \le \rho^r_{\mathbf{S}_k} \|\lambda_k^0 - \lambda_k^*\|^2_{\mathbf{M}_k}, \tag{4.9}$$

where $\lambda_k^0 = 0$ *for any* $k > 0$ *and* $\rho_{\mathbf{S}_k} \in (0,1)$ *for every choice of* $\mathbf{S}_k \sim \mathcal{D}$*. Let exactness hold, and let* $\{x_k\}_{k=0}^\infty$ *be the iterates produced by Algorithm* 4.1 *with unit stepsize* ($\omega = 1$)*. Set* $x_* = \Pi_{\mathcal{L},\mathbf{B}}(x_0)$*. Suppose further that there exists a scalar* $\theta < 1$ *such that with probability* 1*,* $\rho_{\mathbf{S}_k} \le \theta$*. Then, Algorithm* 4.1 *converges linearly with*

$$\mathbb{E}[\|x_k - x_*\|^2_{\mathbf{B}}] \le \left[1 - (1-\theta^r)\,\lambda_{\min}^+\right]^k \|x_0 - x_*\|^2_{\mathbf{B}}.$$

Proof. Theorem 4.5 can be interpreted as a corollary of the general Theorem 3.9(iii). Thus, it is sufficient to show that Algorithm 4.1 satisfies Assumptions 3.4 and 3.5. First, note that from Lemma 4.2, Assumption 3.5 is true. Moreover,

$$\begin{aligned}\mathbb{E}[\|\epsilon_k\|^2_{\mathbf{M}_k} \mid x_k, \mathbf{S}_k] &\overset{(4.8)}{=} \mathbb{E}[\|\lambda_k^r - \lambda_k^*\|^2_{\mathbf{M}_k} \mid x_k, \mathbf{S}_k] \overset{(4.9)}{\le} \rho^r_{\mathbf{S}_k} \|\lambda_k^0 - \lambda_k^*\|^2_{\mathbf{M}_k} \\ &\le \theta^r \|\lambda_k^0 - \lambda_k^*\|^2_{\mathbf{M}_k} \overset{\lambda_k^0 = 0}{=} \theta^r \|\lambda_k^*\|^2_{\mathbf{M}_k} \overset{(4.7)}{=} 2\theta^r f_{\mathbf{S}_k}(x_k), \end{aligned}$$

which means that Assumption 3.4 also holds with $q = \theta^{r/2} \in (0,1)$. This completes the proof. ∎

Having presented the main result of this section, let us now state some remarks that will help us understand the convergence rate of the last theorem.

Remark 4.6. From its definition $\theta^r \in (0,1)$ and as a result $(1-\theta^r)\,\lambda_{\min}^+ \le \lambda_{\min}^+$. This means that the method converges linearly but always with a worse rate than its exact variant.

Remark 4.7. Let us assume that θ is fixed. Then, as the number of iterations in step 2 of the algorithm ($r \to \infty$) increases, $(1-\theta^r) \to 1$, and as a result the method behaves similarly to the exact case.

Remark 4.8. The $\lambda_{\min}^+$ depends only on the random matrices $\mathbf{S} \sim \mathcal{D}$ and on the positive definite matrix $\mathbf{B}$ and is independent of the iterative process used in step 2. The iterative process of step 2 controls only the parameter θ of the convergence rate.

[13] In the case that the deterministic iterative method is used, like CG, we have that $\|\lambda_k^r - \lambda_k^*\|^2_{\mathbf{M}_k} \le \rho^r_{\mathbf{S}_k} \|\lambda_k^0 - \lambda_k^*\|^2_{\mathbf{M}_k}$, which is also true in expectation.

Remark 4.9. Let us assume that we run Algorithm 4.1 two separate times for two different choices of the linearly convergent iterative method of step 2. Let us also assume that the distribution $\mathcal{D}$ of the random matrices and the positive definite matrix $\mathbf{B}$ are the same for both instances and that for step 2 the iterative method runs for r steps for both algorithms. Let assume that $\theta_1 < \theta_2$; then we have that $\rho_1 = 1 - (1-\theta_1^r)\,\lambda_{\min}^+ < 1 - (1-\theta_2^r)\,\lambda_{\min}^+ = \rho_2$. This means that in the case that θ is easily computable, we should always prefer the inexact method with smaller θ.

The convergence of Theorem 4.5 is quite general, and it holds for any linearly convergent methods that can inexactly solve (4.1). However, in case that the iterative method is known, we can have more concrete results. See below the more specific results for the cases of CG and SPM.

Convergence of InexactCG. CG is a deterministic iterative method for solving linear systems $\mathbf{A}x = b$ with symmetric and positive definite matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ in a finite number of iterations. In particular, it can be shown that it converges to the unique solution in at most n steps. The worst case behavior of CG is given by [64, 22][14]

$$\|x_k - x_*\|_{\mathbf{A}} \leq \left(\frac{\sqrt{\kappa(\mathbf{A})}-1}{\sqrt{\kappa(\mathbf{A})}+1}\right)^{2k} \|x_0 - x_*\|_{\mathbf{A}}, \tag{4.10}$$

where x_k is the k^{th} iteration of the method and $\kappa(\mathbf{A})$ is the condition number of matrix $\mathbf{A}$.

Having presented the convergence of CG for general linear systems, let us now return to our setting. We denote $\lambda_k^r \in \mathbb{R}^q$ to be the approximate solution of the inner linear system (4.1) after r CG steps. Thus, using (4.10), we know that $\|\lambda_k^r - \lambda_k^*\|_{\mathbf{M}_k}^2 \leq \rho_{\mathbf{S}_k}^{4r}\|\lambda_k^0 - \lambda_k^*\|_{\mathbf{M}_k}^2$, where $\rho_{\mathbf{S}_k} = \frac{\sqrt{\kappa(\mathbf{M}_k)}-1}{\sqrt{\kappa(\mathbf{M}_k)}+1}$. Now, by making the same assumption as in the general Theorem 4.5, InexactCG converges as $\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}^2] \leq \left[1 - (1-\theta_{CG}^r)\,\lambda_{\min}^+\right]^k \|x_0 - x_*\|_{\mathbf{B}}^2$, where $\theta_{CG} < 1$ such that $\left(\frac{\sqrt{\kappa(\mathbf{M}_k)}-1}{\sqrt{\kappa(\mathbf{M}_k)}+1}\right)^4 \leq \theta_{CG}$ with probability 1.

Convergence of InexactSP. In this setting we suggest running SPM for solving inexactly the linear system (4.1). This allow us to have no assumptions on the structure of the original system $\mathbf{A}x = b$, and as a result we are able to solve more general problems than InexactCG can solve.[15] Like before, by making the same assumptions as in Theorem 4.5, the more specific convergence $\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}^2] \leq \left[1 - (1-\theta_{SP}^r)\,\lambda_{\min}^+\right]^k \|x_0 - x_*\|_{\mathbf{B}}^2$ for InexactSP can be obtained. Now the quantity $\rho_{\mathbf{S}_k}$ denotes the convergence rate of the exact basic method[16] when this is applied to solve linear system (4.1), and $\theta_{SP} < 1$ is a scalar such that $\rho_{\mathbf{S}_k} \leq \theta_{SP}$ with probability 1.

5. Inexact dual method. In the previous sections we focused on the analysis of inexact stochastic methods for solving the stochastic optimization problem (1.1)

[14] A sharper convergence rate of CG [64] for solving $\mathbf{A}x = b$ can also be used:

$$\|x_k - x_*\|_{\mathbf{A}}^2 \leq \left(\frac{\lambda_{n-k}-\lambda_1}{\lambda_{n-k}+\lambda_1}\right)^2 \|x_0 - x_*\|_{\mathbf{A}}^2,$$

where matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ has $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ eigenvalues.

[15] Recall that InexactCG requires the matrix $\mathbf{M}_k$ to be positive definite (this is true when matrix $\mathbf{A}$ is a full rank matrix).

[16] Recall that iBasic and its exact variant ($\epsilon_k = 0$) can be expressed as SPMs (2.1).

and the best approximation (1.7). In this section we turn to the dual of the best approximation (1.10), and we propose and analyze an inexact variant of the SDSA (1.11). We call the new method iSDSA and formalize it as Algorithm 5.1. In the update rule ϵ_k^d indicates the dual inexactness error that appears in the k^{th} iteration of iSDSA.

Algorithm 5.1. Inexact stochastic dual subspace ascent (iSDSA).

Input: Distribution $\mathcal{D}$ from which we draw random matrices $\mathbf{S}$, positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$, stepsize $\omega > 0$.
Initialize: $y_0 = 0 \in \mathbb{R}^m$, $x_0 \in \mathbb{R}^n$
for $k = 0, 1, 2, \ldots$ **do**
1: Generate a fresh sample $\mathbf{S}_k \sim \mathcal{D}$
2: Set $y_{k+1} = y_k + \omega \mathbf{S}_k \left(\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top \left(b - \mathbf{A}(x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k)\right) + \epsilon_k^d$
end for

5.1. Correspondence between the primal and dual methods. With the sequence of the dual iterates $\{y_k\}_{k=0}^\infty$ produced by the iSDSA we can associate a sequence of primal iterates $\{x_k\}_{k=0}^\infty$ using the affine mapping (1.13). In our first result we show that the random iterates produced by iBasic arise as an affine image of iSDSA under this affine mapping.

THEOREM 5.1 (correspondence between the primal and dual methods). *Let $\{x_k\}_{k=0}^\infty$ be the iterates produced by iBasic (Algorithm 2.1). Let $y_0 = 0$, and let $\{y_k\}_{k=0}^\infty$ be the iterates of the iSDSA. Assume that the two methods use the same stepsize $\omega > 0$ and the same sequence of random matrices $\mathbf{S}_k$. Assume also that $\epsilon_k = \mathbf{B}^{-1}\mathbf{A}^\top \epsilon_k^d$, where ϵ_k and ϵ_k^d are the inexactness errors that appear in the update rules of iBasic and iSDSA, respectively. Then*

$$x_k = \phi(y_k) = x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k$$

for all $k \geq 0$. That is, the primal iterates arise as affine images of the dual iterates.

