Convergence Analysis of Inexact Randomized Iterative Methods

Nicolas Loizou * Peter Richtárik †

May 28, 2019

Abstract

In this paper we present a convergence rate analysis of inexact variants of several randomized iterative methods. 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 sub-problem needs to be solved exactly. We relax this requirement by allowing for the sub-problem to be solved inexactly. In particular, we propose and analyze inexact 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. 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, randomized Gaussian Kaczmarz and randomized block coordinate descent, can be cast as special cases. Numerical experiments demonstrate the benefits of allowing inexactness.

Keywords Inexact methods · Iteration complexity · Linear systems · Randomized block coordinate descent · Randomized block Kaczmarz · Stochastic gradient descent · Stochastic Newton method· Quadratic optimization · Convex optimization
Mathematical Subject Classifications 68Q25 · 68W20 · 68W40 · 65Y20 · 90C15 · 90C20 · 90C25 · 15A06 · 15B52 · 65F10

1 Introduction

In the era of big data where data sets become continuously larger, randomized iterative methods become very popular and they are now playing 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 in the stochastic methods under study.

1.1 The Setting

In this paper we are interested to solve three closely related problems:

- Stochastic Quadratic Optimization Problem
- Best Approximation Problem
- 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 solving each one of them.

*University of Edinburgh
†King Abdullah University of Science and Technology (KAUST); University of Edinburgh, MIPT

Stochastic Optimization Problem: We study the stochastic quadratic optimization problem

$$\min_{x\in\mathbb{R}^n} f(x) := \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}[f_{\mathbf{S}}(x)], \tag{1}$$

first proposed in [46] for reformulating *consistent* linear systems

$$\mathbf{A}x = b. \tag{2}$$

In particular, problem (1) is defined by setting:

$$f_{\mathbf{S}}(x) := \frac{1}{2}\|\mathbf{A}x-b\|^2_{\mathbf{H}} = \frac{1}{2}(\mathbf{A}x-b)^\top\mathbf{H}(\mathbf{A}x-b), \tag{3}$$

where $\mathbf{H}$ is a random symmetric positive semi-definite matrix $\mathbf{H} := \mathbf{S}(\mathbf{S}^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S})^\dagger\mathbf{S}^\top$ that depends on three different matrices: the data matrix $\mathbf{A}\in\mathbb{R}^{m\times n}$ of the linear system (2), a random matrix $\mathbf{S}\in\mathbb{R}^{m\times q}\sim\mathcal{D}$ and on an $n\times n$ positive definite matrix $\mathbf{B}$ which defines the geometry of the space. 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) is over random matrices $\mathbf{S}$ with m rows (and arbitrary number of columns q, e.g., $q=1$) drawn from an arbitrary (user defined) distribution $\mathcal{D}$. The authors of [46] give necessary and sufficient conditions that distribution D needs to be satisfied for the set of solutions of (1) to be equal to the set of solutions of the linear system (2); a property for which the term exactness was coined in (see Section 3 for more details on exactness).

In [46], problem (1) was solved via Stochastic Gradient Descent (SGD)[1]:

$$x_{k+1} = x_k - \omega\nabla f_{\mathbf{S}_k}(x_k), \tag{4}$$

and a linear rate of convergence was proved despite the fact that f is not necessarily strongly convex, (1) is not a finite-sum problem and a fixed stepsize $\omega>0$ is used.

The stochastic optimization problem (1) has many unique characteristics mainly because it has constructed in a particular way in order to capture all the information of the linear system (2). 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 its 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) have exactly the same behavior with SGD [46]:

$$\textit{Stochastic Newton Method (SNM)}^2: \; x_{k+1} = x_k - \omega(\nabla^2 f_{\mathbf{S}_k}(x_k))^{\dagger_{\mathbf{B}}}\nabla f_{\mathbf{S}_k}(x_k), \tag{5}$$

$$\textit{Stochastic Proximal Point Method (SPPM)}^3: \; x_{k+1} = \arg\min_{x\in\mathbb{R}^n}\left\{f_{\mathbf{S}_k}(x) + \frac{1-\omega}{2\omega}\|x-x_k\|^2_{\mathbf{B}}\right\}. \tag{6}$$

In all methods $\omega>0$ is a fixed stepsize and $\mathbf{S}_k$ is sampled afresh in each iteration from distribution $\mathcal{D}$. See [46] for more insights into the reformulation (1), its properties and other equivalent reformulations (e.g., stochastic fixed point problem, probabilistic intersection problem, and stochastic linear system).

Best Approximation Problem and Sketch and Project Method: In [46, 29], it has been shown that for the case of consistent linear systems with multiple solutions, SGD (and as a result SNM (5) and SPPM (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 (2). This naturally leads to the *best approximation problem*:

$$\min_{x\in\mathbb{R}^n} P(x) := \tfrac{1}{2}\|x-x_0\|^2_{\mathbf{B}} \quad \text{subject to} \quad \mathbf{A}x=b. \tag{7}$$

[1] The gradient is computed with respect to the inner product $\langle \mathbf{B}x, y\rangle$.

[2] 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. When $\mathbf{B}=\mathbf{I}$, the $\mathbf{B}$ pseudoinverse specializes to the standard Moore-Penrose pseudoinverse.

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

Unlike, the linear system (2) which is allowed to have multiple solutions, the best approximation problem has always (from its construction) a unique solution. For solving problem (7), the *Sketch and Project Method (SPM)*:

$$x_{k+1} = \omega \Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k) + (1-\omega)x_k, \tag{8}$$

was analyzed in [18, 46]. 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 (8) simplifies to

$$x_{k+1} = \Pi_{\mathcal{L}_{\mathbf{S}},\mathbf{B}}(x_k), \tag{9}$$

first proposed in [18]. The name *Sketch and Project method* is justified by the iteration structure which follows two steps: (i) Choose the *sketched* system $\mathcal{L}_{\mathbf{S}_k} := \{x \ : \ \mathbf{S}^\top \mathbf{A} x = \mathbf{S}^\top b\}$, (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 SDSA: The Fenchel dual of (7) is the (bounded) unconstrained concave quadratic maximization problem

$$\max_{y \in \mathbb{R}^m} D(y) := (b - \mathbf{A}x_0)^\top y - \tfrac{1}{2}\|\mathbf{A}^\top y\|^2_{\mathbf{B}^{-1}}. \tag{10}$$

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 [29].

A direct dual method for solving problem (10) was first proposed in [19]. The dual method—*Stochastic Dual Subspace Ascent (SDSA)*— updates the dual vectors y_k as follows:

$$y_{k+1} = y_k + \omega \mathbf{S}_k \lambda_k, \tag{11}$$

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

$$\lambda_k = \left(\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top \left(b - \mathbf{A}(x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k)\right). \tag{12}$$

It can be proved, [19, 29], that the iterates $\{x_k\}_{k\geq 0}$ of the sketch and project method (8) arise as affine images of the iterates $\{y_k\}_{k\geq 0}$ of the dual method (11) as follows:

$$x_k = x(y_k) = x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k. \tag{13}$$

In [19] the dual method was analyzed for the case of unit stepsize ($\omega = 1$). Later in [29] the analysis extended to capture the cases of $\omega \in (0, 2)$. Momentum variants of the dual method that provide further speed up have been also studied in [29].

An interesting property that holds between the suboptimalities of the Sketch and Project method 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 [19, 29]. That is,

$$D(y_*) - D(y) = \frac{1}{2}\|x(y_*) - x(y)\|^2_{\mathbf{B}}. \tag{14}$$

This simple to derive result (by combining the expression of the dual function $D(y)$ (10) and the equation (13)) gives for free the convergence analysis of SDSA, in terms of dual function suboptimality once the analysis of Sketch and Project 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 to 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 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 (4), (5), (6) for solving the stochastic optimization problem (1) and algorithm (8) for solving the best approximation problem (7). It is an inexact variant of the *Basic Method*, first presented in [46], 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 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 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 linear rate of decay of the quantity $\mathbb{E}[\|x_k - x_*\|^2_{\mathbf{B}}]$ (i.e. L2 convergence)[4].

Subsequently, in Section 4 we apply our general convergence results to a more structured notion of inexactness error and propose a concrete mechanisms leading to such errors. We provide theoretical guarantees for this method in situations when a linearly convergent iterative method (e.g., Conjugate Gradient) 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 (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 the Appendix C. In particular, with boldface upper-case letters we denote matrices and $\mathbf{I}$ is the identity matrix. By $\mathcal{L}$

[4] As we explain later, a convergence of the expected function values of problem 1 can be easily obtained as a corollary of L2 convergence.

Assumption on the Inexactness error ϵ_k	ω	Upper Bounds	Theorem
Assumption 1*a*	$(0,2)$	$\rho^{k/2}\|x_0-x_*\|_{\mathbf{B}}+\sum_{i=0}^{k-1}\rho^{\frac{k-1-i}{2}}\sigma_i$	1
Assumption 1*b*	$(0,2)$	$\left(\sqrt{\rho}+q\right)^{2k}\|x_0-x_*\|_{\mathbf{B}}^2$	2
Assumptions 1,2	$(0,2)$	$\rho^k\|x_0-x_*\|_{\mathbf{B}}^2+\sum_{i=0}^{k-1}\rho^{k-1-i}\bar{\sigma}_i^2$	3(i)
Assumptions 1*b*,2	$(0,2)$	$\left(\rho+q^2\right)^k\|x_0-x_*\|_{\mathbf{B}}^2$	3(ii)
Assumptions 1*c*,2	$(0,2)$	$\left(\rho+q^2\lambda_{\min}^+\right)^k\|x_0-x_*\|_{\mathbf{B}}^2$	3(iii)

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 appear in the convergence results (here $\lambda_{\min}^+$ denotes the minimum non zero eigenvalue of matrix $\mathbf{W}$, see equation (19)). Inexactness parameter q is chosen always 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 1 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 $\mathbb{E}\left[D(y_*)-D(y_0)\right]$.

