Zeroth-Order Stochastic Optimization: Deterministic Perturbations and Non-Asymptotic Bounds

A THESIS

submitted by

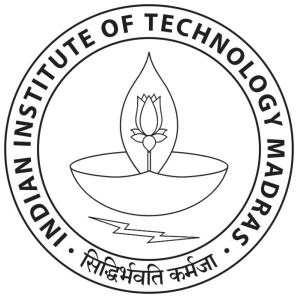
BHAVSAR NIRAV NARHARIBHAI

for the award of the degree

of

MASTER OF SCIENCE

(by Research)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

JULY 2020

THESIS CERTIFICATE

This is to certify that the thesis titled **Zeroth-Order Stochastic Optimization: Deterministic Perturbations and Non-Asymptotic Bounds**, submitted by **Bhavsar Nirav Narharibhai (CS17S016)**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Science (by Research)**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. L. A. Prashanth
Research Guide
Assistant Professor
Dept. of Computer Science and Engineering
IIT-Madras, 600 036

Place: Chennai

Date:

ACKNOWLEDGEMENTS

I want to express my profound gratitude to the people who have helped and supported me in this journey.

First and foremost, I would like to thank my research advisor Dr. L. A. Prashanth, who has supported me throughout my research. The door to my advisor's office was always open, and I am thankful for his constant guidance, which not only helped me sharpen the understanding of the subject but also helped me in improving my writing and presentation skills. Without him, this thesis would not have been completed.

Next, I would like to thank Dr. Shalabh Bhatnagar, Dr. Michael C. Fu, and Dr. Steven Marcus, whose valuable feedback and suggestions improved the quality of this thesis considerably. I would also like to thank my general test committee members: Dr. Madhu Mutyam, Dr. C Chandra Sekhar, and Dr. Puduru Viswanadha Reddy, for their valuable feedback. I extend my gratitude to all the faculties of the Department of Computer Science and Engineering for sharing their knowledge and for their guidance.

Finally, I would like to express special thanks to my friends: Ajay, Rajendra, Sidharth, Pawandeep, and many others who made my stay pleasant and enjoyable. I would also like to thank my family members for their love, support and blessings.

ABSTRACT

KEYWORDS: Zeroth-Order Stochastic Optimization; Stochastic Approximation; Simultaneous Perturbation; Random Directions Stochastic Approximation; Gaussian Smoothing.

Problems of optimization under uncertainty arise in many areas of science and engineering, such as machine learning, communication networks, manufacturing systems, vehicular traffic control, service systems, and several others. Specific applications are varied but include: running simulations to refine the placement of acoustic sensors on a beam, deciding when to switch traffic lights at signal junctions for optimal flow, and optimizing the parameters of a statistical model for a given data set. The usual way to model these problems analytically is by defining an objective or a cost function whose optimum constitutes the desired solution. However, a large number of input variables, randomness (noise) in the input data, and the lack of a system model prohibit a precise analytical solution. A viable alternative is to employ simulation-based optimization.

We consider the following stochastic optimization problem $\min_{x\in\mathbb{R}^d}\{f(x)=\mathbb{E}_\xi[F(x,\xi)]\}$, where the function $f:\mathbb{R}^d\to\mathbb{R}$ is assumed to be smooth, and ξ is the noise factor that captures stochastic nature of the problem. We operate in a *simulation optimization* setting (Fu, 2015), where $F(\cdot,\xi)$ is not given explicitly, but through a black-box simulation procedure. Gradient-based methods are popular for solving such optimization problems. In the simulation-optimization context, gradient information is typically unavailable and has to be estimated from noisy function measurements. This setting is also referred to as the zeroth-order stochastic optimization. Simultaneous perturbation (Bhatnagar *et al.*, 2013; Nesterov and Spokoiny, 2017) refers to a class of algorithms that can provide biased gradient and Hessian information, albeit with a bias that can be controlled, usually at the cost of increased variance in the gradient and Hessian estimate, using noisy function measurements.