Proof.

$$\begin{aligned}
\phi(y_{k+1}) &\overset{(1.13)}{=} x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_{k+1} \overset{(1.12),\text{Alg.5.1}}{=} x_0 + \mathbf{B}^{-1}\mathbf{A}^\top \left[y_k + \omega \mathbf{S}_k \lambda_k + \epsilon_k^d\right] \\
&\overset{(1.16),(1.12)}{=} \underbrace{x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k}_{\phi(y_k)} + \omega \mathbf{B}^{-1}\mathbf{Z}_k \left(x_* - (\underbrace{x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k}_{\phi(y_k)})\right) + \mathbf{B}^{-1}\mathbf{A}^\top \epsilon_k^d \\
&= \phi(y_k) - \omega \mathbf{B}^{-1}\mathbf{Z}_k(\phi(y_k) - x_*) + \mathbf{B}^{-1}\mathbf{A}^\top \epsilon_k^d.
\end{aligned}$$

Thus, by choosing the inexactness error of the primal method to be $\epsilon_k = \mathbf{B}^{-1}\mathbf{A}^\top \epsilon_k^d$, the sequence of vectors $\{\phi(y_k)\}$ satisfies the same recursion as the sequence $\{x_k\}$ defined by iBasic. It remains to check that the first elements of both recursions coincide. Indeed, since $y_0 = 0$, we have $x_0 = \phi(0) = \phi(y_0)$. ∎

5.2. iSDSA with structured inexactness error. In this subsection we present Algorithm 5.2. It can be seen as a special case of iSDSA but with a more structured inexactness error.

Similar to the primal variants, it can be easily checked that Algorithm 5.2 is a special case of the iSDSA (Algorithm 5.1) when the dual inexactness error is chosen

Algorithm 5.2. iSDSA with structured inexactness error.

Input: Distribution $\mathcal{D}$ from which we draw random matrices $\mathbf{S}$, positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$, stepsize $\omega > 0$.
Initialize: $y_0 = 0 \in \mathbb{R}^m$, $x_0 \in \mathbb{R}^n$
for $k = 0, 1, 2, \ldots$ **do**

1: Generate a fresh sample $\mathbf{S}_k \sim \mathcal{D}$
2: Using an iterative method compute an approximation $\lambda_k^{\approx}$ of the least norm solution of the linear system:

$$\underbrace{\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k}_{\mathbf{M}_k} \lambda = \underbrace{\mathbf{S}_k^\top (b - \mathbf{A}(x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k)}_{d_k} \tag{5.1}$$

3: Set $y_{k+1} = y_k + \omega \mathbf{S}_k \lambda_k^{\approx}$.
end for

to be $\epsilon_k^d = \mathbf{S}_k(\lambda_k^r - \lambda_k^*)$. Noting that, using the observation of Remark 4.1 that $\epsilon_k = \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k(\lambda_k^r - \lambda_k^*)$ and the above expression of ϵ_k^d, we can easily verify that the expression $\epsilon_k = \mathbf{B}^{-1}\mathbf{A}^\top \epsilon_k^d$ holds. This is precisely the connection between the primal and dual inexactness errors that have already been used in the proof of Theorem 5.1.

5.3. Convergence of dual function values. We are now ready to state a linear convergence result describing the behavior of the inexact dual method in terms of the function values $D(y_k)$. The following result is focused on the convergence of iSDSA by making an assumption similar to Assumption 3.3. Similar convergence results can be obtained using any other assumption of section 3.1. The convergence of Algorithm 5.2 can also be easily derived using arguments similar to the one presented in section 4 and the convergence guarantees of Theorem 4.5.

THEOREM 5.2 (convergence of dual objective). *Assume exactness. Let $y_0 = 0$ and let $\{y_k\}_{k=0}^\infty$ be the dual iterates of iSDSA (Algorithm 5.1) with $\omega \in (0, 2)$. Set $x_* = \Pi_{\mathcal{L},\mathbf{B}}(x_0)$ and let y_* be any dual optimal solution. Consider the inexactness error ϵ_k^d to be such that it satisfies $\mathbb{E}[\|\mathbf{B}^{-1}\mathbf{A}^\top \epsilon_k^d\|_{\mathbf{B}}^2 \mid y_k, \mathbf{S}_k] \le \sigma_k^2 = q^2 2\,[D(y_*) - D(y_k)]$, where $0 \le q < 1 - \sqrt{\rho}$. Then*

$$\mathbb{E}[D(y_*) - D(y_k)] \le \left(\sqrt{\rho} + q\right)^{2k} [D(y_*) - D(y_0)]\,. \tag{5.2}$$

Proof. The proof follows by applying Theorem 3.8 together with Theorem 5.1 and the identity $\frac{1}{2}\|x_k - x_*\|_{\mathbf{B}}^2 = D(y_*) - D(y_k)$ (1.14). ∎

Note that in the case that $q = 0$, iSDSA simplifies to its exact variant SDSA and the convergence rate coincides with the one presented in [35, 24]. Following arguments similar to those in [24], the same rate can be proved for the duality gap $\mathbb{E}[P(x_k) - D(y_k)]$.

6. Numerical evaluation. In this section we perform preliminary numerical tests for studying the computational behavior of iBasic with structured inexactness error when it is used to solve the best approximation problem (1.7) or equivalently the stochastic optimization problem (1.1).[17] As we have already mentioned, iBasic can be interpreted as an SPM, and as a result a comprehensive array of well-known

[17] Note that from section 5 and the correspondence between the primal and dual methods, iSDSA will have similar behavior when it is applied to the dual problem (1.10).

algorithms can be recovered as special cases by varying the main parameters of the methods (section 2.3). In particular, in the first part of our experiments (sections 6.1–6.4), we focus on the evaluation of two popular special cases, inexact randomized block Kaczmarz (iRBK) (see (2.2)) and inexact randomized block coordinate descent (iRBCD) (see (2.3)). Later in section 6.5, we explain how our general framework can capture block Gaussian methods as special cases, and we show that inexactness can be beneficial for these methods as well. In the last part of the numerical evaluation (section 6.6) we focus on randomized methods for solving the average consensus problem and describe how the proposed iRBK can work as an efficient randomized gossip algorithm. We implement Algorithm 4.1, presented in section 4, using several methods to inexactly solve the linear system of the update rule (see (4.1)). More specifically, for the experiments in sections 6.2, 6.3, and 6.5 we use CG[18] to solve the linear system (4.1). Recall that in this case we named the method InexactCG. For the experiments presented in sections 6.4 and 6.6 we evaluate the performance of Algorithm 4.1 when methods like MINRES, LSQR, LSMR, and randomized Kaczmarz (RK) are used for solving the linear system (4.1).

The convergence analysis of previous sections is quite general and holds for several combinations of the two main parameters of the method, the positive definite matrix $\mathbf{B}$ and the distribution $\mathcal{D}$ of the random matrices $\mathbf{S}$. For obtaining iRBK as special case, we have to choose $\mathbf{B} = \mathbf{I} \in \mathbb{R}^{n\times n}$ (identity matrix), and for the iRBCD the given matrix $\mathbf{A}$ should be positive definite and we choose $\mathbf{B} = \mathbf{A}$. For both methods the distribution $\mathcal{D}$ should be over random matrices $\mathbf{S} = \mathbf{I}_{:C}$, where $\mathbf{I}_{:C}$ is the column concatenation of the $m \times m$ identity matrix indexed by a random subset C of $[m]$. In our experiments we choose to have one specific distribution over these matrices. In particular, we assume that the random matrix in each iteration is chosen uniformly at random to be $\mathbf{S} = \mathbf{I}_{:d}$ with the subset d of $[m]$ having fixed prespecified cardinality.

Recently, two exotic variants of RBCD and RBK have been proposed in [24, 41] that use Gaussian matrices in their update rules: Gaussian block coordinate descent (GBCD) and Gaussian block Kaczmarz (GBK). As we will explain later, these two algorithms can be also obtained as special cases of our general framework by choosing the random matrix $\mathbf{S} \in \mathbb{R}^{m\times d}$ to be a standard Gaussian matrix with independent entries and the positive definite matrix $\mathbf{B}$ to be $\mathbf{B} = \mathbf{A}$ for GBCD and $\mathbf{B} = \mathbf{I} \in \mathbb{R}^{n\times n}$ for GBK. We use iGBCD and iGBK to denote the inexact variants of these methods. In the experiments related to Gaussian methods we assume that the subset d of $[m]$ has fixed prespecified cardinality.

The code for all experiments is written in the Julia 0.6.3 programming language and run on a Mac laptop computer (OS X El Capitan), 2.7 GHz Intel Core i5 with 8 GB of RAM.

In view of the theoretical convergence results of Algorithm 4.1, the relaxation parameter (stepsize) of the methods studied in our experiments is chosen to be $\omega = 1$ (no relaxation). In all implementations, except for the experiments on average consensus, we use $x_0 = 0 \in \mathbb{R}^n$ as an initial point. In the experiments on average consensus, the starting point must be the vector with the initial private values of the nodes of the network. In comparing the methods with their inexact variants, we use the relative error measure $\|x_k - x_*\|^2_{\mathbf{B}}/\|x_0 - x_*\|^2_{\mathbf{B}}$. We run each method (exact and

[18] Recall that in order to use CG, the matrix $\mathbf{M}_k$ that appears in linear system (4.1) should be positive definite. This is true in the case that the matrix $\mathbf{A}$ of the original system has a full column rank matrix. Note, however, that the analysis of section 4 holds for any consistent linear system $\mathbf{A}x = b$ and without making any further assumption on its structure or the linear convergence methods.

inexact) until the relative error is below 10^{-5}. For the horizontal axis we use either the number of iterations or the wall-clock time measured using the tic-toc Julia function. In the experiments on average consensus the horizontal axis represents the number of communications. In the exact variants, the linear system (4.1) in Algorithm 4.1 needs to be solved exactly. In our experiments we follow the implementation of [23] for exact RBCD, exact RBK, exact GBCD, and exact GBK, where the built-in direct solver (sometimes referred to as "backslash") is used.

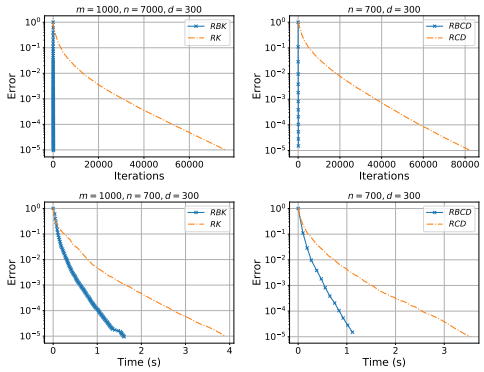
Experimental setup. For the construction of consistent linear systems $\mathbf{A}x = b$ we use the following setup:

- **For iRBK and iGBK:** Let matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ be given (it can be either synthetic or real data). Then a vector $z \in \mathbb{R}^n$ is chosen to be i.i.d. $\mathcal{N}(0,1)$ and the right-hand side of the linear system is set to $b = \mathbf{A}z$. In this way the consistency of the linear system with matrix $\mathbf{A}$ and right-hand side b is ensured.
- **For iRBCD and iGBCD:** *Synthetic data:* A Gaussian or sparse Gaussian matrix $\mathbf{P} \in \mathbb{R}^{m\times n}$ is generated, and then matrix $\mathbf{A} = \mathbf{P}^\top\mathbf{P} \in \mathbb{R}^{n\times n}$ is used in the linear system (in this way matrix $\mathbf{A}$ is positive definite with probability 1). The vector $z \in \mathbb{R}^n$ is chosen to be i.i.d. $\mathcal{N}(0,1)$, and again, to ensure consistency of the linear system, the right-hand side is set to $b = \mathbf{A}z$. *Real data:* Newton systems arising from ridge-regression problems using data from the library of support vector machine problems LIBSVM [12] (for more details see section 6.5.).