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 sub-problem and $\cdot^{\approx}$ to indicate its inexact variant. Unless stated 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

$$\Pi_{\mathcal{L},\mathbf{B}}(x):=\arg\min_{x'\in\mathcal{L}}\|x'-x\|_{\mathbf{B}}=x-\mathbf{B}^{-1}\mathbf{A}^\top(\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top)^\dagger(\mathbf{A}x-b). \tag{15}$$

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 into 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[5]:

$$\mathbf{Z}:=\mathbf{A}^\top\mathbf{H}\mathbf{A}=\mathbf{A}^\top\mathbf{S}(\mathbf{S}^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S})^\dagger\mathbf{S}^\top\mathbf{A}. \tag{16}$$

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) can be expressed as

$$f_{\mathbf{S}}(x)=\frac{1}{2}(\mathbf{A}x-b)^\top\mathbf{H}(\mathbf{A}x-b)=\frac{1}{2}(x-x_*)^\top\mathbf{Z}(x-x_*), \tag{17}$$

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

$$\nabla f_{\mathbf{S}}(x)\overset{(3)}{=}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{H}(\mathbf{A}x-b)=\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{H}\mathbf{A}(x-x_*)=\mathbf{B}^{-1}\mathbf{Z}(x-x_*), \tag{18}$$

and $\nabla^2 f_{\mathbf{S}}(x)=\mathbf{B}^{-1}\mathbf{Z}$ [46]. Similarly the gradient and Hessian of the objective function f of problem (1) are $\nabla f(x)=\mathbf{B}^{-1}\mathbb{E}\left[\mathbf{Z}\right](x-x_*)$ and $\nabla^2 f(x)=\mathbf{B}^{-1}\mathbb{E}\left[\mathbf{Z}\right]$, respectively.

A key matrix in our analysis is

$$\mathbf{W}:=\mathbf{B}^{-\frac{1}{2}}\mathbb{E}[\mathbf{Z}]\mathbf{B}^{-\frac{1}{2}}, \tag{19}$$

which has the same spectrum with the matrix $\nabla^2 f(x)$ but at the same time is symmetric and positive semi-definite[6]. We denote with $\lambda_1\leq\lambda_2\leq\cdots\leq\lambda_n$ the n eigenvalues of $\mathbf{W}$. With $\lambda_{\min}^+$ we indicate the smallest nonzero eigenvalue, and with $\lambda_{\max}=\lambda_n$ the largest eigenvalue. It was shown in [46] that $0\leq\lambda_i\leq 1$ for all $i\in[n]$.

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

[6] Note that matrix $\nabla^2 f(x)$ is not symmetric but it is self-adjoint with respect to the $\mathbf{B}$-inner product.

2 Inexact update rules

In this section we start by explaining the key sub-problems that need to be solved exactly in the update rules of the previously described methods. We present iBasic, a method that solves problems (1) and (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 Sub-problems in Update Rules

Let us devote this subsection on explaining how the inexactness can be introduced in the current exact update rules of SGD[7] (4), Sketch and Project (8) and SDSA (11) for solving the stochastic optimization, 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 in the update rule of each one of them can have different interpretation.

For example for the case of SGD for solving the stochastic optimization problem (1) (see also Section 4.1 and 4.2 for more details), 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 $\nabla f_{\mathbf{S}_k}(x_k) \overset{(18)}{=} -\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_k^*$ 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 to use an approximation $\lambda_k^{\approx}$ of the exact solution and with this way avoid executing the possibly expensive step of the update rule. Thus the inexact update is taking 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 to be always 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 an 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 similar way, the expensive operation of SPM (8) is the exact computation of the projection $\Pi^*_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k)$. Thus we are suggesting to replace this step with an inexact variant and compute an approximation of this projection. The inexactness here can be also interpreted using both, the abstract ϵ_k error and its more structured version $\epsilon_k = \omega \left(\Pi^{\approx}_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k) - \Pi^*_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k) \right)$. At this point, observe that, by using the expression (15) the structure of the ϵ_k in SPM and SGD has the same form.

In the SDSA the expensive subproblem in the update rule is the computation of the λ_k^* that satisfy $\lambda_k^* \in \arg\max_\lambda D(y_k + \mathbf{S}_k \lambda)$. Using the definition of the dual function (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 sub-problem that needs to be solved in each algorithm as well as the part where the inexact error is appeared in the update rule.

[7] Note that SGD has identical updates to the Stochastic Newton and Stochastic proximal point method. Thus the inexactness can be added to these updates in similar way.

Exact Algorithms	Key Subproblem (problem that we solve inexactly)	Inexact Update Rules (abstract and structured inexactness error)
SGD (4)	Exact computation of λ_k^*, where $\lambda_k^* = \arg\min_{\lambda:\mathbf{M}_k\lambda=d_k} \|\lambda\|$. 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 (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 (11)	Exact computation of λ_k^*, where $\lambda_k^* \in \arg\max_\lambda D(y_k + \mathbf{S}_k\lambda)$.	$y_{k+1} = y_k + \omega\mathbf{S}_k\lambda_k^{\approx} = y_k + \omega\mathbf{S}_k\lambda_k^* + \underbrace{\omega\mathbf{S}_k(\lambda_k^{\approx} - \lambda_k^*)}_{\epsilon_k^d}$

Table 2: The exact algorithms under study with the potentially expensive to compute key sub-problems of their update rule. The inexact update rules are presented in the last column for both notions of inexactness (abstract and more 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.

2.2 The Inexact Basic Method

In each iteration of the all 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 $\mathbf{S} \in \mathbb{R}^{m\times q}$ requires to have m rows but no assumption on the number of columns is made which means that the number of columns q allows to vary through 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 requires to solve large subproblems in each iteration have been extensively studied and it was shown that are really beneficial when they compared to their single coordinates variants ($\mathbf{S} \in \mathbb{R}^{m\times 1}$) [34, 35, 44, 27]. However, in theses cases the evaluation of an exact solution for the suproblem in the update rule can be computationally very expensive. In this work we propose and analyze inexact variants by allowing to solve the subproblem that appear in the update rules of the stochastic methods, inexactly. In particular, following the convention established in [46] of naming the main algorithm of the paper *Basic method* we propose the *inexact Basic method (iBasic)* (Algorithm 1).

Algorithm 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: $x_0 \in \mathbb{R}^n$

1: **for** $k = 0, 1, 2, \cdots$ **do**
2: Generate a fresh sample $\mathbf{S}_k \sim \mathcal{D}$
3: Set $x_{k+1} = x_k - \omega\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k(\mathbf{S}_k^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k)^\dagger\mathbf{S}_k^\top(\mathbf{A}x_k - b) + \epsilon_k$
4: **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 allow as to study the methods (4),(5),(6) for solving the stochastic optimization problem and the sketch and project method (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). From (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 approximation

problem (7), iBasic can be interpreted as inexact Sketch and Project method (iSPM) as follows:

$$\begin{aligned} x_{k+1} &= x_k - \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k (\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top (\mathbf{A}x_k - b) + \epsilon_k \\ &= \omega \left[x_k - \mathbf{B}^{-1}(\mathbf{S}_k^\top \mathbf{A})^\top (\mathbf{S}_k^\top \mathbf{A} B^{-1} (\mathbf{S}_k^\top \mathbf{A})^\top)^\dagger (\mathbf{S}_k^\top \mathbf{A}x - \mathbf{S}_k^\top b) \right] + (1-\omega)x_k + \epsilon_k \\ &\overset{(15)}{=} \omega \Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k) + (1-\omega)x_k + \epsilon_k \end{aligned} \tag{20}$$

For the dual problem (10) we devote Section 5 for presenting an inexact variant of the SDSA (iSDSA) and analyze its convergence using the rates obtained for the iBasic in Sections 3 and 4.

2.3 General Framework and Further Special Cases

The proposed inexact methods, iBasic (Algorithm 1) and iSDSA (Section 5), belong in the general *sketch and project* framework, first proposed from Gower and Richtarik in [18] for solving consistent linear systems and where a unified analysis of several randomized methods was studied. This interpretation of the algorithms allow 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, the 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 the Randomized Kaczmarz, Randomized Gaussian Kaczmarz[8], 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 [18] and Section 7 of [29]. Here we present only the inexact update rules of two special cases that we will later use in the numerical evaluation.

Special Cases: Let us define with $\mathbf{I}_{:C}$ 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 pick in each iteration the random matrix $\mathbf{S} = \mathbf{I}_{:C} \sim \mathcal{D}$. In this setup the update rule of the iBasic simplifies to

$$x_{k+1} = x_k - \omega \mathbf{A}_{C:}^\top (\mathbf{A}_{C:}\mathbf{A}_{C:}^\top)^\dagger (\mathbf{A}_{C:}x_k - b_C) + \epsilon_k. \tag{21}$$

- *Inexact Randomized Block Coordinate Descent (iRBCD)*[9]: If the matrix $\mathbf{A}$ of the linear system is positive definite then we can choose $\mathbf{B} = \mathbf{A}$. Let also pick in each iteration the random matrix $\mathbf{S} = \mathbf{I}_{:C} \sim \mathcal{D}$. In this setup the update rule of the iBasic simplifies to

$$x_{k+1} = x_k - \omega \mathbf{I}_{:C} (\mathbf{I}_{:C}^\top \mathbf{A} \mathbf{I}_{:C})^\dagger \mathbf{I}_{:C}^\top (\mathbf{A}x_k - b) + \epsilon_k. \tag{22}$$

For more papers related to Kaczmarz method (randomized, greedy, cyclic update rules) we refer the interested reader to [23, 28, 38, 5, 37, 39, 9, 33, 34, 13, 31, 59, 35, 50]. For the coordinate descent method (a.k.a Gauss-Seidel for linear systems) and its block variant, Randomized Block Coordinate Descent we suggest [25, 36, 44, 45, 40, 41, 43, 7, 24, 14, 1, 54].

2.4 Other Related Work on Inexact Methods

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

[8] Special case of the 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 $\mathbf{S} \sim N(0, \Sigma)$ [18, 29].

[9] In the setting of solving linear systems Randomized Coordinate Descent is known also as Gauss-Seidel method. Its block variant can be also interpret as randomized coordinate Newton method (see [42]).

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 [49, 11, 51, 16, 32]. 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 the exact gradient descent is apparent. For further deterministic inexact methods check [10] for Inexact Newton methods, [52, 47] for Inexact Proximal Point methods and [3] for Inexact Fixed point methods.