We consider two problems in the context of zeroth-order stochastic optimization. In

the first problem, we introduce deterministic perturbation schemes for the recently proposed random directions stochastic approximation (RDSA) method (Prashanth *et al.*, 2017), and propose new first-order and second-order algorithms. In the latter case, these are the first second-order algorithms to incorporate deterministic perturbations. We show that the gradient and/or Hessian estimates in the resulting algorithms with deterministic perturbations are asymptotically unbiased, so that the algorithms are provably convergent. Furthermore, we derive convergence rates to establish the superiority of the first-order and second-order algorithms, for the special case of a convex and a quadratic optimization problem, respectively. Finally, we perform numerical experiments to validate our theoretical results.

In the second problem, we consider the problem of optimizing an objective function with and without convexity in a simulation-optimization context, where only stochastic zeroth-order information is available. We consider two techniques for estimating gradient/Hessian, namely simultaneous perturbation (SP) and Gaussian smoothing (GS). We introduce an optimization oracle to capture a setting where the function measurements have an estimation error that can be controlled. Our oracle is appealing in several practical contexts where the objective has to be estimated from i.i.d. samples, and increasing the number of samples reduces the estimation error. In the stochastic non-convex optimization context, we analyze the zeroth-order variant of the randomized stochastic gradient (RSG) (Ghadimi and Lan, 2013) and quasi-Newton (RSQN) (Wang *et al.*, 2017) algorithms with a biased gradient/Hessian oracle, and with its variant involving an estimation error component. In particular, we provide non-asymptotic bounds on the performance of both algorithms. Our results provide a guideline for choosing the batch size for estimation, so that the overall error bound matches with the one obtained when there is no estimation error. Next, in the stochastic convex optimization setting, we provide non-asymptotic bounds that hold in expectation for the last iterate of a stochastic gradient descent (SGD) algorithm, and our bound for the GS variant of SGD matches the bound for SGD with unbiased gradient information. We perform simulation experiments on synthetic as well as real-world datasets, and the empirical results validate the theoretical findings.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

K-W	Kiefer-Wolfowitz
FDSA	Finite Difference Stochastic Approximation
SPSA	Simultaneous Perturbation Stochastic Approximation
RDSA	Random Directions Stochastic Approximation
SP	Simultaneous Perturbation
GS	Gaussian Smoothing
SF	Smoothed Functional
DP	Deterministic Perturbation
SGD	Stochastic Gradient Descent
RSG	Randomized Stochastic Gradient
RSQN	Randomized Stochastic Quasi-Newton
ODE	Ordinary Differential Equation
MDP	Markov Decision Process
RL	Reinforcement Learning

CHAPTER 1

Introduction

1.1 Motivation and Overview

Optimization problems involving uncertainties are common in many areas of science and engineering, such as machine learning, vehicular traffic control, service systems, communication networks, financial systems, and several others. For instance, in a general traffic signal control setting, a goal could be to dynamically find the optimal order to switch traffic lights at signal junctions and the amount of time that a lane signal should be green when inputs such as the number of vehicles waiting at other lanes are provided. Similarly, in a general communication network, a goal could be to optimally allocate link bandwidth amongst competing traffic flows. The problems themselves may involve system identification, model fitting, optimal control, or performance evaluation based on observed data. A usual way to model these problems analytically is by defining an objective or a cost function whose optimum constitutes the desired solution. However, a large number of input variables, randomness (noise) in the input data, and the lack of a system model prohibit a precise analytical solution, and a viable alternative is to employ simulation-based optimization.

We consider the following stochastic optimization problem $\min_{x\in\mathbb{R}^d}\{f(x)=\mathbb{E}_\xi[F(x,\xi)]\}$, where the function $f:\mathbb{R}^d\to\mathbb{R}$ is assumed to be smooth, and ξ is the noise factor that captures stochastic nature of the problem. We operate in a *simulation optimization* setting (Fu, 2015), where $F(\cdot,\xi)$ is not given explicitly, but through a black-box simulation procedure. The idea here is to simulate the stochastic system under consideration a few times while updating the system parameters until a good enough solution is obtained, and gradient-based methods are popular for solving such optimization problems. In the simulation-optimization context, gradient information is typically unavailable and has to be estimated from noisy function measurements. This setting is also referred to as the zeroth-order stochastic optimization, where an optimization algorithm is provided with noisy

function measurements and has to construct gradient/Hessian estimates from these measurements.