6.1. Importance of large block size. Many recent works have shown that using larger block sizes can be very beneficial for the performance of randomized iterative algorithms [23, 52, 42, 33]. In Figure 2 we numerically verify this statement. We show that both RBK and RBCD (no inexact updates) outperform, in number of iterations and wall-clock time, their serial variants where only one coordinate is chosen (block of size $d = 1$) per iteration. This justifies the necessity of choosing methods with large block sizes. Recall that this is precisely the class of algorithms that could have an expensive subproblem in their update rule which is required to be solved exactly and as a result can benefit the most from the introduction of inexactness.

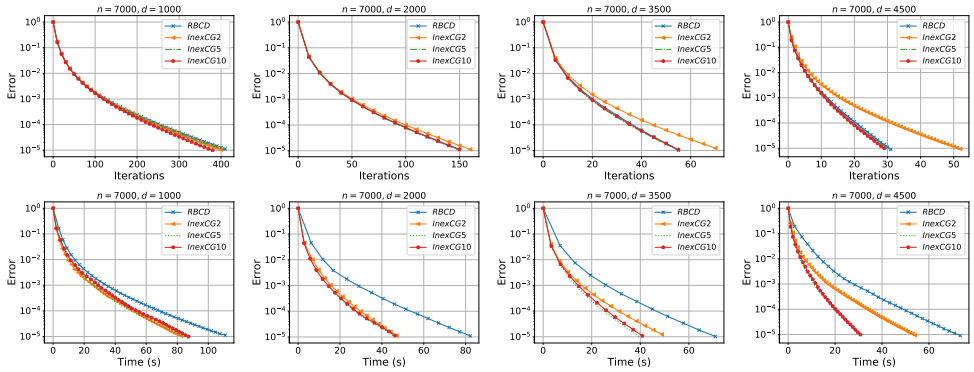
6.2. Inexactness and block size (iRBCD). In this experiment, we first construct a positive definite linear system following the previously described procedure for iRBCD. We first generate a Gaussian matrix $\mathbf{P} \in \mathbb{R}^{10000\times 7000}$ and then the positive definite matrix $\mathbf{A} = \mathbf{P}^\top\mathbf{P} \in \mathbb{R}^{7000\times 7000}$ is used to define a consistent liner system. We run iRBCD in this specific linear system and compare its performance with its exact variance for several block sizes d (numbers of columns of matrix $\mathbf{S}$). For evaluating the inexact solution of the linear system in the update rule we run CG for 2, 5, or 10 iterations. In Figure 3, we plot the evolution of the relative error in terms of both the number of iterations and the wall-clock time.

We observe that for any block size the inexact methods are always faster in terms of wall-clock time than their exact variants even if they require (as is expected) an equal or larger number of iterations. Moreover, it is obvious that the performance of the inexact method becomes much better than the exact variant as the size d increases, and as a result the subproblem that needs to be solved in each step becomes more expensive. It is worth highlighting that for the chosen systems, the exact RBCD behaves better in terms of wall-clock time as the block size increases (this coincides with the findings of the previous experiment).



(a) RK vs RBK (b) RCD vs RBCD

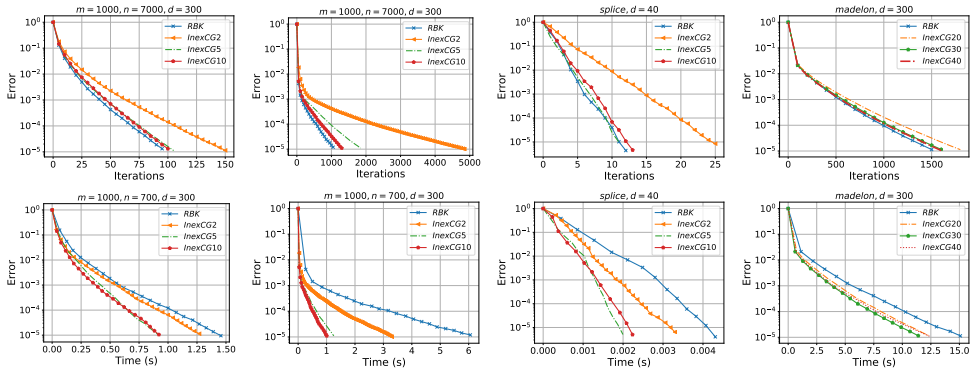
FIG. 2. *Comparison of the performance of the exact RBK and RBCD with their nonblock variants RK and RCD. For the Kaczmarz methods (first column)* $\mathbf{A} \in \mathbb{R}^{1000,700}$ *is a Gaussian matrix and for the coordinate descent methods (second column)* $\mathbf{A} = \mathbf{P}^\top \mathbf{P} \in \mathbb{R}^{700\times 700}$ *where* $\mathbf{P} \in \mathbb{R}^{1000\times 700}$ *is a Gaussian matrix. To guarantee consistency,* $b = \mathbf{A}z$ *where* z *is also a Gaussian vector. The block size that chosen for the block variants is* $d = 300$.



(a) d=1000 (b) d=2000 (c) d=3500 (d) d=4500

FIG. 3. *Performance of iRBCD (InexactCG) and exact RBCD for solving a consistent linear system with* $\mathbf{A} = \mathbf{P}^\top \mathbf{P} \in \mathbb{R}^{7000\times 7000}$, *where* $\mathbf{P} \in \mathbb{R}^{10000\times 7000}$ *is a Gaussian matrix. The right-hand side for the system is chosen to be* $b = \mathbf{A}z$ *where* z *is also a Gaussian vector. Several block sizes are used:* $d = 1000, 2000, 3500, 4500$. *The graphs in the first (second) row plot the iterations (time) against relative error* $\|x_k - x_*\|^2_{\mathbf{A}} / \|x_*\|^2_{\mathbf{A}}$.

6.3. Evaluation of iRBK. In the last experiment we evaluate the performance of iRBK in both synthetic and real datasets. For computing the inexact solution of the linear system in the update rule we run CG for a prespecified number of iterations that can vary depending on the datasets. In particular, we compare iRBK and RBK on synthetic linear systems generated with the Julia Gaussian matrix functions "randn(m,n)" and "sprandn(m,n,r)" (input r of the sprandn function indicates the density of the matrix). For the real datasets, we test the performance of iRBK and RBK using real matrices from the library of support vector machine problems LIBSVM [12]. Each dataset of the LIBSVM consists of a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ (m features and n characteristics) and a vector of labels $b \in \mathbb{R}^m$. In our experiments we choose



(a) randn(m,n) (b) sprandn(m,n,0.01) (c) splice (d) madelon

FIG. 4. *The performance of iRBK (InexactCG) and RBK on synthetic and real datasets. Synthetic matrices:* (a) *randn(m,n) with (m,n)*=(1000,700), (b) *sprandn(m,n,*0.01*) with (m,n)*=(1000,700). *Real matrices from the LIBSVM* [12]*:* (c) *Splice: (m,n)*=(1000,60). (d) *Madelon: (m,n)*=(2000,500). *The graphs in the first (second) row plot the iterations (time) against relative error* $\|x_k - x_*\|^2/\|x_*\|^2$. *The quantity d in the title of each plot indicates the block size for both iRBK and RBK.*

to use only the matrices of the datasets and ignore the label vectors.[19] As before, to ensure consistency of the linear system, we choose a Gaussian vector $z \in \mathbb{R}^n$, and the right-hand side of the linear system is set to $b = \mathbf{A}z$ (for both the synthetic and the real matrices). By observing Figure 4 it is clear that for all problems under study the performance of iRBK in terms of wall-clock time is much better than its exact variant RBK.

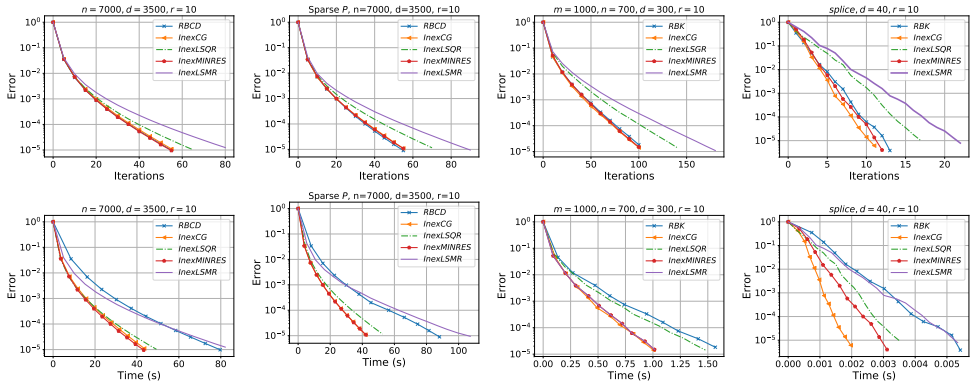
6.4. Different linear system solvers. As we have already mentioned in section 4.1, several classical linear system solvers can be used for solving the symmetric linear system (4.1) of Algorithm 4.1. In this experiment we numerically compare the performance of Algorithm 4.1 when several algorithms like CG, MINRES, LSQR, and LSMR are used for solving the linear system (4.1).[20]

The comparison is made for two special cases of iBasic, the iRBCD and iRBK. In particular, we compare the performance of the exact methods RBCD and RBK with their inexact variants InexCG, InexMINRES, InexLSQR, and InexLSMR for solving several linear systems. In the implementation of all inexact methods we use the same fixed number of iterations, $r = 10$, to approximately solve the linear system (4.1). Our findings are available in Figure 5. Note that for the linear systems under study, InexCG and InexMINRES are the best methods in terms of wall-clock time. InexLSGR is also faster than the corresponding exact method but not as fast as InexCG and InexMINRES. On the other hand, InexLSMR does not perform well in all linear systems under study. However, we speculate that for other datasets and by allowing LSMR to run for more iterations in each step, $r > 10$, InexLSMR can also be beneficial.