In the recent years, with the explosion that happens in areas like machine learning and data science inexactness enters also 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 introduced to the update rules of several stochastic optimization algorithms and new methods have been proposed and analyzed. One of the most interesting work on inexact stochastic algorithms appears in the area of second order methods. In particular on inexact variants of the Sketch-Newton method and subsampled Newton Method for minimize convex and non-convex functions [48, 2, 4, 56, 57, 58]. Note that our results are related also with this literature since our algorithm can be seen as inexact stochastic Newton method (see equation (5)). To the best or our knowledge our work is the first that provide convergence analysis of inexact stochastic proximal point methods (equation (6)) in any setting. From numerical linear algebra viewpoint inexact sketch and project methods for solving the best approximation problem and its dual problem where also never analyzed before.

As we already mentioned our framework is quite general and many algorithms, like iRBK (21) and iRBCD (22) 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 [34] an analysis of the exact randomized block Kacmzarz method has been proposed and in the experiments an inexact variant was used to speedup the method. However, no iteration complexity results were presented for the inexact variant and both the analysis and numerical evaluation have been made for linear systems with full rank matrices that come with 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 [53, 15, 6, 12].

Finally an analysis of approximate stochastic gradient descent for solving the empirical risk minimization problem using quadratic constraints and sequential semi-definite programs has been presented in [22].

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 be hold in all of our results is exactness. A formal presentation is presented below. We state it here and we highlight that is a requirement for all of our convergence results (exactness is also required in the analysis of the exact variants [46]).

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 [46]. 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) and $\mathcal{L} = \{x : \mathbf{A}x = b\}$ the solution set of the linear system (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 1*a*, 1*b* and 1*c* are special cases of Assumption 1. Moreover Assumption 2 is algorithmic dependent and can hold in addition of any of the other four assumptions. In our analysis, depending on the result we aim at, we will require either one of the first four Assumptions to hold by itself, or to hold together with Assumption 2. 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 to mention that for the convergence analysis we never assume that the inexactness error has zero mean, that is $\mathbb{E}[\epsilon_k] = 0$.

Assumption 1.

$$\mathbb{E}[\|\epsilon_k\|_{\mathbf{B}}^2 \mid x_k, \mathbf{S}_k] \leq \sigma_k^2, \tag{23}$$

where the upper bound σ_k is a sequence of random variables (that can possibly depends 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 allow us to obtain stronger and more controlled results.

Assumption 1a.

$$\mathbb{E}[\|\epsilon_k\|_{\mathbf{B}}^2 \mid x_k, \mathbf{S}_k] \leq \sigma_k^2, \tag{24}$$

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

Assumption 1b.

$$\mathbb{E}[\|\epsilon_k\|_{\mathbf{B}}^2 \mid x_k, \mathbf{S}_k] \leq \sigma_k^2 = q^2\|x_k - x_*\|_{\mathbf{B}}^2, \tag{25}$$

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

Assumption 1c.

$$\mathbb{E}[\|\epsilon_k\|_{\mathbf{B}}^2 \mid x_k, \mathbf{S}_k] \leq \sigma_k^2 = 2q^2 f_{\mathbf{S}_k}(x_k), \tag{26}$$

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

Finally the next assumption is more algorithmic oriented. It holds in cases where the inexactness error ϵ_k in the update rule is chosen to be orthogonal with respect to the $\mathbf{B}$-inner product to the vector $\Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k) - x_* = (\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k - x_*)$. This statement may seem odd at this point but its usefulness will become more apparent in the next section where inexact algorithms with structured inexactness error will be analyzed. As it turns out, in the case of structured inexactness error (Algorithm 2) this assumption is satisfied.

Assumption 2.

$$\mathbb{E}[\langle(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k - x_*), \epsilon_k\rangle_{\mathbf{B}}] = 0. \tag{27}$$

3.2 Convergence Results

In this section we present the analysis of the convergence rates of iBasic by assuming several combination 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 10) we can directly deduce a convergence rate for the function values.

The exact Basic method (Algorithm 1 with $\epsilon_k = 0$), has been analyzed in [46] 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 special case when we

assume that the upper bound is $\sigma_k = 0, \quad \forall k \geq 0$. For brevity, in he 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 1*a* holds for the inexactness error.

Theorem 1. *Let 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 be such that it satisfies Assumption 1a. Then,*

$$\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}] \leq \rho^{k/2}\|x_0 - x_*\|_{\mathbf{B}} + \sum_{i=0}^{k-1} \rho^{\frac{k-1-i}{2}}\sigma_i. \tag{28}$$

Proof. See Appendix B.1. □

Corollary 1. *In the special case that the upper bound σ_k in Assumption 1a is fixed, that is $\sigma_k = \sigma$ for all $k > 0$ then inequality (28) of Theorem 1 takes the following form:*

$$\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}] \leq \rho^{k/2}\|x_0 - x_*\|_{\mathbf{B}} + \sigma\frac{\rho^{1/2}}{1-\rho}. \tag{29}$$

This means that we obtain a linear convergence rate up to a solution level that is proportional to the upper bound σ[10]*.*

Proof. See Appendix B.2. □

Inspired from [16], let us now analyze iBasic using the sequence of upper bounds that described in Assumption 1*b*. This construction of the upper bounds allows us to obtain stronger and more controlled results. In particular using the upper bound of Assumption 1*b* the sequence of expected errors converge linearly to the exact x_* (not in a potential neighborhood like the previous result). In addition Assumption 1*b* guarantees that the distance to the optimal solution reduces with the increasing 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 that this value has a close form expression and can be computed before hand without any further cost. See for example [27, 30, 26, 21] 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 2. *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 be such that it satisfies Assumption 1b, with $0 \leq q < 1 - \sqrt{\rho}$. Then*

$$\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}^2] \leq \left(\sqrt{\rho} + q\right)^{2k}\|x_0 - x_*\|_{\mathbf{B}}^2. \tag{30}$$

Proof. See Appendix B.3. □

At Theorem 2, 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 1. Not surprisingly the fastest convergence rate is obtained when $q = 0$; in such case the method becomes equivalent with 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 $\omega = 1$ [46].

Moreover, the upper bound σ_k of Assumption 1*b* 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 natural interpretation. In particular the inexactness error is allowed

[10] 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 (28) of Theorem 1 takes the form: $\mathbb{E}[\|x_k - x_*\|_{\mathbf{B}}] \leq O(C^k)$ (see [51, 16] for similar results).

to be large either when the current iterate is far from the optimal solution ($\|x_k - x_*\|^2_{\mathbf{B}}$ large) or when the problem is well conditioned and $\lambda^+_{\min}$ is large. In the opposite scenario, when we have ill conditioned problem or we are already close enough to the optimum x_* we should be more careful and allow less errors to the updates of the method.

In the next theorem we provide the complexity results of iBasic in the case that the Assumption 2 is satisfied combined with one of the previous assumptions.

Theorem 3. *Let 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 also assume that the inexactness error ϵ_k be such that it satisfies Assumption 2. Then:*

(i) If Assumption 1 holds:

$$\mathbb{E}[\|x_k - x_*\|^2_{\mathbf{B}}] \leq \rho^k \|x_0 - x_*\|^2_{\mathbf{B}} + \sum_{i=0}^{k-1} \rho^{k-1-i} \bar{\sigma}_i^2, \tag{31}$$

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

(ii) If Assumption 1b holds with $q \in \left(0, \sqrt{\rho}\right)$:

$$\mathbb{E}[\|x_k - x_*\|^2_{\mathbf{B}}] \quad \leq \quad (\rho + q^2)^k \|x_0 - x_*\|^2_{\mathbf{B}}. \tag{32}$$

(iii) If Assumption 1c holds with $q \in \left(0, \sqrt{\omega(2-\omega)}\right)$:

$$\mathbb{E}[\|x_k - x_*\|^2_{\mathbf{B}}] \leq (1 - (\omega(2-\omega) - q^2)\lambda^+_{\min})^k \|x_0 - x_*\|^2_{\mathbf{B}} = (\rho + q^2\lambda^+_{\min})^k \|x_0 - x_*\|^2_{\mathbf{B}}. \tag{33}$$

Proof. See Appendix B.4. □

Remark 1. *In the case that Assumptions 1a and 2 hold simultaneously, the convergence of iBasic is similar to (31) but in this case $\bar{\sigma}_i^2 = \sigma_i^2$, $\forall i \in [k-1]$ (due to Assumption 1a, $\sigma_k \in \mathbb{R}$ is a sequence of real numbers). In addition, note that for $q \in (0, \min\{\sqrt{\rho}, 1-\sqrt{\rho}\})$ having Assumption 2 on top of Assumption 1b leads to improvement of the convergence rate. In particular, from Theorem 2, 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 $\rho + q^2$ (32).*

4 iBasic with Structured Inexactness Error

Up to this point, the analysis of iBasic was focused in more general abstract cases where the inexactness error ϵ_k of the update rule satisfies several general assumptions. In this section we are focusing 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 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[11] 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 $\mathcal{Q}_k = \{\lambda \in \mathbb{R}^q : \mathbf{M}_k \lambda = d_k\}$.

[11] 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 $\mathbf{M}_k\lambda = d_k$.

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:

Algorithm 2 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: $x_0 \in \mathbb{R}^n$

1: **for** $k = 0, 1, 2, \cdots$ **do**

2: Generate a fresh sample $\mathbf{S}_k \sim \mathcal{D}$

3: Using an iterative method compute an approximation $\lambda_k^{\approx}$ of the least norm solution of the linear system:

$$\underbrace{\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k}_{\mathbf{M}_k} \lambda = \underbrace{\mathbf{S}_k^\top (b - \mathbf{A}x_k)}_{d_k}. \tag{34}$$

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

5: **end for**

For the computation of the inexact solution of the linear system (34) 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 (34), conjugate gradient (CG) or sketch and project method (SPM) can be used for the execution of step 3. In these cases, we name Algorithm 2, *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 to find the least norm solution of $\mathbf{M}_k \lambda_k = d_k$ which means that the starting point of the sketch and project at the k^{th} iteration should be always $\lambda_k^0 = 0$. Recall that any special case of the sketch and project method (Section 2.3) solves the best approximation problem.

Let us now define λ_k^r to be the approximate solution $\lambda_k^{\approx}$ of the $q \times q$ linear system (34) obtained after r steps of the linearly convergent iterative method. Using this, the update rule of Algorithm 2, takes the form:

$$x_{k+1} = x_k + \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_k^r. \tag{35}$$

Remark 2. *The update rule (35) of Algorithm 2 is equivalent to the update rule of iBasic (Algorithm 1) when the error ϵ_k is chosen to be,*

$$\epsilon_k = \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k (\lambda_k^r - \lambda_k^*). \tag{36}$$

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

Let us now define a Lemma that is useful for the analysis of this section and it verifies that Algorithm 2 with unit stepsize satisfies the general Assumption 2 presented in Section 3.1.