Robbins and Monro (1951) developed an incremental-update algorithm that estimates the zeros of the function f when only its noisy measurements are available. This algorithm has found applications in several engineering domains such as signal processing, manufacturing, communication networks, autonomous systems, vehicular traffic networks, etc., where it is often used to find either (a) the fixed points of a certain function or (b) the optima of a certain objective given noisy function measurements.

The earliest gradient search algorithm in this setting is the Kiefer and Wolfowitz (1952) procedure. This, however, requires $2d$ function measurements when the parameter dimension is d. In (Katkovnik and Kulchitsky, 1972; Rubinstein, 1981), the authors proposed a random search technique that became known as the smoothed functional (SF) algorithm. The key idea here is that the convolution of the objective function gradient with a multivariate Gaussian PDF is seen via an integration-by-parts argument as the convolution of the objective function itself with a scaled multivariate Gaussian. Thus, a single noisy function measurement at a perturbed value of the parameter update, perturbed using a multivariate Gaussian, is sufficient to obtain an estimate of the full gradient. This results in a one-measurement estimator that however has high bias. A balanced two-sided estimator of the gradient (requiring two function measurements) that has significantly lower bias than the one-measurement SF estimator was proposed in (Styblinski and Tang, 1990), see also (Chin, 1997) for comparisons of the one-measurement and two-measurement SF algorithms.

Random directions stochastic approximation (RDSA) (Kushner and Clark, 1978) is another gradient search procedure, in which the perturbation variables are considered to be uniformly distributed over the surface of the unit sphere in $\mathbb{R}^d$. Obtaining these perturbation random variables is, however, computationally expensive, particularly when the dimension d is large. In a landmark paper, (Spall, 1992) introduced the simultaneous perturbation stochastic approximation (SPSA) algorithm, a random search technique that estimates the gradient using random perturbations that are independent, symmetric, zero-mean and satisfying an inverse moment bound. The most commonly used and studied class of perturbations within this category are those that are independent, symmetric, ± 1-valued, Bernoulli random variables. This algorithm

(the standard SPSA, as it is known), requires two function measurements at each update step, and became popular because of its computational simplicity, as well as the convergence and rate guarantees that it provides. In another paper (Spall, 1997), presented a one-measurement counterpart of SPSA. This algorithm, however, does not show good performance, as it suffers from a large bias in its gradient estimates. (Bhatnagar *et al.*, 2003) presented certain deterministic perturbation variants of SPSA. Here two constructions for the perturbation variates were proposed, of which, a construction based on Hadamard matrices is seen to show remarkable improvements in the empirical performance of one-measurement SPSA.

Adaptive Newton-type schemes that estimate the Hessian using noisy objective function measurements, as with the gradient, have also gathered considerable attention over the years. The earliest such scheme, due to (Fabian, 1971), estimated the Hessian using finite-difference estimates and required $\mathcal{O}(d^2)$ samples of the objective function at each update epoch. (Spall, 2000), presented a simultaneous perturbation estimate of the Hessian that was based on four noisy function measurements. Two of these measurements also estimate the gradient. In the case when noisy gradient measurements are directly available, he also presented a Newton scheme requiring three measurements. (Bhatnagar, 2005) presented three additional algorithms that estimate the Hessian as well as the gradient, using simultaneous perturbation estimates. In the process, new gradient and Hessian SPSA estimators were developed. (Bhatnagar and Prashanth, 2015) presented a balanced estimator of the Hessian using three function measurements. This paper also presented two algorithms, one of which estimated the inverse Hessian using a recursive procedure based on the Sherman-Morrison-Woodbury lemma, while the other did not require one to compute or estimate the inverse Hessian at each step. It was shown nonetheless that the asymptotic behaviour of the latter algorithm is analogous to a Newton algorithm that would involve a computation of the inverse Hessian matrix at each update step. (Spall, 2009) presented enhancements to the four-simulation Hessian estimator of (Spall, 2000) using certain weighting and feedback mechanisms. These enhancements are seen to improve the performance of the resulting scheme.