6.5. Block Gaussian sketching. Up to this point, in our numerical evaluation we focus on the performance of iRBK and iRBCD. However, as we have already

[19]Note that the real matrices of the Splice and Madelon datasets are full rank matrices.

[20]In the implementation of these methods we use the available built-in iterative linear system solvers of Julia.



(a) RBCD randn(m,n) (b) RBCD sprandn(m,n,0.01) (c) RBK randn(m,n) (d) RBK splice

FIG. 5. *Different linear system solvers: Comparison of iRBCD and iRBK when different solvers are used in their inexact update rules. InexCG, InexLSQR, InexMINRES, and InexLSMR denote the inexact variants of RBCD and RBK when CG, LSQR, MINRES, and LSMR are used to solve the potentially expensive linear system* (4.1). *The graphs in the first (second) row plot the iterations (time) against relative error* $\|x_k - x_*\|^2/\|x_*\|^2$. *The quantity* d *in the title of each plot indicates the block size for both inexact and noninexact methods. The quantity* r *indicates the number of iterations of each solver (*r *is selected to be the same for all methods).*

mentioned, our theoretical results are more general and capture several other more exotic methods as special cases. In this experiment, we evaluate the performance of the inexact variants of two recently proposed methods that use Gaussian matrices in their updates, Gaussian block coordinate descent (GBCD) and the Gaussian block Kaczmarz method (GBK). We use iGBCD and iGBK to denote the inexact variants of these methods.

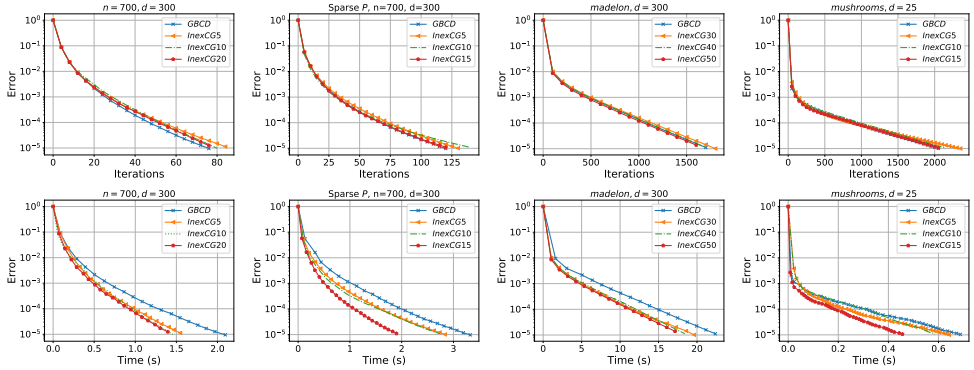
The Gaussian SPMs, GBCD and GBK, first proposed in [24] for solving consistent linear systems and a tight theoretical analysis of GBK, have been presented in [41] under a more general setting. In [23] it was shown that for these methods $\mathbb{E}[\mathbf{H}] \succeq 0$. As a result, the main assumption of *exactness* is satisfied and the theoretical results proposed in section 3.2 and 4.3 hold for the inexact variants of these methods (see discussion in section 3).

In Figure 6 we evaluate the performance of GBCD and its inexact variant iGBCD, on both synthetic and real datasets. For the synthetic datasets we first construct positive definite linear systems following the experimental setup described in the beginning of the section. We first generate a Gaussian matrix $\mathbf{P} \in \mathbb{R}^{1000\times 700}$, and then the positive definite matrix $\mathbf{A} = \mathbf{P}^\top\mathbf{P} \in \mathbb{R}^{700\times 700}$ is used to define a consistent linear system (for the figure (b) in Figure 6 we follow the same procedure but with sparse Gaussian matrix $\mathbf{P}$).

For the real datasets presented in Figure 6 we focus on ridge-regression problems. In particular, we use GBCD and iGBCD for solving the Newton system $\nabla^2 f(w_0)x = -\nabla f(w_0)$ arising from ridge-regression problems of the form

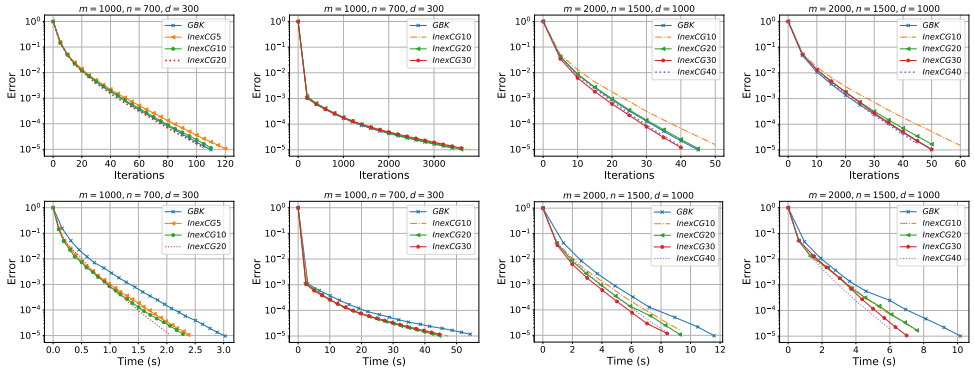
$$\min\left\{f(w) = \tfrac{1}{2}\|\mathbf{A}w - b\|^2 + \tfrac{\lambda}{2}\|w\|^2\right\},$$

using real data from the LIBSVM [12]. In particular, we set $w_0 = 0$ and use $\lambda = 1$ as the regularization parameter, whence $\nabla f(w_0) = \mathbf{A}^\top b$ and $\nabla^2 f(w_0) = \mathbf{A}^\top\mathbf{A} + \mathbf{I}$. Note that this is the same setting used in [23] for the evaluation of GBCD and guarantees



(a) randn(m,n) (b) sprandn(m,n,0.01) (c) madelon (d) mushrooms

FIG. 6. *Performance of iGBCD (InexactCG) and GBCD on synthetic and real datasets. Synthetic matrices:* $\mathbf{A} = \mathbf{P}^\top \mathbf{P} \in \mathbb{R}^{700\times 700}$, *where* $\mathbf{P} \in \mathbb{R}^{1000\times 700}$ *is* (a) *a Gaussian matrix or* (b) *sparse Gaussian. The right-hand side for the system is chosen to be* $b = \mathbf{A}z$, *where* z *is also a Gaussian vector. Real matrices from the LIBSVM* [12]*:* (c) *Madelon: (m,n)*=(2000,500). (d) *Mushrooms: (m,n)*=(8124,112). *The graphs in the first (second) row plot the iterations (time) against relative error* $\|x_k - x_*\|^2/\|x_*\|^2$. *The quantity d in the title of each plot indicates the block size for both iGBCD and GBCD.*



(a) randn(m,n) (b) sprandn(m,n,0.01) (c) randn(m,n) (d) sprandn(m,n,0.01)

FIG. 7. *Performance of iGBK (InexactCG) and GBK on synthetic datasets. Matrices of different sizes:* (a) *randn*(1000, 700), (b) *sprandn*(1000, 700, 0.01), (c) *randn*(2000, 1500) (d) *sprandn*(2000, 1500, 0.01). *The graphs in the first (second) row plot the iterations (time) against relative error* $\|x_k - x_*\|^2/\|x_*\|^2$. *The quantity d in the title of each plot indicates the block size for both iGBK and GBK.*

that the linear system has a positive definite matrix (a requirement for running GBCD and iGBCD).

For the numerical evaluation of GBK and its inexact variant iGBK we focus on synthetic datasets where matrix $\mathbf{A}$ of the consistent linear system is selected to be either a Gaussian or a sparse Gaussian matrix of different sizes (Figure 7). To guarantee consistency we follow the experimental setup described in the beginning of the section.

By observing Figures 6 and 7, it is clear that for all problems under study the performance of iGBCD and iGBK in terms of wall-clock time is much better than that of their exact variants GBCD and GBK.

6.6. Randomized block gossip algorithms and communication. Average consensus (AC) is a fundamental problem in distributed computing and multiagent systems [8, 16, 37]. Consider a connected undirected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V} = \{1, 2, \ldots, n\}$ and edges $\mathcal{E}$ ($|\mathcal{E}| = m$), where each node $i \in \mathcal{V}$ owns a private value $c_i \in \mathbb{R}$. The goal of the AC problem is for each node of the network to compute the average of these private values, $\bar{c} := \frac{1}{n}\sum_i c_i$, via a protocol which allows communication between neighbors only. The problem comes up in many real-world applications such as coordination of autonomous agents, estimation, rumor spreading in social networks, PageRank and distributed data fusion on ad hoc networks, and decentralized optimization.

Recently in [33, 36, 32, 37] it was shown how classical randomized iterative methods for solving linear systems can be interpreted as gossip algorithms when applied to special systems encoding the underlying network, and their decentralized nature was explained in detail. The starting point of the connection between the two areas of research is the observation that the AC problem can be expressed as an optimization problem as follows:

$$\min_{x=(x^1,\ldots,x^n)\in\mathbb{R}^n} P(x) := \tfrac{1}{2}\|x - x_0\|^2_{\mathbf{B}} \quad \text{subject to} \quad x^1 = x^2 = \cdots = x^n. \tag{6.1}$$

Note that problem (6.1) can be seen as a special case of the best approximation problem (1.7) when $\mathbf{A}$ and b are selected so that the constraint $\mathbf{A}x = b$ of problem (1.7) is equivalent to the requirement that $x^i = x^j$ (the value stored at node i is equal to the value stored at node j) for all $(i, j) \in \mathcal{E}$.

Following the derivation from [37], one example of a linear system that can be used to encode the constraints $x^i = x^j$ for $(i, j) \in \mathcal{E}$ is the homogeneous linear system ($b = 0$) with matrix $\mathbf{A} = \mathbf{Q} \in \mathbb{R}^{|\mathcal{E}|\times n}$ as the incidence matrix of $\mathcal{G}$. That is, row $e = (i, j) \in \mathcal{E}$ of matrix $\mathbf{Q}$ contains value 1 in column i, value -1 in column j (here an arbitrary but fixed order of node defining each edge is used in order to fix $\mathbf{Q}$), and zeros elsewhere.

Under this setting one of the algorithms proposed in [33, 37] for solving the AC problem is the randomized block Kaczmarz (RBK) method. In particular, it was shown that RBK can work as a block gossip algorithm as follows: In the k^{th} iteration of the method, (i) select a random set of edges $C \subseteq \mathcal{E}$;[21] (ii) form subgraph $\mathcal{G}_k$ of $\mathcal{G}$ from the selected edges; (iii) for each connected component of $\mathcal{G}_k$, replace node values with their average.