Lemma 4. *Let us denote $x_k^* = \Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x_k)$ 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 2 it holds that:*

$$\langle x_k^* - x_*, \epsilon_k \rangle_{\mathbf{B}} = \left\langle (\mathbf{I} - \omega \mathbf{B}^{-1}\mathbf{Z}_k)(x_k - x_*), \epsilon_k \right\rangle_{\mathbf{B}} = 0, \quad \forall k \geq 0. \tag{37}$$

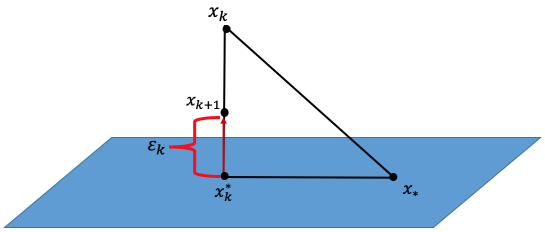


Figure 1: Graphical interpretation of orthogonality (justifies equation (37)). It shows that the two vectors, $x_k^* - x_*$ and ϵ_k, are orthogonal complements of each other with respect to the $\mathbf{B}$-inner product. x_{k+1} is the point that Algorithm 2 computes in each step. The colored region 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.

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 $\epsilon_k \overset{(36)}{=} \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k(\lambda_k^r - \lambda_k^*) \in Range(\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)$. 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 2 using the sketch and project approach. That will make us appreciate more 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 equation (9)) the exact sketch and project method perform updates of the form:

$$x_{k+1} = \text{argmin}_{x \in \mathbb{R}^n} \tfrac{1}{2}\|x - x_k\|_{\mathbf{B}}^2 \quad \text{subject to} \quad \mathbf{S}_k^\top \mathbf{A} x = \mathbf{S}_k^\top b. \tag{38}$$

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

In general, execute a projection step is one of the most common task 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 to move to the dual space where the inexactness can be easily controlled.

Observe that the update rule of equation (38) has the same structure as the best approximation problem (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:

$$\max_{\lambda \in \mathbb{R}^q} D_k(\lambda) := (\mathbf{S}_k^\top b - \mathbf{S}_k^\top \mathbf{A} x_k)^\top \lambda - \tfrac{1}{2}\|\mathbf{A}^\top \mathbf{S}_k \lambda\|_{\mathbf{B}^{-1}}^2. \tag{39}$$

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 with 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 2. Thus, computing an inexact solution $\lambda_k \approx$ of the linear system is equivalent with computing an inexact solution of the dual problem (39). Then by using the affine mapping (13) that connects the primal and the dual spaces we can also evaluate an inexact solution of the original primal problem (38).

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 5. *Let us define $\lambda_k^* \in \mathbb{R}^q$ 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 dual problem (39). Let us also denote with $\lambda_k^{\approx} \in \mathbb{R}^q$ the inexact solution. Then:*

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

Proof.

$$\begin{aligned}
D_k(\lambda_k^*) - D_k(\lambda_k^{\approx}) &\overset{(39)}{=} [\mathbf{S}_k^\top b - \mathbf{S}_k^\top \mathbf{A}x_k]^\top [\lambda_k^* - \lambda_k^{\approx}] - \frac{1}{2}(\lambda_k^*)^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_k^* \\
&\quad + \frac{1}{2}(\lambda_k^{\approx})^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_k^{\approx} \\
&\overset{(13)}{=} (\lambda_k^*)^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k [\lambda_k^* - \lambda_k^{\approx}] - \frac{1}{2}(\lambda_k^*)^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_k^* \\
&\quad + \frac{1}{2}(\lambda_k^{\approx})^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_k^{\approx} \\
&= \frac{1}{2}(\lambda_k^{\approx} - \lambda_k^*)^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k (\lambda_k^{\approx} - \lambda_k^*) \\
&= \frac{1}{2}\|\lambda_k^{\approx} - \lambda_k^*\|^2_{\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k}
\end{aligned}$$

where in the second equality we use equation (13) to connect the optimal solutions of (38) and (39) 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 2 when a linearly convergent iterative method is used for solving inexactly the linear system (34) in step 3 of Algorithm 2 . We denote with λ_k^r the approximate solution of the linear system after we run the iterative method for r steps.

Before state the main convergence result let us present a lemma that summarize some observations that are true in our setting.

Lemma 6. *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 approximate solution of the linear system (34). 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.

$$\begin{aligned}
\|\lambda_k^*\|^2_{\mathbf{M}_k} &= \|\mathbf{M}_k^\dagger \mathbf{S}_k^\top \mathbf{A}(x_* - x_k)\|^2_{\mathbf{M}_k} = (x_k - x_*)^\top \mathbf{A}^\top \mathbf{S}_k \underbrace{\mathbf{M}_k^\dagger \mathbf{M}_k \mathbf{M}_k^\dagger}_{\mathbf{M}_k^\dagger} \mathbf{S}_k^\top \mathbf{A}(x_k - x_*) \\
&\overset{(16)}{=} (x_k - x_*)^\top \mathbf{Z}_k (x_k - x_*) \overset{(17)}{=} 2f_{\mathbf{S}_k}(x_k). \qquad (40)
\end{aligned}$$

Moreover,

$$\|\epsilon_k\|^2_{\mathbf{B}} \overset{Remark\ 2}{=} \|\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k(\lambda_k^r - \lambda_k^*)\|^2_{\mathbf{B}} = \|\lambda_k^r - \lambda_k^*\|^2_{\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k} = \|\lambda_k^r - \lambda_k^*\|^2_{\mathbf{M}_k}. \qquad (41)$$

□

Theorem 7. *Let us assume that for the computation of the inexact solution of the linear system (34) in step 3 of Algorithm 2, a linearly convergent iterative method is chosen such that [12]:*

$$\mathbb{E}[\|\lambda_k^r - \lambda_k^*\|^2_{\mathbf{M}_k} \mid x_k, \mathbf{S}_k] \leq \rho^r_{\mathbf{S}_k} \|\lambda_k^0 - \lambda_k^*\|^2_{\mathbf{M}_k}, \qquad (42)$$

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 2 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, $\rho_{\mathbf{S}_k} \leq \theta$. Then, Algorithm 2 converges linearly with:

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

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

Proof. Theorem 7 can be interpreted as corollary of the general Theorem 3(iii). Thus, it is sufficient to show that Algorithm 2 satisfies the two Assumptions 1*c* and 2. Firstly, note that from Lemma 4, Assumption 2 is true. Moreover,

$$\begin{aligned}\mathbb{E}[\|\epsilon_k\|^2_{\mathbf{M}_k} \mid x_k, \mathbf{S}_k] &\overset{(41)}{=} \mathbb{E}[\|\lambda^r_k - \lambda^*_k\|^2_{\mathbf{M}_k} \mid x_k, \mathbf{S}_k] \overset{(42)}{\leq} \rho^r_{\mathbf{S}_k}\|\lambda^0_k - \lambda^*_k\|^2_{\mathbf{M}_k} \\ &\leq \theta^r\|\lambda^0_k - \lambda^*_k\|^2_{\mathbf{M}_k} \overset{\lambda^0_k=0}{=} \theta^r\|\lambda^*_k\|^2_{\mathbf{M}_k} \overset{(40)}{=} 2\theta^r f_{\mathbf{S}_k}(x_k)\end{aligned}$$

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

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

Remark 3. *From its definition $\theta^r \in (0,1)$ and as a result $(1-\theta^r)\,\lambda^+_{\min} \leq \lambda^+_{\min}$. This means that the method converges linearly but always with worst rate than its exact variant.*

Remark 4. *Let us assume that θ is fixed. Then as the number of iterations in step 3 of the algorithm ($r \to \infty$) increasing $(1-\theta^r) \to 1$ and as a result the method behaves similar to the exact case.*

Remark 5. *The $\lambda^+_{\min}$ depends only on the random matrices $\mathbf{S} \sim \mathcal{D}$ and to the positive definite matrix $\mathbf{B}$ and is independent to the iterative process used in step 3. The iterative process of step 3 controls only the parameter θ of the convergence rate.*

Remark 6. *Let us assume that we run Algorithm 2 two separate times for two different choices of the linearly convergence iterative method of step 3. Let 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 3 the iterative method run for r steps for both algorithms. Let assume that $\theta_1 < \theta_2$ then we have that $\rho_1 = 1-(1-\theta^r_1)\,\lambda^+_{\min} < 1-(1-\theta^r_2)\,\lambda^+_{\min} = \rho_2$. This means in the case that θ is easily computable, we should always prefer the inexact method with smaller θ.*

The convergence of Theorem 7 is quite general and it holds for any linearly convergent methods that can inexactly solve (34). However, in case that the iterative method is known we can have more concrete results. See below the more specified results for the cases of Conjugate gradient (CG) and Sketch and project method (SPM).

Convergence of InexactCG: CG is 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 finite number of iterations. In particular, it can be shown that converges to the unique solution in at most n steps. The worst case behavior of CG is given by [55, 17] [13]:

$$\|x_k - x_*\|_{\mathbf{A}} \leq \left(\frac{\sqrt{\kappa(\mathbf{A})}-1}{\sqrt{\kappa(\mathbf{A})}+1}\right)^{2k}\|x_0 - x_*\|_{\mathbf{A}}, \tag{43}$$

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

Having present the convergence of CG for general linear systems, let us now return back to our setting. We denote $\lambda^r_k \in \mathbb{R}^q$ to be the approximate solution of the inner linear system (34) after r conjugate gradient steps. Thus using (43) we know that $\|\lambda^r_k - \lambda^*_k\|^2_{\mathbf{M}_k} \leq \rho^{4r}_{\mathbf{S}_k}\|\lambda^0_k - \lambda^*_k\|^2_{\mathbf{M}_k}$, where $\rho_{\mathbf{S}_k} = \left(\frac{\sqrt{\kappa(\mathbf{M}_k)}-1}{\sqrt{\kappa(\mathbf{M}_k)}+1}\right)$. Now by making the same assumption as the general Theorem 7 the InexactCG converges with $\mathbb{E}[\|x_k - x_*\|^2_{\mathbf{B}}] \leq \left[1-(1-\theta^r_{CG})\,\lambda^+_{\min}\right]^k\|x_0 - x_*\|^2_{\mathbf{B}}$, where $\theta_{CG} < 1$ such that $\rho_{\mathbf{S}_k} = \left(\frac{\sqrt{\kappa(\mathbf{M}_k)}-1}{\sqrt{\kappa(\mathbf{M}_k)}+1}\right)^4 \leq \theta_{CG}$ with probability 1.