The class of SF algorithms was extended by (Bhatnagar, 2007) to include two Newton-based algorithms governed by standard Gaussian perturbations. As with the gradient estimator, the Hessian estimator was obtained from the idea that if one convolves the Hessian with a multivariate Gaussian density, then from an integration-by-

parts argument applied twice, the same can be viewed as a convolution of the objective function with a scaled multivariate Gaussian. This results in a single-measurement Hessian estimator - the same measurement also estimates the gradient. A two-measurement SF algorithm presented there, involving a balanced (two-measurement) Hessian estimator, is seen to work better in practice - again the same two measurements also estimate the gradient. In (Ghoshdastidar *et al.*, 2014*b*,*a*), gradient and Newton SF algorithms based on the multi-variate q-Gaussian density as the smoothing functional, i.e., the perturbation distribution, have been presented. This gives rise to a class of smoothing densities parameterized by the q-parameter. Densities such as multivariate Normal, Cauchy and Uniform that were known to satisfy the properties required of smoothing functionals (Rubinstein and Shapiro, 1993) in SF algorithms, emerge as special cases of the q-Gaussian density for different values of the parameter q. Thus, these papers have served to significantly extend the class of perturbations that play the role of smoothing densities in SF algorithms.

Finally, the RDSA procedure has recently been revisited in detail by (Prashanth *et al.*, 2017), and novel gradient and Newton algorithms have been devised. Recall that in the original RDSA procedure described in (Kushner and Clark, 1978), the perturbation variates are required to be uniformly distributed over the surface of the unit sphere in $\mathbb{R}^d$, d being the parameter dimension. The approach taken in (Prashanth *et al.*, 2017) involves a uniform distribution over a unit cube as opposed to the surface of the unit sphere. The perturbation (component) random variables are thus allowed to be independent, symmetric, and uniformly distributed over an interval that is symmetric around zero. Another class of perturbations, namely asymmetric Bernoulli, have been investigated and found to work nearly as well as SPSA in both theory and practice. Hessian estimators derived from these perturbations have also been proposed in (Prashanth *et al.*, 2017), and both gradient and Newton algorithms have been investigated in detail. The reader is referred to (Bhatnagar *et al.*, 2013) for a rigorous introduction to the class of simultaneous perturbation methods.

Of particular interest to our work is simultaneous perturbation stochastic approximation (SPSA) and its close cousin random directions stochastic approximation (RDSA) algorithm. SPSA, proposed in (Spall, 1992), has been shown to perform well, both in theory and in practice, using Bernoulli perturbations. RDSA, proposed first in (Kushner and Clark, 1978), uses perturbations drawn randomly on a d-dimensional

unit sphere. A recent enhancement to RDSA, proposed in (Prashanth *et al.*, 2017), uses asymmetric Bernoulli perturbations and has been shown to work nearly as well as SPSA in theory and practice.

1.2 Our Contributions

We consider two problems in the context of zeroth-order stochastic optimization.

- **Deterministic perturbations:** We introduce deterministic perturbation schemes in the random directions stochastic approximation (RDSA) method and propose new first-order and second-order algorithms. We show that the gradient and/or Hessian estimates in the resulting algorithms with deterministic perturbations are asymptotically unbiased, so that the algorithms are provably convergent. Furthermore, we derive convergence rates to establish the superiority of the first-order and second-order algorithms, for the special case of a convex and quadratic optimization problem, respectively. Finally, we perform numerical experiments to validate our theoretical results.