The gossip nature of the RBK has many applications, and many popular gossip algorithms like the *path averaging* algorithm [2] and clique gossiping [31] can be cast as special cases of this framework. However, the main drawback of this method is that in each step we assume simultaneous communication between all nodes of the activated subgraph $\mathcal{G}_k$. This could be possible in some special scenarios [68, 5], but normally in practice the nodes can only communicate with their direct neighbors. In other words, there are no guarantees on the number of communications required in each step (how many messages should be sent between the nodes of the subgraph $\mathcal{G}_k$ in order to update their values to their average).

The inexact RBK (iRBK) proposed in this work resolves this problem and allows us to compute the exact number of communications required to achieve the given

[21] Recall that in RBK, the random matrix is $\mathbf{S} = \mathbf{I}_{:C} \sim \mathcal{D}$.

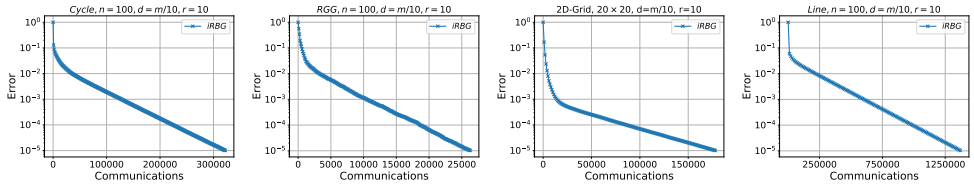


FIG. 8. *Performance of Inexact Randomized Block Gossip Algorithm (iRBG) for solving the average consensus problem in a cycle graph, line graph, random geometric graph (RGG), and 2D grid graph. The graphs plot the number of communications against the relative error* $\|x_k - x_*\|_{\mathbf{B}}^2/\|x_0 - x_*\|_{\mathbf{B}}^2 \overset{\mathbf{B}=\mathbf{I},x_0=c}{=} \|x_k - x_*\|^2/\|c - x_*\|^2$. *For all networks the starting vector* $x^0 = c \in \mathbb{R}^n$ *is a Gaussian vector. The* n *and* m *in the title of each plot indicate the number of nodes and edges of the network, respectively. For the 2D-grid graph this is* $n \times n$. *The quantities* d *and* r *in the title of each plot indicate the block size and the number of RK updates, respectively.*

accuracy.[22]

In particular, we suggest using RK in order to approximately solve the linear system (4.1) appearing in the update rule of iRBK (special case of Algorithm 4.1). In this way the iRBK method would work as gossip algorithm as follows: (i) Select a random set of edges $C \subseteq \mathcal{E}$; (ii) form subgraph $\mathcal{G}_k$ of $\mathcal{G}$ from the selected edges; (iii) for each connected component of $\mathcal{G}_k$, replace node values with their approximate average (value computed after running the RK method in the subgraph $\mathcal{G}_k$ for r number of iterations). In this case, each iteration of the RK method is equivalent to one message exchange between two neighbors (see [37] for more details on the behavior of RK in the gossip setting). We name the new method the inexact randomized block gossip algorithm (iRBG).

In Figure 8 we present the performance in terms of communications of the proposed iRBG algorithm for solving the AC problem in four popular graph topologies from the area of wireless sensor networks. These are the cycle graph, the random geometric graph (RGG), the 2D grid network, and the line graph.

We highlight again that computing the number of communications required in the exact RBK (randomized block gossip algorithm from [33]) is not possible. Hence, in this setting, inexactness is not only beneficial in terms of time but also because it allows us to compute important quantities of interest like the number of communications.

7. Conclusions. In this work we propose and analyze inexact variants of several stochastic algorithms for solving quadratic optimization problems and linear systems. We provide the linear convergence rate under several assumptions on the inexactness error. The proposed methods require more iterations than their exact variants to achieve the same accuracy. However, as we show through our numerical evaluations, the inexact algorithms require significantly less time to converge.

With the continuously increasing size of datasets, we believe that inexactness should be a tool that practitioners could use even in the case of stochastic methods that have much cheaper-to-compute iteration complexity than their deterministic variants.

Recently, accelerated and parallel stochastic optimization methods [35, 54, 63] have been proposed for solving linear systems (not necessarily consistent). We spec-

[22] Recall that in our experiments we run the method until the relative error $\|x_k - x_*\|_{\mathbf{B}}^2/\|x_0 - x_*\|_{\mathbf{B}}^2 \overset{\mathbf{B}=\mathbf{I},x_0=c}{=} \|x_k - x_*\|^2/\|c - x_*\|^2$ is below 10^{-5}.

ulate that the addition of inexactness to these update rules will lead to methods that are faster in practice. We also believe that our approach and complexity results can be extended to the more general case of minimization of convex and nonconvex functions in the stochastic setting. Finally, as we have already mentioned in section 6.6, the sketch and project algorithms have been successfully used for solving the AC problem [33, 25]. We believe that following our approach more efficient randomized gossip algorithms that use inexactness in their update rule could be proposed.

Appendix A. Technical preliminaries.

LEMMA A.1 (Lemma 4.2 [54]: quadratic bounds). *For all $x \in \mathbb{R}^n$ and $x_* \in \mathcal{L}$ the following hold: $\lambda_{\min}^+ f(x) \leq \frac{1}{2}\|\nabla f(x)\|_{\mathbf{B}}^2 \leq \lambda_{\max} f(x)$ and $f(x) \leq \frac{\lambda_{\max}}{2}\|x - x_*\|_{\mathbf{B}}^2$. Furthermore, if exactness is satisfied and we let $x_* = \Pi_{\mathcal{L},\mathbf{B}}(x_0)$, then we have*

$$\frac{\lambda_{\min}^+}{2}\|x - x_*\|_{\mathbf{B}}^2 \leq f(x). \tag{A.1}$$

LEMMA A.2 (see [54]). *Let $x_* \in \mathcal{L}$ and $\{x_k\}_{k\geq 0}$ be the random iterates produced by the exact basic method (Algorithm 2.1 with $\epsilon_k = 0$) with an arbitrary stepsize $\omega \in \mathbb{R}$. Then*

$$\begin{aligned}\|x_{k+1} - x_*\|_{\mathbf{B}}^2 &= \|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k - x_*)\|_{\mathbf{B}}^2 \\ &= \|x_k - x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f_{\mathbf{S}_k}(x).\end{aligned} \tag{A.2}$$

By taking the expectation condition on x_k (that is, the expectation is with respect to $\mathbf{S}_k$) and assuming $\omega \in (0,2)$ we can further obtain

$$\begin{aligned}\mathbb{E}\left[\|x_{k+1} - x_*\|_{\mathbf{B}}^2 \mid x_k\right] &= \|x_k - x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f(x_k) \\ &\overset{(A.1)}{\leq} \left[1 - \omega(2-\omega)\lambda_{\min}^+\right]\|x_k - x_*\|_{\mathbf{B}}^2.\end{aligned} \tag{A.3}$$

Remark A.3. Let x and y be random vectors, and let σ be a positive constant. If we assume $\mathbb{E}[\|x\|_{\mathbf{B}}^2 \mid y] \leq \sigma^2$, then by using the variance inequality (check Table 3) we obtain $\mathbb{E}[\|x\|_{\mathbf{B}} \mid y] \leq \sigma$. In our setting if we assume $\mathbb{E}[\|\epsilon_k\|_{\mathbf{B}}^2 \mid x_k, \mathbf{S}_k] \leq \sigma_k^2$, where ϵ_k is the inexactness error and x_k is the current iterate, then by the variance inequality we have $\mathbb{E}[\|\epsilon_k\|_{\mathbf{B}} \mid x_k, \mathbf{S}_k] \leq \sigma_k$.

Appendix B. Proofs of main results. In our convergence analysis we use several known inequalities. Consult Table 3 for the abbreviations and the relevant formulas.

A key step in the proofs of the theorems is to use the tower property of the expectation. We use it in the form

$$\mathbb{E}[\mathbb{E}[\mathbb{E}[X \mid x_k, \mathbf{S}_k] \mid x_k]] = \mathbb{E}[X], \tag{B.1}$$

where X is some random variable. In all proofs we perform the three expectations in order, from the innermost to the outermost. Similar to the main part of the paper we use $\rho = 1 - \omega(2-\omega)\lambda_{\min}^+$.

B.1. Proof of Theorem 3.6.

Proof. First we decompose

$$\begin{aligned}\|x_{k+1} - x_*\|_{\mathbf{B}}^2 &= \|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k - x_*) + \epsilon_k\|_{\mathbf{B}}^2 \\ &= \|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k - x_*)\|_{\mathbf{B}}^2 + \|\epsilon_k\|_{\mathbf{B}}^2 \\ &\quad + 2\left\langle(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k - x_*), \epsilon_k\right\rangle_{\mathbf{B}}.\end{aligned} \tag{B.2}$$

TABLE 3
Popular inequalities with abbreviations and formulas.

Useful inequalities			
Inequalities (full names)	Abbrev.	Formula	Assumptions
Jensen inequality	*Jensen*	$f(\mathbb{E}[x]) \leq \mathbb{E}[f(x)]$	f is convex
Conditional Jensen inequality	*C. Jensen*	$f(\mathbb{E}[x \mid s]) \leq \mathbb{E}[f(x) \mid s]$	f is convex
Cauchy–Schwarz ($\mathbf{B}$-norm)	*C.S.*	$\lvert\langle a,b\rangle_{\mathbf{B}}\rvert \leq \|a\|_{\mathbf{B}}\|b\|_{\mathbf{B}}$	$a,b \in \mathbb{R}^n$
Variance inequality	*V.I.*	$(\mathbb{E}[X])^2 \leq \mathbb{E}[X^2]$	X random variable

Applying the innermost expectation of (B.1) to (B.2), we get

$$\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] = \underbrace{\mathbb{E}[\|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k]}_{T1} + \underbrace{\mathbb{E}[\|\epsilon_k\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k]}_{T2}$$
$$+2\underbrace{\mathbb{E}[\left\langle(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*),\epsilon_k\right\rangle_{\mathbf{B}} \mid x_k,\mathbf{S}_k]}_{T3}. \tag{B.3}$$

We now analyze the three expressions $T1$, $T2$, and $T3$ separately. We have

$$\begin{aligned} T1 &= \|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}}^2 \\ &\overset{(A.2)}{=} \|x_k-x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f_{\mathbf{S}_k}(x_k); \end{aligned} \tag{B.4}$$

the bound on $T2$ follows by assumption,

$$T2 \leq \sigma_k^2, \tag{B.5}$$

and

$$\begin{aligned} T3 &= \left\langle(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*),\mathbb{E}[\epsilon_k \mid x_k,\mathbf{S}_k]\right\rangle_{\mathbf{B}} \\ &\overset{\text{C.S.}}{\leq} \|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}}\|\mathbb{E}[\epsilon_k \mid x_k,\mathbf{S}_k]\|_{\mathbf{B}} \\ &\overset{\text{C.Jensen}}{\leq} \|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}}\mathbb{E}[\|\epsilon_k\|_{\mathbf{B}} \mid x_k,\mathbf{S}_k] \\ &\overset{\text{Remark A.3 and (3.2)}}{\leq} \|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}}\sigma_k. \end{aligned} \tag{B.6}$$