[13] A sharper convergence rate of CG [55] for solving $\mathbf{A}x = b$ can be also used

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

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

Convergence of InexactSP: In this setting we suggest to run the sketch and project method (SPM) for solving inexactly the linear system (34). 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 compared to what problems InexactCG can solve[14]. Like before, by making the same assumptions as in Theorem 7 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 the InexactSP can be obtained. Now the quantity $\rho_{\mathbf{S}_k}$ denotes the convergence rate of the exact Basic method[15] when this applied to solve linear system (34) 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) and the best approximation (7). In this section we turn into the dual of the best approximation (10) and we propose and analyze an inexact variant of the SDSA (11). We call the new method iSDSA and is formalized as Algorithm 3. In the update rule ϵ_k^d indicates the dual inexactness error that appears in the k^{th} iteration of iSDSA.

Algorithm 3 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: $y_0 = 0 \in \mathbb{R}^m$, $x_0 \in \mathbb{R}^n$

1: **for** $k = 0, 1, 2, \cdots$ **do**

2: Draw a fresh sample $\mathbf{S}_k \sim \mathcal{D}$

3: Set $y_{k+1} = y_k + \omega \mathbf{S}_k \left(\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top \left(b - \mathbf{A}(x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k)\right) + \epsilon_k^d$

4: **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 (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 8. *(Correspondence between the primal and dual methods) Let $\{x_k\}_{k=0}^\infty$ be the iterates produced by iBasic (Algorithm 1). Let $y_0 = 0$, and $\{y_k\}_{k=0}^\infty$ 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 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.

$$\begin{aligned}
\phi(y_{k+1}) &\overset{(13)}{=} x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_{k+1} \overset{(12),\text{Alg.3}}{=} x_0 + \mathbf{B}^{-1}\mathbf{A}^\top \left[y_k + \omega \mathbf{S}_k \lambda_k + \epsilon_k^d\right] \\
&\overset{(16),(12)}{=} \underbrace{x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k}_{\phi(y_k)} + \omega \mathbf{B}^{-1}\mathbf{Z}_k \left(x_* - \underbrace{(x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k)}_{\phi(y_k)}\right) + \mathbf{B}^{-1}\mathbf{A}^\top \epsilon_k^d \\
&= \phi(y_k) - \omega \mathbf{B}^{-1}\mathbf{Z}_k(\phi(y_k) - x_*) + \mathbf{B}^{-1}\mathbf{A}^\top \epsilon_k^d
\end{aligned}$$

[14] 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)

[15] Recall that iBasic and its exact variant ($\epsilon_k = 0$) can be expressed as sketch and project methods (20).

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 element 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 4. It can be seen as a special case of iSDSA but with a more structured inexactness error.

Algorithm 4 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: $y_0 = 0 \in \mathbb{R}^m$, $x_0 \in \mathbb{R}^n$

1: **for** $k = 0, 1, 2, \cdots$ **do**

2: Generate a fresh sample $\mathbf{S}_k \sim \mathcal{D}$

3: Using an Iterative method compute an approximation $\lambda_k^{\approx}$ of the least norm solution of the linear system:

$$\underbrace{\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k}_{\mathbf{M}_k} \lambda = \underbrace{\mathbf{S}_k^\top (b - \mathbf{A}(x_0 + \mathbf{B}^{-1}\mathbf{A}^\top y_k)}_{d_k} \tag{44}$$

4: Set $y_{k+1} = y_k + \omega \mathbf{S}_k \lambda_k^{\approx}$

5: **end for**

Similar to their primal variants, it can be easily checked that Algorithm 4 is a special case of the iSDSA (Algorithm 3) when the dual inexactness error is chosen to be $\epsilon_k^d = \mathbf{S}_k(\lambda_k^r - \lambda_k^*)$. Note that, using the observation of Remark 2 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 8.

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 similar assumption to Assumption 1*b*. Similar convergence results can be obtained using any other assumption of Section 3.1. The convergence of Algorithm 4, can be also easily derived using similar arguments with the one presented in Section 4 and the convergence guarantees of Theorem 7.

Theorem 9. *(Convergence of dual objective). Assume exactness. Let $y_0 = 0$ and let $\{y_k\}_{k=0}^\infty$ to be the dual iterates of iSDSA (Algorithm 3) 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 be such that it satisfies $\mathbb{E}[\|\mathbf{B}^{-1}\mathbf{A}^\top \epsilon_k^d\|_{\mathbf{B}}^2 \mid y_k, \mathbf{S}_k] \leq \sigma_k^2 = q^2 2\left[D(y_*) - D(y_k)\right]$ where $0 \leq q < 1 - \sqrt{\rho}$. Then*

$$\mathbb{E}[D(y_*) - D(y_k)] \leq (\sqrt{\rho} + q)^{2k} \left[D(y_*) - D(y_0)\right]. \tag{45}$$

Proof. The proof follows by applying Theorem 2 together with Theorem 8 and the identity $\frac{1}{2}\|x_k - x_*\|_{\mathbf{B}}^2 = D(y_*) - D(y_k)$ (14). □

Note that in the case that $q = 0$, iSDSA simplifies to its exact variant SDSA and the convergence rate coincide with the one presented in [29, 19]. Following similar arguments to those in [19], 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 is used to solve the best approximation problem (7) or equivalently the stochastic optimization problem (1)[16]. As we have already mentioned, iBasic can be interpreted as sketch-and-project method, and as a result a comprehensive array of well-known algorithms can be recovered as special cases by varying the main parameters of the methods (Section 2.3). In particular, in our experiments we focus on the evaluation of two popular special cases, the inexact Randomized Block Kaczmarz (iRBK) (equation (21)) and inexact randomized block coordinate descent method (iRBCD) (equation (22))We implement Algorithm 2 presented in Section 4 using CG [17] to inexactly solve the linear system of the update rule (equation (34)). Recall that in this case we named the method InexactCG.

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 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 $[m]$ to have fixed pre-specified 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.

To coincide with the theoretical convergence results of Algorithm 2 the relaxation parameter (stepsize) of the methods study in our experiments is chosen to be $\omega = 1$ (no relaxation). In all implementations, we use $x_0 = 0 \in \mathbb{R}^n$ as an initial point and in comparing the methods with their inexact variants we use the relative error measure $\|x_k - x_*\|_{\mathbf{B}}^2/\|x_0 - x_*\|_{\mathbf{B}}^2 \overset{x_0=0}{=} \|x_k - x_*\|_{\mathbf{B}}^2/\|x_*\|_{\mathbf{B}}^2$. We run each method (exact and 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 exact variants, the linear system (34) in Algorithm 2 needs to be solved exactly. In our experiments we follow the implementation of [18] for both exact RBCD and exact RBK 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:

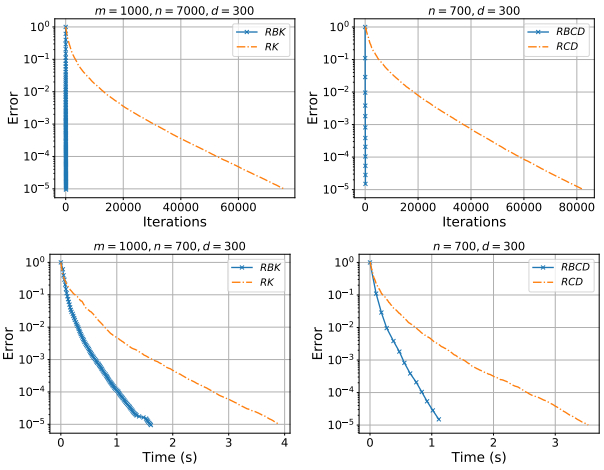
- **For iRBK:** Let matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ being given (it can be either synthetic or real data). Then a vector $z \in \mathbb{R}^n$ is chosen to be i.i.d $\mathcal{N}(0, 1)$ and the right hand side of the linear system is set to $b = \mathbf{A}z$. With this way the consistency of the linear system with matrix $\mathbf{A}$ and right hand side b is ensured.
- **For iRBCD:** A 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 (with 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 $b = \mathbf{A}z$.

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 [18, 44, 34, 27]. In Figure 2 we numerically verify this

[16] Note that from Section 5 and the correspondence between the primal and dual methods, iSDSA will have similar behavior when is applied to the dual problem (10).

[17] Recall that in order to use CG, the matrix $\mathbf{M}_k$ that appears in linear system (34) should be positive definite. This is true in the case that the matrix $\mathbf{A}$ of the original system has 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 linearly convergence methods.



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

Figure 2: Comparison of the performance of the exact RBK and RBCD with their non-block 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) $\mathbf{A} = \mathbf{P}^\top\mathbf{P} \in \mathbb{R}^{700\times700}$ where $\mathbf{P} \in \mathbb{R}^{1000\times700}$ is Gaussian matrix. To guarantee consistency $b = \mathbf{A}z$ where z is also Gaussian vector. The block size that chosen for the block variants is $d = 300$.

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 justify 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.