- **Non-asymptotic bounds:** We study gradient-based algorithms for solving zeroth-order stochastic convex and non-convex optimization problems given a biased gradient/Hessian oracle, and with its variant involving an estimation error component. For the case of a convex objective function, we provide non-asymptotic bounds that hold in expectation for the last iterate of a stochastic gradient descent (SGD) algorithm. For a non-convex objective function, we analyze the zeroth-order variant of the randomized stochastic gradient (RSG) and stochastic quasi-Newton (RSQN) algorithm and provide non-asymptotic bounds. In both convex and non-convex setting, we provide a guideline for choosing the batch size for estimation, so that the overall bound matches with the one obtained when there is no estimation error. Finally, we validate our theoretical findings through simulation experiments on synthetic and real-world datasets.

1.3 Organization of the thesis

The rest of the thesis is organised as follows:

- Chapter 2 provides the background material on stochastic optimization, and various methods for estimating gradient and Hessian from noisy function measurements, namely finite difference stochastic approximation (FDSA), simultaneous perturbation stochastic approximation (SPSA), random direction stochastic approximation (RDSA) and Gaussian smoothing (GS).

- Chapter 3 introduces a deterministic perturbation scheme in the RDSA method, and proposes new first-order as well as second-order algorithms. This chapter provides a convergence analysis that includes asymptotic unbiasedness, strong convergence, and convergence rate results. Finally, this chapter presents results from numerical experiments.

- Chapter 4 studies gradient-based algorithms for solving zeroth-order stochastic convex and non-convex optimization problems. This chapter introduces an optimization oracle to capture a setting where the function measurements have an estimation error that can be controlled. For both convex and non-convex objective function, this chapter provides non-asymptotic bounds that hold in expectation. The bounds provide a guideline for choosing the batch size for estimation, so that the overall bound matches with the one obtained when there is no estimation error. Finally, this chapter presents results from simulation experiments on synthetic as well as real-world datasets.

- Chapter 5 concludes the thesis and discusses a few directions for future research.

CHAPTER 2

Background

A general optimization problem has the following form:

$$\text{Find } x^* \text{ that solves } \min_{x \in \mathcal{X}} f(x), \tag{2.1}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is called the objective function, x is a d-dimensional parameter of interest and $\mathcal{X} \subset \mathbb{R}^d$ is the feasible region in which x takes values.

Optimization problems can be classified into two categories as deterministic and stochastic optimization problem. If we have complete information about the objective function f, its derivatives, and the set $\mathcal{X}$ then (2.1) would be a deterministic optimization problem. Furthermore, one can use this information to search the optima deterministically. Unfortunately, many real-world problems do not fall in this class, since the function f cannot be known accurately for a variety of reasons. The first reason is due to a simple measurement error. The second reason is that some data represent information about the future (e.g., product demand or price for a future time period) and simply cannot be known with certainty. Optimization problems involving uncertainties are very common in many areas of science and engineering, such as machine learning, vehicular traffic control, manufacturing systems, service systems, communication networks, financial systems, and several others. Specific applications are varied but include: running simulations to refine the placement of acoustic sensors on a beam, deciding when to switch traffic lights at signal junctions for optimal flow, and optimizing the parameters of a statistical model for a given data set.

2.1 Stochastic Optimization

Optimization under uncertainty or stochastic optimization refers to a collection of methods for minimizing or maximizing an objective function when randomness is present. The randomness may be present as either noise in measurements or Monte Carlo randomness in the search procedure, or both. Random input data arise in many areas

such as real-time estimation and control, problems where there is an experimental (random) error in the measurements, and simulation-based optimization where Monte Carlo simulations are run as estimates of an actual system. In stochastic optimization the uncertainty is incorporated into the model, it presumes that we have little knowledge on the structure of f and moreover f cannot be obtained directly, but we are given sample access, i.e.,

$$f(x) = \mathbb{E}_{\xi}[F(x, \xi)], \tag{2.2}$$

where ξ is the noise factor that captures stochastic nature of the problem, and one is allowed to observe only the $F(x, \xi)$ samples. These kinds of optimization problems are more challenging to solve in comparison to a deterministic optimization problem because we have to find $x^* = \arg\min_{x \in \mathcal{X}} f(x)$, given only noisy function samples. A large number of input variables, randomness (noise) in the input data, and the lack of a system model prohibit a precise analytical solution, and a viable alternative is to employ simulation-based optimization.