By substituting the bounds (B.5), (B.4), and (B.6) into (B.3) we obtain

$$\begin{aligned} \mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] &\leq \|x_k-x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f_{\mathbf{S}_k}(x_k) + \sigma_k^2 \\ &\quad +2\|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}}\sigma_k. \end{aligned} \tag{B.7}$$

We now take the middle expectation (see (B.1)) and apply it to inequality (B.7):

$$\begin{aligned} \mathbb{E}[\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] \mid x_k] &\leq \|x_k-x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f(x_k) + \sigma_k^2 \\ &\quad +2\mathbb{E}[\|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}} \mid x_k]\sigma_k. \end{aligned} \tag{B.8}$$

Now let us find a bound on the quantity $\mathbb{E}\left[\|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}} \mid x_k\right]$. Note that from (A.3) and (A.2) we have that $\mathbb{E}\left[\|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}}^2 \mid x_k\right] \leq \rho\|x_k-x_*\|_{\mathbf{B}}^2$. By using Remark A.3 in the last inequality we obtain

$$\mathbb{E}\left[\|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}} \mid x_k\right] = \sqrt{\rho}\|x_k-x_*\|_{\mathbf{B}}. \tag{B.9}$$

By substituting (B.9) into (B.8) we obtain

$$\begin{aligned}\mathbb{E}[\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] \mid x_k] &\leq \|x_k-x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f(x_k)+\sigma_k^2 \\ &\quad +2\sigma_k\sqrt{\rho}\|x_k-x_*\|_{\mathbf{B}} \\ &\overset{(A.3)}{\leq} \rho\|x_k-x_*\|_{\mathbf{B}}^2+\sigma_k^2+2\sigma_k\sqrt{\rho}\|x_k-x_*\|_{\mathbf{B}}. \end{aligned} \tag{B.10}$$

We take the final expectation (outermost expectation in the tower rule (B.1)) on the above expression to find

$$\begin{aligned}\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2] &= \mathbb{E}[\mathbb{E}[\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] \mid x_k]] \\ &\leq \rho\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}^2]+\sigma_k^2+2\sigma_k\sqrt{\rho}\ \mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}] \\ &\overset{V.I}{\leq} \rho\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}^2]+\sigma_k^2+2\sigma_k\sqrt{\rho}\sqrt{\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}^2]}. \end{aligned} \tag{B.11}$$

Using $r_k=\mathbb{E}\left[\|x_k-x_*\|_{\mathbf{B}}^2\right]$ we find that (B.11) takes the form

$$r_{k+1}\leq \rho r_k+\sigma_k^2+2\sigma_k\sqrt{\rho}\sqrt{r_k}=\left(\sqrt{\rho r_k}+\sigma_k\right)^2.$$

If we further substitute $p_k=\sqrt{r_k}$ and $\ell=\sqrt{\rho}$, the recurrence simplifies to $p_{k+1}\leq \ell p_k+\sigma_k$. By unrolling the final inequality, we obtain

$$p_k\leq \ell^k r_0+(\ell^0\sigma_{k-1}+\ell\sigma_{k-2}+\cdots+\ell^{k-1}\sigma_0)=\ell^k p_0+\sum_{i=0}^{k-1}\ell^{k-1-i}\sigma_i.$$

Hence,

$$\sqrt{\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}^2]}\leq \rho^{k/2}\|x_0-x_*\|_{\mathbf{B}}+\sum_{i=0}^{k-1}\rho^{\frac{k-1-i}{2}}\sigma_i.$$

The result is obtained by using V.I (see Table 3) in the last expression. ∎

B.2. Proof of Corollary 3.7. By denoting $r_k=\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}]$ in (3.6) we obtain

$$r_k\leq \rho^{k/2}r_0+\rho^{1/2}\sigma\sum_{i=0}^{k-1}\rho^{k-1-i}=\rho^{k/2}r_0+\rho^{1/2}\sigma\sum_{i=0}^{k-1}\rho^i=\rho^{k/2}r_0+\rho^{1/2}\sigma\frac{1-\rho^k}{1-\rho}.$$

Since $1-\rho^k\leq 1$, the result is obtained.

B.3. Proof of Theorem 3.8. In order to prove Theorem 3.8 we need to follow steps similar to those in the proof of Theorem 3.6. The main differences between the two proofs appear at the points where we need to upper bound the norm of the inexactness error ($\|\epsilon_k\|^2$). In particular, instead of using the general sequence $\sigma_k^2\in\mathbb{R}$, we utilize the bound $q^2\|x_k-x_*\|_{\mathbf{B}}^2$ from Assumption 3.3. Thus, it is sufficient to focus on the parts of the proof in which this bound is used.

Similar to the proof of Theorem 3.6 we first decompose to obtain (B.3). There, the expression $T1$ can be upper bounded from (B.4), but now using Assumption 3.3 the expressions $T2$ and $T3$ can be upper bounded as follows:

$$T2=\mathbb{E}\left[\|\epsilon_k\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k\right]\leq q^2\|x_k-x_*\|_{\mathbf{B}}^2. \tag{B.12}$$

$$T3 = \mathbb{E}\left[\left\langle(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*),\epsilon_k\right\rangle_{\mathbf{B}} \mid x_k,\mathbf{S}_k\right]$$

$$\overset{\text{Remark A.3 and (B.6)}}{\leq} \|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}} q\|x_k-x_*\|_{\mathbf{B}}. \tag{B.13}$$

As a result, by substituting the bounds (B.4), (B.12), and (B.13) into (B.3), we obtain

$$\begin{aligned}\mathbb{E}\left[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k\right] &\overset{(B.3)}{\leq} \|x_k-x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f_{\mathbf{S}_k}(x_k) + q^2\|x_k-x_*\|_{\mathbf{B}}^2 \\ &\quad +2\|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}} q\|x_k-x_*\|_{\mathbf{B}}.\end{aligned} \tag{B.14}$$

By following the same steps of the proof of Theorem 3.6, (B.10) takes the form

$$\begin{aligned}\mathbb{E}\left[\mathbb{E}\left[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k\right] \mid x_k\right] &\leq \rho\|x_k-x_*\|_{\mathbf{B}}^2 + q^2\|x_k-x_*\|_{\mathbf{B}}^2 \\ &\quad +2q\|x_k-x_*\|_{\mathbf{B}}\sqrt{\rho}\|x_k-x_*\|_{\mathbf{B}} \\ &= \left(\rho+2q\sqrt{\rho}+q^2\right)\|x_k-x_*\|_{\mathbf{B}}^2 \\ &= \left(\sqrt{\rho}+q\right)^2\|x_k-x_*\|_{\mathbf{B}}^2.\end{aligned} \tag{B.15}$$

We take the final expectation (outermost expectation in the tower rule (B.1)) on the above expression to find

$$\begin{aligned}\mathbb{E}\left[\|x_{k+1}-x_*\|_{\mathbf{B}}^2\right] &= \mathbb{E}\left[\mathbb{E}\left[\mathbb{E}\left[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k\right] \mid x_k\right]\right] \\ &\leq \left(\sqrt{\rho}+q\right)^2\mathbb{E}\left[\|x_k-x_*\|_{\mathbf{B}}^2\right].\end{aligned} \tag{B.16}$$

The final result follows by unrolling the recurrence.

B.4. Proof of Theorem 3.9.

Proof. Similar to the previous two proofs, by decomposing the update rule and using the innermost expectation of (B.1) we obtain (B.3). An upper bound of expression $T1$ is again given by inequality (B.4). For the expression $T2$, depending on the assumption that we have on the norm of the inexactness error, different upper bounds can be used. In particular, the following hold:

(i) If Assumption 3.1 holds, then $T2 = \mathbb{E}[\|\epsilon_k\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] \leq \sigma_k^2$.
(ii) If Assumption 3.3 holds, then $T2 = \mathbb{E}[\|\epsilon_k\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] \leq \sigma_k^2 = q^2\|x_k-x_*\|_{\mathbf{B}}^2$.
(iii) If Assumption 3.4 holds, then $T2 = \mathbb{E}[\|\epsilon_k\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] \leq \sigma_k^2 = 2q^2 f_{S_k}(x_k)$.

The main difference from the previous proofs is that due to Assumption 3.5 and the tower property (B.1), expression $T3$ will eventually be equal to zero. More specifically, we have that

$$\begin{aligned}&\mathbb{E}\left[\mathbb{E}\left[\mathbb{E}\left[\left\langle(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*),\epsilon_k\right\rangle_{\mathbf{B}} \mid x_k,\mathbf{S}_k\right] \mid x_k\right]\right] \\ &\qquad = \mathbb{E}\left[\left\langle(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*),\epsilon_k\right\rangle_{\mathbf{B}}\right] = T3 = 0.\end{aligned}$$

Thus, in this case (B.7) takes the form

$$\mathbb{E}\left[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k\right] \leq \|x_k-x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f_{\mathbf{S}_k}(x_k) + \sigma_k^2. \tag{B.17}$$

Using the above expression, depending on the assumptions, we obtain the following results:

(i) By taking the middle expectation (see (B.1)) and applying it to the above inequality,

$$\begin{aligned}\mathbb{E}\left[\mathbb{E}\left[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k\right] \mid x_k\right] &\leq \|x_k-x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f(x_k) \\ &\quad +\mathbb{E}\left[\sigma_k^2 \mid x_k\right] \\ &\overset{(A.3)}{\leq} \rho\|x_k-x_*\|_{\mathbf{B}}^2 + \mathbb{E}\left[\sigma_k^2 \mid x_k\right].\end{aligned} \tag{B.18}$$

We take the final expectation (outermost expectation in the tower rule (B.1)) on the above expression to find

$$\begin{aligned}\mathbb{E}\left[\|x_{k+1}-x_*\|_{\mathbf{B}}^2\right] &= \mathbb{E}\left[\mathbb{E}\left[\mathbb{E}\left[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k, \mathbf{S}_k\right] \mid x_k\right]\right] \\ &\leq \rho\mathbb{E}\left[\|x_k-x_*\|_{\mathbf{B}}^2\right] + \mathbb{E}\left[\mathbb{E}\left[\sigma_k^2 \mid x_k\right]\right] \\ &= \rho\mathbb{E}\left[\|x_k-x_*\|_{\mathbf{B}}^2\right] + \mathbb{E}\left[\sigma_k^2\right] \\ &= \rho\mathbb{E}\left[\|x_k-x_*\|_{\mathbf{B}}^2\right] + \bar{\sigma}_k^2. \qquad \text{(B.19)}\end{aligned}$$