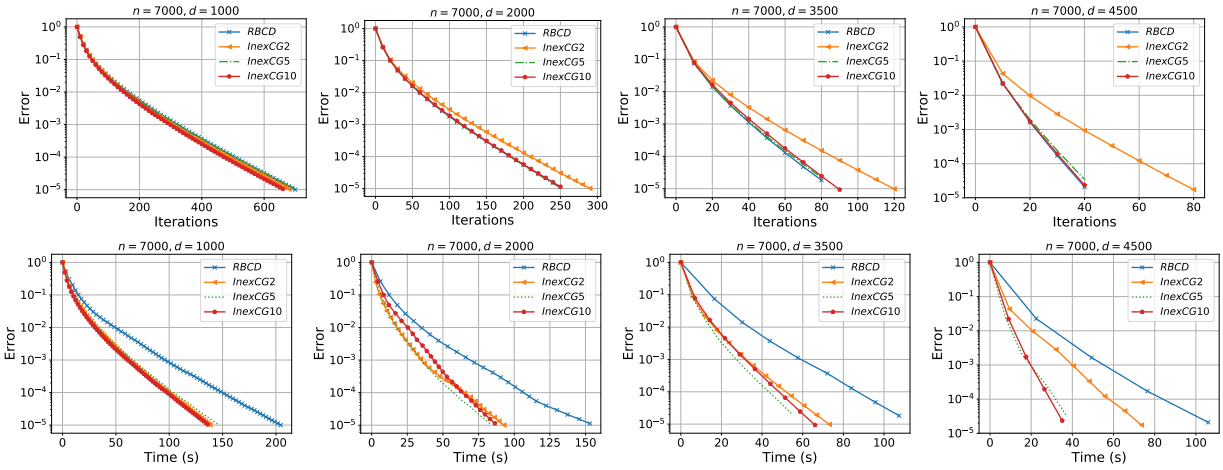
6.2 Inexactness and Block Size (iRBCD)

In this experiment, we first construct a positive definite linear system following the previously described procedure for iRBCD. We first generate a Gaussian matrix $\mathbf{P} \in \mathbb{R}^{10000\times7000}$ and then the positive definite matrix $\mathbf{A} = \mathbf{P}^\top\mathbf{P} \in \mathbb{R}^{7000\times7000}$ 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 column of matrix $\mathbf{S}$). For evaluating the inexact solution of the linear system in the update rule we run CG for either 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) 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 results the sub-problem that needs to be solved in each step becomes more expensive. It is worth to highlight that for the chosen systems, the exact RBCD behaves better in terms of wall clock time as the size of block increases (this coincides with the findings of the previous experiment).

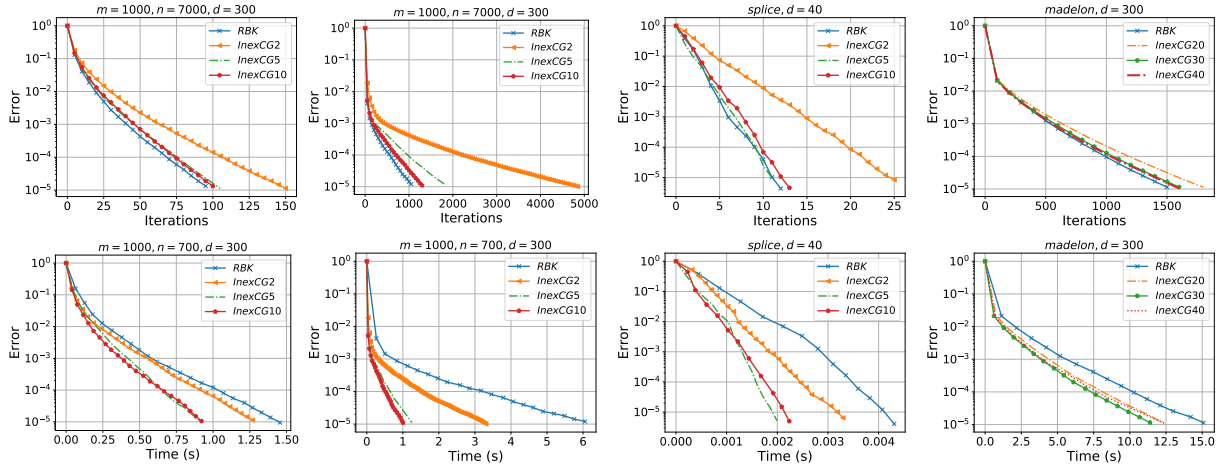
6.3 Evaluation of iRBK

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 pre-specified number of iterations that can vary depending 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 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 [8]. Each dataset of the LIBSVM consists of a matrix



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

Figure 3: Performance of iRBCD (InexactCG) and exact RBCD for solving a consistent linear systems 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 $\|x_k - x_*\|^2_{\mathbf{A}}/\|x_*\|^2_{\mathbf{A}}$.



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

Figure 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 LIBSVM [8] : (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 size of the block size for both iRBK and RBK.

$\mathbf{A} \in \mathbb{R}^{m\times n}$ (m features and n characteristics) and a vector of labels $b \in \mathbb{R}^m$. In our experiments we choose to use only the matrices of the datasets and ignore the label vectors [18]. 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.

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

7 Conclusion

In this work we propose and analyze inexact variants of several stochastic algorithms for solving quadratic optimization problems and linear systems. We provide 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, inexactness should definitely be a tool that practitioners should use in their implementations 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 [29, 46, 54] have been proposed for solving linear systems. We speculate that the addition of inexactness to these update rules will lead to methods 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 non-convex functions in the stochastic setting. Finally, sketch-and-project algorithms have been used for solving the average consensus problem [27, 20] popular in distributed optimization literature. Our results could also be useful in this area and lead to the development of novel randomized gossip algorithms that use inexactness in their update rule.

8 Acknowledgements

The first author would like to acknowledge Aritra Dutta (KAUST), Robert Mansel Gower (Télécom ParisTech), Georgios Loizou (Edinburgh) and Rachael Tappenden (University of Canterbury) 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*, pages 1110–1119, 2016.

[2] A.S. Berahas, R. Bollapragada, and J. Nocedal. An investigation of Newton-sketch and subsampled Newton methods. *arXiv preprint arXiv:1705.06211*, 2017.

[3] P. Birken. Termination criteria for inexact fixed-point schemes. *Numer. Linear Algebra Appl.*, 22(4):702–716, 2015.

[4] R. Bollapragada, R. Byrd, and J. Nocedal. Exact and inexact subsampled Newton methods for optimization. *arXiv preprint arXiv:1609.08502*, 2016.

[5] C.L. Byrne. *Applied iterative methods*. AK Peters Wellesley, 2008.

[6] A. Cassioli, D. Di Lorenzo, and M. Sciandrone. On the convergence of inexact block coordinate descent methods for constrained optimization. *European Journal of Operational Research*, 231(2):274–281, 2013.

[7] A. Chambolle, M.J. Ehrhardt, P. Richtárik, and C.B. Schönlieb. Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *SIAM J. Optim.*, 28(4):2783–2808, 2018.

[8] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[9] D. Csiba and P. Richtárik. Global convergence of arbitrary-block gradient methods for generalized Polyak-Lojasiewicz functions. *arXiv preprint arXiv:1709.03014*, 2017.

[10] R.S. Dembo, S.C. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM J. Numer. Anal.*, 19(2):400–408, 1982.

[11] O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1-2):37–75, 2014.

[12] 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). *arXiv preprint arXiv:1707.08486*, 2017.

[13] Y.C. Eldar and D. Needell. Acceleration of randomized Kaczmarz method via the Johnson–Lindenstrauss lemma. *Numerical Algorithms*, 58(2):163–177, 2011.

[14] O. Fercoq and P. Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM J. Optim.*, 25(4):1997–2023, 2015.

[15] K. Fountoulakis and R. Tappenden. A flexible coordinate descent method. *Computational Optimization and Applications*, 70(2):351–394, 2018.

[16] M.P. Friedlander and M. Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM J. Sci. Comput.*, 34(3):A1380–A1405, 2012.

[17] G.H. Golub and C.F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[18] R.M. Gower and P. Richtárik. Randomized iterative methods for linear systems. *SIAM. J. Matrix Anal. & Appl.*, 36(4):1660–1690, 2015.

[19] R.M. Gower and P. Richtárik. Stochastic dual ascent for solving linear systems. *arXiv preprint arXiv:1512.06890*, 2015.

[20] F. Hanzely, J. Konečný, N. Loizou, P. Richtárik, and D. Grishchenko. Privacy preserving randomized gossip algorithms. *arXiv preprint arXiv:1706.07636*, 2017.

[21] F. Hanzely, J. Konečnỳ, N. Loizou, P. Richtárik, and D. Grishchenko. A privacy preserving randomized gossip algorithm via controlled noise insertion. *NeurIPS Privacy Preserving Machine Learning Workshop*, 2018.

[22] B. Hu, P. Seiler, and L. Lessard. Analysis of approximate stochastic gradient using quadratic constraints and sequential semidefinite programs. *arXiv preprint arXiv:1711.00987*, 2017.

[23] S. Kaczmarz. Angenäherte auflösung von systemen linearer gleichungen. *Bulletin International de lAcademie Polonaise des Sciences et des Lettres*, 35:355–357, 1937.

[24] Y.T. Lee and A. Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 147–156. IEEE, 2013.

[25] D. Leventhal and A.S. Lewis. Randomized methods for linear constraints: convergence rates and conditioning. *Mathematics of Operations Research*, 35(3):641–654, 2010.

[26] N. Loizou, M. Rabbat, and P. Richtárik. Provably accelerated randomized gossip algorithms. *arXiv preprint arXiv:1810.13084*, 2018.

[27] N. Loizou and P. Richtárik. A new perspective on randomized gossip algorithms. In *4th IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2016.

[28] N. Loizou and P. Richtárik. Linearly convergent stochastic heavy ball method for minimizing generalization error. *NIPS-Workshop on Optimization for Machine Learning [arXiv preprint arXiv:1710.10737]*, 2017.

[29] N. Loizou and P. Richtárik. Momentum and stochastic momentum for stochastic gradient, Newton, proximal point and subspace descent methods. *arXiv preprint arXiv:1712.09677*, 2017.

[30] N. Loizou and P. Richtárik. Accelerated gossip via stochastic heavy ball method. *56th Annual Allerton Conference on Communication, Control, and Computing*, 2018.

[31] A. Ma, D. Needell, and A. Ramdas. Convergence properties of the randomized extended Gauss-Seidel and Kaczmarz methods. *SIAM. J. Matrix Anal. & Appl.*, 36(4):1590–1604, 2015.

[32] I. Necoara and V. Nedelcu. Rate analysis of inexact dual first-order methods application to dual decomposition. *IEEE Transactions on Automatic Control*, 59(5):1232–1243, 2014.

[33] D. Needell. Randomized Kaczmarz solver for noisy linear systems. *BIT Numerical Mathematics*, 50(2):395–403, 2010.

[34] D. Needell and J.A. Tropp. Paved with good intentions: analysis of a randomized block Kaczmarz method. *Linear Algebra Appl.*, 441:199–221, 2014.

[35] D. Needell, R. Zhao, and A. Zouzias. Randomized block Kaczmarz method with projection for solving least squares. *Linear Algebra Appl.*, 484:322–343, 2015.

[36] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM J. Optim.*, 22(2):341–362, 2012.

[37] 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*, pages 547–556. AUAI Press, 2016.

[38] C. Popa. Least-squares solution of overdetermined inconsistent linear systems using Kaczmarz's relaxation. *International Journal of Computer Mathematics*, 55(1-2):79–89, 1995.

[39] C. Popa. Convergence rates for Kaczmarz-type algorithms. *arXiv preprint arXiv:1701.08002*, 2017.

[40] Z. Qu and P. Richtárik. Coordinate descent with arbitrary sampling i: Algorithms and complexity. *Optimization Methods and Software*, 31(5):829–857, 2016.

[41] Z. Qu and P. Richtárik. Coordinate descent with arbitrary sampling ii: Expected separable overapproximation. *Optimization Methods and Software*, 31(5):858–884, 2016.

[42] Z. Qu, P. Richtárik, M. Takáč, and O. Fercoq. SDNA: Stochastic dual Newton ascent for empirical risk minimization. *ICML*, 2016.

[43] Z. Qu, P. Richtárik, and T. Zhang. Quartz: Randomized dual coordinate ascent with arbitrary sampling. In *Advances in Neural Information Processing Systems*, pages 865–873, 2015.

[44] P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

[45] P. Richtárik and M. Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1-2):433–484, 2016.

[46] P. Richtárik and M. Takáč. Stochastic reformulations of linear systems: algorithms and convergence theory. *arXiv:1706.01108*, 2017.

[47] S. Salzo and S. Villa. Inexact and accelerated proximal point algorithms. *Journal of Convex Analysis*, 19(4):1167–1192, 2012.

[48] M. Schmidt, D. Kim, and S. Sra. Projected Newton-type methods in machine learning. *Optimization for Machine Learning*, page 305, 2011.

[49] M. Schmidt, N.L. Roux, and F.R. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in Neural Information Processing Systems*, pages 1458–1466, 2011.

[50] F. Schöpfer and D.A. Lorenz. Linear convergence of the randomized sparse Kaczmarz method. *arXiv preprint arXiv:1610.02889*, 2016.

[51] Anthony Man-Cho So and Z. Zhou. Non-asymptotic convergence analysis of inexact gradient methods for machine learning without strong convexity. *Optimization Methods and Software*, 32(4):963–992, 2017.

[52] M.V. Solodov and B.F. Svaiter. A unified framework for some inexact proximal point algorithms. *Numer. Func. Anal. Opt.*, 22(7-8):1013–1035, 2001.

[53] R. Tappenden, P. Richtárik, and J. Gondzio. Inexact coordinate descent: complexity and preconditioning. *Journal of Optimization Theory and Applications*, 170(1):144–176, 2016.

[54] S. Tu, S. Venkataraman, A.C. Wilson, A. Gittens, M.I. Jordan, and B. Recht. Breaking locality accelerates block Gauss-Seidel. In *ICML*, 2017.

[55] S. Wright and J. Nocedal. Numerical optimization. *Springer Science*, 35(67-68):7, 1999.

[56] P. Xu, F. Roosta-Khorasani, and M.W. Mahoney. Newton-type methods for non-convex optimization under inexact hessian information. *arXiv preprint arXiv:1708.07164*, 2017.

[57] P. Xu, . Yang, J, F. Roosta-Khorasani, C. Ré, and M.W. Mahoney. Sub-sampled Newton methods with non-uniform sampling. In *Advances in Neural Information Processing Systems*, pages 3000–3008, 2016.

[58] Zhewei Yao, Peng Xu, Farbod Roosta-Khorasani, and Michael W Mahoney. Inexact non-convex Newton-type methods. *arXiv preprint arXiv:1802.06925*, 2018.

[59] A. Zouzias and N.M. Freris. Randomized extended Kaczmarz for solving least squares. *SIAM. J. Matrix Anal. & Appl.*, 34(2):773–793, 2013.

A Technical Preliminaries

Lemma 10. *(Lemma 4.2 [46]: 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 $x_* = \Pi_{\mathcal{L},\mathbf{B}}(x_0)$ we have*

$$\frac{\lambda_{\min}^+}{2}\|x - x_*\|_{\mathbf{B}}^2 \leq f(x). \tag{46}$$

Lemma 11. *[46] Let $x_* \in \mathcal{L}$ and $\{x_k\}_{k\geq 0}$ be the random iterates produced by the exact Basic method (Algorithm 1 with $\epsilon_k = 0$) with an arbitrary stepsize $\omega \in \mathbb{R}$. Then:*

$$\|x_{k+1} - x_*\|_{\mathbf{B}}^2 = \|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k - x_*)\|_{\mathbf{B}}^2 = \|x_k - x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f_{\mathbf{S}_k}(x). \tag{47}$$

By taking 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:

$$\mathbb{E}\left[\|x_{k+1} - x_*\|_{\mathbf{B}}^2 \mid x_k\right] = \|x_k - x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f(x_k) \overset{(46)}{\leq} \left[1 - \omega(2-\omega)\lambda_{\min}^+\right]\|x_k - x_*\|_{\mathbf{B}}^2. \tag{48}$$

Remark 7. *Let x and y be random vectors and let σ 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 it holds that $\mathbb{E}[\|\epsilon_k\|_{\mathbf{B}} \mid x_k, \mathbf{S}_k] \leq \sigma_k$.*

B Proofs of Main Results

In our convergence analysis we use several popular inequalities. Look Table 3 in Appendix C 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

$$\mathbb{E}[\mathbb{E}[\mathbb{E}[X \mid x_k, \mathbf{S}_k] \mid x_k]] = \mathbb{E}[X], \tag{49}$$

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 1

Proof. First we decompose:

$$\begin{aligned} \|x_{k+1}-x_*\|_{\mathbf{B}}^2 &= \|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)+\epsilon_k\|_{\mathbf{B}}^2 \\ &= \|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}}^2 + \|\epsilon_k\|_{\mathbf{B}}^2 + 2\left\langle(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*),\epsilon_k\right\rangle. \end{aligned} \tag{50}$$

Applying the innermost expectation of (49) to (50), we get:

$$\begin{aligned} \mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] &= \underbrace{\mathbb{E}[\|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k]}_{T1} + \underbrace{\mathbb{E}[\|\epsilon_k\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k]}_{T2} \\ &\quad + 2\underbrace{\mathbb{E}[\left\langle(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*),\epsilon_k\right\rangle_{\mathbf{B}} \mid x_k,\mathbf{S}_k]}_{T3}. \end{aligned} \tag{51}$$

We now analyze the three expression T1,T2,T3 separately.

Note that an upper bound for the expression T2 can be directly obtained from the assumption

$$T2 = \mathbb{E}[\|\epsilon_k\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] \le \sigma_k^2. \tag{52}$$

The first expression can be written as:

$$\begin{aligned} T1 = \mathbb{E}[\|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] &= \|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}}^2 \\ &\overset{(47)}{=} \|x_k-x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f_{\mathbf{S}_k}(x_k). \end{aligned} \tag{53}$$

For expression T3:

$$\begin{aligned} \mathbb{E}[\left\langle(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*),\epsilon_k\right\rangle_{\mathbf{B}} \mid x_k,\mathbf{S}_k] &= \left\langle(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*),\mathbb{E}[\epsilon_k \mid x_k,\mathbf{S}_k]\right\rangle_{\mathbf{B}} \\ &\overset{\text{C.S.}}{\le} \|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}}\|\mathbb{E}[\epsilon_k \mid x_k,\mathbf{S}_k]\|_{\mathbf{B}} \\ &\overset{\text{Cond.Jensen}}{\le} \|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}}\mathbb{E}[\|\epsilon_k\|_{\mathbf{B}} \mid x_k,\mathbf{S}_k] \\ &\overset{\text{Remark 7 and (24)}}{\le} \|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}}\sigma_k. \end{aligned} \tag{54}$$

By substituting the bounds (52), (53), and (54) into (51) we obtain:

$$\begin{aligned} \mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] &\le \|x_k-x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f_{\mathbf{S}_k}(x_k) + \sigma_k^2 \\ &\quad + 2\|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}}\sigma_k. \end{aligned} \tag{55}$$

We now take the middle expectation (see (49)) and apply it to inequality (55):

$$\begin{aligned} \mathbb{E}[\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] \mid x_k] &\le \|x_k-x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f(x_k) + \sigma_k^2 \\ &\quad + 2\mathbb{E}[\|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}} \mid x_k]\sigma_k. \end{aligned} \tag{56}$$

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 (48) and (47) 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 7 in the last inequality we obtain:

$$\mathbb{E}\left[\|(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x_k-x_*)\|_{\mathbf{B}} \mid x_k\right] \quad = \quad \sqrt{\rho}\|x_k-x_*\|_{\mathbf{B}}. \tag{57}$$

By substituting (57) in (56):

$$\begin{aligned}\mathbb{E}[\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] \mid x_k] &\leq \|x_k-x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f(x_k)+\sigma_k^2 \\ &\quad +2\sigma_k\sqrt{\rho}\|x_k-x_*\|_{\mathbf{B}} \\ &\overset{(48)}{\leq} \rho\|x_k-x_*\|_{\mathbf{B}}^2+\sigma_k^2+2\sigma_k\sqrt{\rho}\|x_k-x_*\|_{\mathbf{B}} \end{aligned} \tag{58}$$

We take the final expectation (outermost expectation in the tower rule (49)) on the above expression to find:

$$\begin{aligned}\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2] &= \mathbb{E}[\mathbb{E}[\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] \mid x_k]] \\ &\leq \rho\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}^2]+\sigma_k^2+2\sigma_k\sqrt{\rho}\ \mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}] \\ &\overset{V.I}{\leq} \rho\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}^2]+\sigma_k^2+2\sigma_k\sqrt{\rho}\sqrt{\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}^2]} \end{aligned} \tag{59}$$

Using $r_k=\mathbb{E}\left[\|x_k-x_*\|_{\mathbf{B}}^2\right]$ equation (59) takes the form:

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

If we further substitute $p_k=\sqrt{r_k}$ and $\ell=\sqrt{\rho}$ the recurrence simplifies to:

$$p_{k+1} \leq \quad \ell p_k+\sigma_k$$

By unrolling the final inequality:

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

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 in the last expression. □

B.2 Proof of Corollary 1

By denoting $r_k=\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}]$ in (28) 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 2

In order to prove Theorem 2 we need to follow similar steps to the proof of Theorem 1. The main differences of the two proofs appear at the points that 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 1*b*. Thus, it is sufficient to focus at the parts of the proof that these bound is used.