Using $r_k = \mathbb{E}\left[\|x_k-x_*\|_{\mathbf{B}}^2\right]$ the last inequality takes the form $r_{k+1} \leq \rho r_k + \bar{\sigma}_k^2$. By unrolling the last expression, $r_k \leq \rho^k r_0 + (\rho^0\bar{\sigma}_{k-1}^2 + \rho\bar{\sigma}_{k-2}^2 + \cdots + \rho^{k-1}\bar{\sigma}_0^2) = \rho^k r_0 + \sum_{i=0}^{k-1}\rho^{k-1-i}\bar{\sigma}_i^2$. Hence,

$$\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}^2] \leq \rho^k\|x_0-x_*\|_{\mathbf{B}}^2 + \sum_{i=0}^{k-1}\rho^{k-1-i}\bar{\sigma}_i^2.$$

(ii) For case (ii) inequality (B.17) takes the form

$$\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k, \mathbf{S}_k] \leq \|x_k-x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f_{\mathbf{S}_k}(x_k) + q^2\|x_k-x_*\|_{\mathbf{B}}^2,$$

and by taking the middle expectation (see (B.1)) we obtain

$$\begin{aligned}\mathbb{E}[\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k, \mathbf{S}_k] \mid x_k] &\leq \|x_k-x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f(x_k) \\ &\quad + q^2\|x_k-x_*\|_{\mathbf{B}}^2 \\ &\overset{\text{(A.3)}}{\leq} \rho\|x_k-x_*\|_{\mathbf{B}}^2 + q^2\|x_k-x_*\|_{\mathbf{B}}^2 \\ &= (\rho+q^2)\|x_k-x_*\|_{\mathbf{B}}^2. \qquad \text{(B.20)}\end{aligned}$$

By taking the final expectation of the tower rule (B.1) and applying it to the above inequality, we get

$$\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2] \leq (\rho+q^2)\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}^2]. \qquad \text{(B.21)}$$

The result is obtained by unrolling the last recursion.

(iii) Inequality (B.17) takes the form

$$\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k, \mathbf{S}_k] \leq \|x_k-x_*\|_{\mathbf{B}}^2 - 2(\omega(2-\omega)-q^2)f_{\mathbf{S}_k}(x_k),$$

and by taking the middle expectation (see (B.1)), we obtain

$$\begin{aligned}\mathbb{E}[\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k, \mathbf{S}_k] \mid x_k] &\leq \|x_k-x_*\|_{\mathbf{B}}^2 - 2(\omega(2-\omega)-q^2)f(x_k) \\ &\overset{\text{(A.1)}}{\leq} \|x_k-x_*\|_{\mathbf{B}}^2 \\ &\quad - (\omega(2-\omega)-q^2)\lambda_{\min}^+\|x_k-x_*\|_{\mathbf{B}}^2 \\ &= (1-(\omega(2-\omega)-q^2)\lambda_{\min}^+)\|x_k-x_*\|_{\mathbf{B}}^2. \qquad \text{(B.22)}\end{aligned}$$

By taking the final expectation of the tower rule (B.1) to the above inequality, we get

$$\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2] \leq \left(1-(\omega(2-\omega)-q^2)\,\lambda_{\min}^+\right)\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}^2], \qquad \text{(B.23)}$$

and the result is obtained by unrolling the last recursion. ∎

Appendix C. Notation glossary.

TABLE 4
Frequently used notation.

The basics	
$\mathbf{A}, b$	$m \times n$ matrix and $m \times 1$ vector defining the system $\mathbf{A}x = b$
$\mathcal{L}$	$\{x \,:\, \mathbf{A}x = b\}$ (solution set of the linear system)
$\mathbf{B}$	$n \times n$ symmetric positive definite matrix
$\langle x, y\rangle_{\mathbf{B}}$	$x^\top \mathbf{B} y$ ($\mathbf{B}$-inner product)
$\|x\|_{\mathbf{B}}$	$\sqrt{\langle x, x\rangle_{\mathbf{B}}}$ ($\mathbf{B}$-norm)
$\mathbf{U}^\dagger$	Moore–Penrose pseudoinverse of matrix $\mathbf{U}$
$\mathbf{U}^{\dagger_{\mathbf{B}}}$	$\mathbf{B}^{-1}\mathbf{U}^\top(\mathbf{U}\mathbf{B}^{-1}\mathbf{U}^\top)^\dagger$ ($\mathbf{B}$-pseudoinverse of matrix $\mathbf{U}$)
$\mathbf{S}$	a random real matrix with m rows
$\mathcal{D}$	distribution from which matrix $\mathbf{S}$ is drawn ($\mathbf{S} \sim \mathcal{D}$)
$\mathbf{H}$	$\mathbf{S}(\mathbf{S}^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S})^\dagger \mathbf{S}^\top$
$\mathbf{Z}$	$\mathbf{A}^\top \mathbf{H}\mathbf{A}$
Range $(\mathbf{U})$	range space of matrix $\mathbf{U}$
Null $(\mathbf{U})$	null space of matrix $\mathbf{U}$
$\mathbb{P}(\cdot)$	probability of an event
$\mathbb{E}[\cdot]$	expectation
Projections	
$\Pi_{\mathcal{L},\mathbf{B}}(x)$	projection of x onto $\mathcal{L}$ in the $\mathbf{B}$-norm
$\mathbf{B}^{-1}\mathbf{Z}$	projection matrix, in the $\mathbf{B}$-norm, onto Range $(\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S})$
Optimization	
$\mathcal{X}$	set of minimizers of f
x_*	a point in $\mathcal{L}$
$f_{\mathbf{S}}$, $\nabla f_{\mathbf{S}}$, $\nabla^2 f_{\mathbf{S}}$	stochastic function, its gradient, and Hessian
$\mathcal{L}_{\mathbf{S}}$	$\{x \,:\, \mathbf{S}^\top \mathbf{A}x = \mathbf{S}^\top b\}$ (set of minimizers of $f_{\mathbf{S}}$)
f	$\mathbb{E}[f_{\mathbf{S}}]$
∇f	gradient of f with respect to the $\mathbf{B}$-inner product
$\nabla^2 f$	$\mathbf{B}^{-1}\mathbb{E}[\mathbf{Z}]$ (Hessian of f in the $\mathbf{B}$-inner product)
Eigenvalues	
$\mathbf{W}$	$\mathbf{B}^{-1/2}\mathbb{E}[\mathbf{Z}]\mathbf{B}^{-1/2}$ (psd matrix with the same spectrum as $\nabla^2 f$)
$\lambda_1, \ldots, \lambda_n$	eigenvalues of $\mathbf{W}$
$\lambda_{\max}, \lambda_{\min}^+$	largest and smallest nonzero eigenvalues of $\mathbf{W}$
Algorithms	
$\mathbf{M}_k$	$\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k$
d_k	$\mathbf{S}_k^\top (b - \mathbf{A}x_k)$
$\mathcal{Q}_k$	$\{\lambda \in \mathbb{R}^q : \mathbf{M}_k \lambda = d_k\}$
λ_k^*	$\arg\min_{\lambda \in \mathcal{Q}_k} \|\lambda\|$
$\lambda_k^{\widetilde{*}}$	approximation of λ_k^*
ω	relaxation parameter / stepsize
ϵ_k	inexactness error
q	inexactness parameter
ρ	$1 - \omega(2-\omega)\lambda_{\min}^+$

Acknowledgments. The authors would like to acknowledge Robert Mansel Gower, Georgios Loizou, Aritra Dutta, and Rachael Tappenden for useful discussions.

REFERENCES

[1] Z. Allen-Zhu, Z. Qu, P. Richtárik, and Y. Yuan, *Even faster accelerated coordinate descent using non-uniform sampling*, in ICML, New York, NY, 2016, pp. 1110–1119.
[2] F. Bénézit, A. Dimakis, P. Thiran, and M. Vetterli, *Order-optimal consensus through randomized path averaging*, IEEE Trans. Inform. Theory, 56 (2010), pp. 5150–5167.
[3] A. Berahas, R. Bollapragada, and J. Nocedal, *An Investigation of Newton-Sketch and Subsampled Newton Methods*, preprint, https://arxiv.org/abs/1705.06211, 2017.