Similar to the proof of Theorem 1 we first decompose to obtain the equation (51). There, the expression T1 can be upper bounded from (53) but now using the Assumption 1*b* the expression T2 and T3 can be upper bounded as follows:

$$T2 = \mathbb{E}[\|\epsilon_k\|^2_{\mathbf{B}} \mid x_k, \mathbf{S}_k] \leq q^2 \|x_k - x_*\|^2_{\mathbf{B}}. \tag{60}$$

$$T3 = \mathbb{E}[\left\langle (\mathbf{I} - \omega \mathbf{B}^{-1}\mathbf{Z}_k)(x_k - x_*), \epsilon_k \right\rangle_{\mathbf{B}} \mid x_k, \mathbf{S}_k] \overset{\text{Remark 7 and (54)}}{\leq} \|(\mathbf{I} - \omega \mathbf{B}^{-1}\mathbf{Z}_k)(x_k - x_*)\|_{\mathbf{B}} q \|x_k - x_*\| \tag{61}$$

As a result by substituting the bounds (53), (60), and (61) into (51) we obtain:

$$\begin{aligned} \mathbb{E}[\|x_{k+1} - x_*\|^2_{\mathbf{B}} \mid x_k, \mathbf{S}_k] \overset{(51)}{\leq} & \|x_k - x_*\|^2_{\mathbf{B}} - 2\omega(2-\omega) f_{\mathbf{S}_k}(x_k) + q^2 \|x_k - x_*\|^2_{\mathbf{B}} \\ & + 2\|(\mathbf{I} - \omega \mathbf{B}^{-1}\mathbf{Z}_k)(x_k - x_*)\|_{\mathbf{B}} q \|x_k - x_*\|_{\mathbf{B}}. \end{aligned} \tag{62}$$

By following the same steps to the proof of Theorem 1 the equation (58) takes the form:

$$\begin{aligned} \mathbb{E}[\mathbb{E}[\|x_{k+1} - x_*\|^2_{\mathbf{B}} \mid x_k, \mathbf{S}_k] \mid x_k] &\leq \rho \|x_k - x_*\|^2_{\mathbf{B}} + q^2 \|x_k - x_*\|^2_{\mathbf{B}} + 2q \|x_k - x_*\|_{\mathbf{B}} \sqrt{\rho} \|x_k - x_*\|_{\mathbf{B}} \\ &= \left(\rho + 2q\sqrt{\rho} + q^2\right) \|x_k - x_*\|^2_{\mathbf{B}}. \\ &= \left(\sqrt{\rho} + q\right)^2 \|x_k - x_*\|^2_{\mathbf{B}} \end{aligned} \tag{63}$$

We take the final expectation (outermost expectation in the tower rule (49)) on the above expression to find:

$$\begin{aligned} \mathbb{E}[\|x_{k+1} - x_*\|^2_{\mathbf{B}}] &= \mathbb{E}[\mathbb{E}[\mathbb{E}[\|x_{k+1} - x_*\|^2_{\mathbf{B}} \mid x_k, \mathbf{S}_k] \mid x_k]] \\ &\leq \left(\sqrt{\rho} + q\right)^2 \mathbb{E}[\|x_k - x_*\|^2_{\mathbf{B}}]. \end{aligned} \tag{64}$$

The final result follows by unrolling the recurrence.

B.4 Proof of Theorem 3

Proof. Similar to the previous two proofs by decomposing the update rule and using the innermost expectation of (49) we obtain equation (51). An upper bound of expression T1 is again given by inequality (53). For the expression T2 depending the assumption that we have on the norm of the inexactness error different upper bounds can be used. In particular,

(i) If Assumption 1 holds then: $T2 = \mathbb{E}[\|\epsilon_k\|^2_{\mathbf{B}} \mid x_k, \mathbf{S}_k] \leq \sigma_k^2$.

(ii) If Assumption 1*b* holds then: $T2 = \mathbb{E}[\|\epsilon_k\|^2_{\mathbf{B}} \mid x_k, \mathbf{S}_k] \leq \sigma_k^2 = q^2 \|x_k - x_*\|^2_{\mathbf{B}}$.

(iii) If Assumption 1*c* holds then: $T2 = \mathbb{E}[\|\epsilon_k\|^2_{\mathbf{B}} \mid x_k, \mathbf{S}_k] \leq \sigma_k^2 = 2q^2 f_{S_k}(x_k)$.

The main difference from the previous proofs, is that due to the Assumption 2 and tower property (49) the expression T3 will eventually be equal to zero. More specifically, we have that:

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

Thus, in this case equation (55) takes the form:

$$\mathbb{E}[\|x_{k+1} - x_*\|^2_{\mathbf{B}} \mid x_k, \mathbf{S}_k] \leq \|x_k - x_*\|^2_{\mathbf{B}} - 2\omega(2-\omega) f_{\mathbf{S}_k}(x_k) + \sigma_k^2. \tag{65}$$

Using the above expression depending the assumption that we have we obtain the following results:

(i) By taking the middle expectation (see (49)) and apply it to the above inequality:

$$\begin{aligned} \mathbb{E}[\mathbb{E}[\|x_{k+1} - x_*\|^2_{\mathbf{B}} \mid x_k, \mathbf{S}_k] \mid x_k] &\leq \|x_k - x_*\|^2_{\mathbf{B}} - 2\omega(2-\omega) f(x_k) + \mathbb{E}[\sigma_k^2 \mid x_k] \\ &\overset{(48)}{\leq} \rho \|x_k - x_*\|^2_{\mathbf{B}} + \mathbb{E}[\sigma_k^2 \mid x_k] \end{aligned} \tag{66}$$

We take the final expectation (outermost expectation in the tower rule (49)) on the above expression to find:

$$\begin{aligned}\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2] &= \mathbb{E}[\mathbb{E}[\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] \mid x_k]] \\ &\leq \rho\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}^2] + \mathbb{E}[\mathbb{E}[\sigma_k^2 \mid x_k]] \\ &= \rho\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}^2] + \mathbb{E}[\sigma_k^2] \\ &= \rho\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}^2] + \bar{\sigma}_k^2 \end{aligned} \tag{67}$$

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 the case (ii) inequality (65) takes the form:

$$\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] \leq \|x_k-x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f_{\mathbf{S}_k}(x_k) + q^2\|x_k-x_*\|_{\mathbf{B}}^2, \tag{68}$$

and by taking the middle expectation (see (49)) we obtain:

$$\begin{aligned}\mathbb{E}[\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] \mid x_k] &\leq \|x_k-x_*\|_{\mathbf{B}}^2 - 2\omega(2-\omega)f(x_k) + q^2\|x_k-x_*\|_{\mathbf{B}}^2 \\ &\overset{(48)}{\leq} \rho\|x_k-x_*\|_{\mathbf{B}}^2 + q^2\|x_k-x_*\|_{\mathbf{B}}^2 \\ &= (\rho+q^2)\|x_k-x_*\|_{\mathbf{B}}^2. \end{aligned} \tag{69}$$

By taking the final expectation of the tower rule (49) and apply it to the above inequality:

$$\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2] \leq (\rho+q^2)\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}^2]. \tag{70}$$

and the result is obtain by unrolling the last expression.

(iii) For the case (iii) inequality (65) takes the form:

$$\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] \leq \|x_k-x_*\|_{\mathbf{B}}^2 - 2(\omega(2-\omega)-q^2)f_{\mathbf{S}_k}(x_k), \tag{71}$$

and by taking the middle expectation (see (49)) we obtain:

$$\begin{aligned}\mathbb{E}[\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2 \mid x_k,\mathbf{S}_k] \mid x_k] &\leq \|x_k-x_*\|_{\mathbf{B}}^2 - 2(\omega(2-\omega)-q^2)f(x_k) \\ &\overset{(46)}{\leq} \|x_k-x_*\|_{\mathbf{B}}^2 - (\omega(2-\omega)-q^2)\lambda_{\min}^+\|x_k-x_*\|_{\mathbf{B}}^2 \\ &= (1-(\omega(2-\omega)-q^2)\lambda_{\min}^+)\|x_k-x_*\|_{\mathbf{B}}^2. \end{aligned} \tag{72}$$

By taking the final expectation of the tower rule (49) to the above inequality:

$$\mathbb{E}[\|x_{k+1}-x_*\|_{\mathbf{B}}^2] \leq (1-(\omega(2-\omega)-q^2)\lambda_{\min}^+)\mathbb{E}[\|x_k-x_*\|_{\mathbf{B}}^2]. \tag{73}$$

and the result is obtain by unrolling the last expression.

□

C Useful Inequalities and Frequently Used Notation

Useful inequalities			
Inequalities (Full names)	Abbreviations	Formula	Assumptions
Jensen Inequality	*Jensen*	$f[\mathbb{E}(x)] \leq \mathbb{E}[f(x)]$	f is convex
Conditional Jensen Inequality	*cond. Jensen*	$f(\mathbb{E}[x \mid s]) \leq \mathbb{E}[f(x) \mid s]$	f is convex
Cauchy-Swartz (B-norm)	*C.S.*	$\lvert\langle a, b\rangle_{\mathbf{B}}\rvert \leq \lVert a\rVert_{\mathbf{B}}\lVert b\rVert_{\mathbf{B}}$	$a, b \in \mathbb{R}^n$
Variance Inequality	*V.I.*	$(\mathbb{E}[X])^2 \leq \mathbb{E}[X^2]$	X random variable

Table 3: Popular inequalities with abbreviations and formulas.

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)
$\lVert x\rVert_{\mathbf{B}}$	$\sqrt{\langle x, x\rangle_{\mathbf{B}}}$ ($\mathbf{B}$-norm)
$\mathbf{M}^\dagger$	Moore-Penrose pseudoinverse of matrix $\mathbf{M}$
$\mathbf{S}$	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 $(\mathbf{M})$	range space of matrix $\mathbf{M}$
Null $(\mathbf{M})$	null space of matrix $\mathbf{M}$
$\mathbb{P}(\cdot)$	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 $\left(\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}\right)$
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	
ω	relaxation parameter / stepsize
ϵ_k	Inexactness error
q	Inexactness parameter
ρ	$1 - \omega(2-\omega)\lambda_{\min}^+$

Table 4: Frequently used notation.