[4] P. Birken, *Termination criteria for inexact fixed-point schemes*, Numer. Linear Algebra Appl., 22 (2015), pp. 702–716.
[5] K. Biswas, V. Muthukkumarasamy, E. Sithirasenan, and M. Usman, *An energy efficient clique based clustering and routing mechanism in wireless sensor networks*, in Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC), IEEE, Washington, DC, 2013, pp. 171–176.
[6] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996, https://doi.org/10.1137/1.9781611971484.
[7] R. Bollapragada, R. Byrd, and J. Nocedal, *Exact and Inexact Subsampled Newton Methods for Optimization*, preprint, https://arxiv.org/abs/1609.08502, 2016.
[8] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, *Randomized gossip algorithms*, IEEE Trans. Inform. Theory, 14 (2006), pp. 2508–2530.
[9] C. Byrne, *Applied Iterative Methods*, A K Peters, Wellesley, MA, 2008.
[10] A. Cassioli, D. Di Lorenzo, and M. Sciandrone, *On the convergence of inexact block coordinate descent methods for constrained optimization*, European J. Oper. Res., 231 (2013), pp. 274–281.
[11] A. Chambolle, M. Ehrhardt, P. Richtárik, and C. Schönlieb, *Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications*, SIAM J. Optim., 28 (2018), pp. 2783–2808.
[12] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, ACM Trans. Intelligent Syst. Tech. 2 (2011), 27.
[13] D. Csiba and P. Richtárik, *Global Convergence of Arbitrary-Block Gradient Methods for Generalized Polyak-Łojasiewicz Functions*, preprint, https://arxiv.org/abs/1709.03014, 2017.
[14] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408, https://doi.org/10.1137/0719025.
[15] O. Devolder, F. Glineur, and Y. Nesterov, *First-order methods of smooth convex optimization with inexact oracle*, Math. Program., 146 (2014), pp. 37–75.
[16] A. Dimakis, S. Kar, J. Moura, M. Rabbat, and A. Scaglione, *Gossip algorithms for distributed signal processing*, Proc. IEEE, 98 (2010), pp. 1847–1864.
[17] P. Dvurechensky, A. Gasnikov, and A. Tiurin, *Randomized Similar Triangles Method: A Unifying Framework for Accelerated Randomized Optimization Methods (Coordinate Descent, Directional Search, Derivative-Free Method)*, preprint, https://arxiv.org/abs/1707.08486, 2017.
[18] Y. Eldar and D. Needell, *Acceleration of randomized Kaczmarz method via the Johnson–Lindenstrauss lemma*, Numer. Algorithms, 58 (2011), pp. 163–177.
[19] O. Fercoq and P. Richtárik, *Accelerated, parallel, and proximal coordinate descent*, SIAM J. Optim., 25 (2015), pp. 1997–2023, https://doi.org/10.1137/130949993.
[20] K. Fountoulakis and R. Tappenden, *A flexible coordinate descent method*, Comput. Optim. Appl., 70 (2018), pp. 351–394.
[21] M. Friedlander and M. Schmidt, *Hybrid deterministic-stochastic methods for data fitting*, SIAM J. Sci. Comput., 34 (2012), pp. A1380–A1405, https://doi.org/10.1137/110830629.
[22] G. Golub and C. Van Loan, *Matrix Computations*, 4th ed., Johns Hopkins Stud. Math. Sci. 3, Johns Hopkins University Press, Baltimore, MD, 2013.
[23] R. M. Gower and P. Richtárik, *Randomized iterative methods for linear systems*, SIAM. J. Matrix Anal. Appl., 36 (2015), pp. 1660–1690, https://doi.org/10.1137/15M1025487.
[24] R. M. Gower and P. Richtárik, *Stochastic Dual Ascent for Solving Linear Systems*, preprint, https://arxiv.org/abs/1512.06890, 2015.
[25] F. Hanzely, J. Konečný, N. Loizou, P. Richtárik, and D. Grishchenko, *Privacy Preserving Randomized Gossip Algorithms*, preprint, https://arxiv.org/abs/1706.07636, 2017.
[26] F. Hanzely, J. Konečnỳ, N. Loizou, P. Richtárik, and D. Grishchenko, *A privacy preserving randomized gossip algorithm via controlled noise insertion*, in NeurIPS Privacy Preserving Machine Learning Workshop, Montreal, Canada, 2018.
[27] B. Hu, P. Seiler, and L. Lessard, *Analysis of Approximate Stochastic Gradient Using Quadratic Constraints and Sequential Semidefinite Programs*, preprint, https://arxiv.org/abs/1711.00987v1, 2017.
[28] S. Kaczmarz, *Angenäherte Auflösung von Systemen linearer Gleichungen*, Bulletin International de l'Academie Polonaise des Sciences et des Lettres, 35 (1937), pp. 355–357.
[29] Y. Lee and A. Sidford, *Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems*, in Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE, Washington, DC, 2013, pp. 147–156.

[30] D. Leventhal and A. Lewis, *Randomized methods for linear constraints: Convergence rates and conditioning*, Math. Oper. Res., 35 (2010), pp. 641–654.
[31] Y. Liu, B. Li, B. Anderson, and G. Shi, *Clique Gossiping*, preprint, https://arxiv.org/abs/1706.02540, 2017.
[32] N. Loizou, M. Rabbat, and P. Richtárik, *Provably accelerated randomized gossip algorithms*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Washington, DC, 2019, pp. 7505–7509.
[33] N. Loizou and P. Richtárik, *A new perspective on randomized gossip algorithms*, in Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP), IEEE, Washington, DC, 2016, pp. 440–444.
[34] N. Loizou and P. Richtárik, *Linearly convergent stochastic heavy ball method for minimizing generalization error*, in NIPS Workshop on Optimization for Machine Learning, 2017; preprint, https://arxiv.org/abs/1710.10737, 2017.
[35] N. Loizou and P. Richtárik, *Momentum and Stochastic Momentum for Stochastic Gradient, Newton, Proximal Point and Subspace Descent Methods*, preprint, https://arxiv.org/abs/1712.09677, 2017.
[36] N. Loizou and P. Richtárik, *Accelerated gossip via stochastic heavy ball method*, in Proceedings of the 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton, IL), IEEE, Washington, DC, 2018, pp. 927–934.
[37] N. Loizou and P. Richtárik, *Revisiting Randomized Gossip Algorithms: General Framework, Convergence Rates and Novel Block and Accelerated Protocols*, preprint, https://arxiv.org/abs/1905.08645, 2019.
[38] A. Ma, D. Needell, and A. Ramdas, *Convergence properties of the randomized extended Gauss–Seidel and Kaczmarz methods*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 1590–1604, https://doi.org/10.1137/15M1014425.
[39] I. Necoara and V. Nedelcu, *Rate analysis of inexact dual first-order methods application to dual decomposition*, IEEE Trans. Automat. Control, 59 (2014), pp. 1232–1243.
[40] D. Needell, *Randomized Kaczmarz solver for noisy linear systems*, BIT, 50 (2010), pp. 395–403.
[41] D. Needell and E. Rebrova, *On Block Gaussian Sketching for Iterative Projections*, preprint, https://arxiv.org/abs/1905.08894v1, 2019.
[42] D. Needell and J. Tropp, *Paved with good intentions: Analysis of a randomized block Kaczmarz method*, Linear Algebra Appl., 441 (2014), pp. 199–221.
[43] D. Needell, R. Zhao, and A. Zouzias, *Randomized block Kaczmarz method with projection for solving least squares*, Linear Algebra Appl., 484 (2015), pp. 322–343.
[44] Y. Nesterov, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM J. Optim., 22 (2012), pp. 341–362, https://doi.org/10.1137/100802001.
[45] J. Nutini, B. Sepehry, I. Laradji, M. Schmidt, H. Koepke, and A. Virani, *Convergence rates for greedy Kaczmarz algorithms, and faster randomized Kaczmarz rules using the orthogonality graph*, in Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, AUAI Press, Arlington, VA, 2016, pp. 547–556.
[46] C. Popa, *Least-squares solution of overdetermined inconsistent linear systems using Kaczmarz's relaxation*, Internat. J. Comput. Math., 55 (1995), pp. 79–89.
[47] C. Popa, *Convergence Rates for Kaczmarz-Type Algorithms*, preprint, https://arxiv.org/abs/1701.08002, 2017.
[48] Z. Qu and P. Richtárik, *Coordinate descent with arbitrary sampling* I: *Algorithms and complexity*, Optim. Methods Softw., 31 (2016), pp. 829–857.
[49] Z. Qu and P. Richtárik, *Coordinate descent with arbitrary sampling* II: *Expected separable overapproximation*, Optim. Methods Softw., 31 (2016), pp. 858–884.
[50] Z. Qu, P. Richtárik, M. Takáč, and O. Fercoq, *SDNA: Stochastic dual Newton ascent for empirical risk minimization*, in ICML, New York, NY, 2016, pp. 1823–1832.
[51] Z. Qu, P. Richtárik, and T. Zhang, *Quartz: Randomized dual coordinate ascent with arbitrary sampling*, in Advances in Neural Information Processing Systems, NeurIPS, San Diego, CA, 2015, pp. 865–873.
[52] P. Richtárik and M. Takáč, *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*, Math. Program., 144 (2014), pp. 1–38.
[53] P. Richtárik and M. Takáč, *Parallel coordinate descent methods for big data optimization*, Math. Program., 156 (2016), pp. 433–484.
[54] P. Richtárik and M. Takáč, *Stochastic Reformulations of Linear Systems: Algorithms and Convergence Theory*, preprint, https://arxiv.org/abs/1706.01108, 2017.

[55] A. N. Sahu, A. Dutta, A. Tiwari, and P. Richtárik, *On the Convergence Analysis of Asynchronous SGD for Solving Consistent Linear Systems*, preprint, https://arxiv.org/abs/2004.02163, 2020.

[56] S. Salzo and S. Villa, *Inexact and accelerated proximal point algorithms*, J. Convex Anal., 19 (2012), pp. 1167–1192.

[57] M. Schmidt, D. Kim, and S. Sra, *Projected Newton-type methods in machine learning*, in Optimization for Machine Learning, MIT Press, Cambridge, MA, 2011, pp. 305–330.

[58] M. Schmidt, N. Roux, and F. Bach, *Convergence rates of inexact proximal-gradient methods for convex optimization*, in Advances in Neural Information Processing Systems, NeurIPS, San Diego, CA, 2011, pp. 1458–1466.

[59] F. Schöpfer and D. Lorenz, *Linear Convergence of the Randomized Sparse Kaczmarz Method*, preprint, https://arxiv.org/abs/1610.02889, 2016.

[60] A. M.-C. So and Z. Zhou, *Non-asymptotic convergence analysis of inexact gradient methods for machine learning without strong convexity*, Optim. Methods Softw., 32 (2017), pp. 963–992.

[61] M. Solodov and B. Svaiter, *A unified framework for some inexact proximal point algorithms*, Numer. Funct. Anal. Optim., 22 (2001), pp. 1013–1035.

[62] R. Tappenden, P. Richtárik, and J. Gondzio, *Inexact coordinate descent: Complexity and preconditioning*, J. Optim. Theory Appl., 170 (2016), pp. 144–176.

[63] S. Tu, S. Venkataraman, A. Wilson, A. Gittens, M. Jordan, and B. Recht, *Breaking locality accelerates block Gauss-Seidel*, in ICML, Sydney, Australia, 2017, pp. 3482–3491.

[64] S. Wright and J. Nocedal, *Numerical Optimization*, Springer Ser. Oper. Res., Springer, New York, 1999.

[65] P. Xu, F. Roosta-Khorasani, and M. Mahoney, *Newton-Type Methods for Non-Convex Optimization under Inexact Hessian Information*, preprint, https://arxiv.org/abs/1708.07164, 2017.

[66] P. Xu, J. Yang, F. Roosta-Khorasani, C. Ré, and M. Mahoney, *Sub-sampled Newton methods with non-uniform sampling*, in Advances in Neural Information Processing Systems, NeurIPS, San Diego, CA, 2016, pp. 3000–3008.

[67] Z. Yao, P. Xu, F. Roosta-Khorasani, and M. W. Mahoney, *Inexact Non-Convex Newton-Type Methods*, preprint, https://arxiv.org/abs/1802.06925, 2018.

[68] Y. Zeng, R. C. Hendriks, and R. Heusdens, *Clique-based distributed beamforming for speech enhancement in wireless sensor networks*, in 21st European Signal Processing Conference (EUSIPCO 2013), IEEE, Washington, DC, 2013, pp. 1–5.

[69] A. Zouzias and N. M. Freris, *Randomized extended Kaczmarz for solving least squares*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 773–793, https://doi.org/10.1137/120889897.