Simulation optimization (Fu, 2015) is built on two assumptions: (i) a closed-form expression of the objective function is unavailable; and (ii) a simulator that outputs (noisy) function measurements for any input parameter, is available. The implicit assumption in these problems is that function evaluation is computationally expensive.

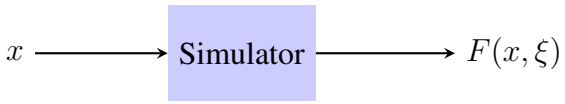


Figure 2.1: Simulation optimization

As illustrated in Figure 2.1, the idea here is to simulate the stochastic system under consideration a few times until a good enough solution is obtained. A standard approach is to devise an iterative algorithm that updates the parameter x_k in the descent direction using the gradient and/or Hessian of the objective function f. Stochastic approximation algorithms are most popular and best suited for solving simulation optimization problems. The first-order stochastic approximation algorithm (SA) takes the following iterative form:

$$x_{k+1} = x_k - \gamma_k \widehat{\nabla} f\left(x_k\right), \tag{2.3}$$

where x_k is the solution found at iteration k, $\widehat{\nabla} f\left(x_k\right)$ is an estimate of the gradient

$\nabla f(x_k)$ and γ_k is a step size (sometimes called the learning rate in machine learning) satisfying the following properties:

$$\sum_{k=1}^{\infty} \gamma_k = \infty, \quad \text{and} \quad \sum_{k=1}^{\infty} \gamma_k^2 < \infty. \tag{2.4}$$

Under appropriate conditions, one can guarantee local convergence to x^* almost surely as $n \to \infty$.

The second-order stochastic approximation algorithm (SA) takes the following iterative form:

$$x_{k+1} = x_k - \gamma_k \left[\Upsilon\left(\bar{H}_k\right)\right]^{-1} \widehat{\nabla} f(x_k), \tag{2.5}$$

where γ_k is the step-size that satisfies (2.4), Υ is an operator that projects a matrix onto the set of positive definite matrices, and $\widehat{\nabla} f(x_k)$ and $\bar{H}_k$ are estimates of the gradient and Hessian, respectively. The first- and second-order SA algorithm can be considered as the stochastic version of the well-known gradient descent and Newton method, respectively.

Second-order methods provide many advantages over their first-order counterparts. The main benefit of second-order methods over first-order methods is that they converge at the optimum rate without requiring knowledge of minimum eigenvalue of $\nabla^2 f(x^*)$ for setting the stepsize. Other benefits include (i) faster convergence in the final phase, i.e., when the iterate is close to the optima as second-order methods minimize a quadratic model of f and (ii) scale-invariance, i.e., second-order methods adjust automatically to the scale of the parameter and hence, the update rule is unaffected. On the flip side, second-order schemes require estimating the Hessian in addition to the gradient of f and have a higher per-iteration cost due to matrix inversion.

In practice, one can only obtain noisy function measurements through black-box simulation, and the challenge is to estimate the gradient and/or Hessian from these measurements. For gradient and Hessian estimation in a zeroth-order optimization setting, we have two important alternatives. The first approach provides an estimate of the objective gradient/Hessian with an additive bias, say of $\mathcal{O}(\eta^2)$, where η is a parameter to be chosen by the optimization algorithm. The variance of the gradient estimate is $\mathcal{O}(1/\eta^2)$, and hence, the choice of η relates to bias-variance tradeoff (Hu *et al.*, 2016). Such an approach can be seen in (Spall, 1992; Bhatnagar *et al.*, 2013; Spall, 2005). We

shall refer to this as the SP approach, as it involves the simultaneous perturbation trick for gradient estimation. The second approach finds an alternative (smooth) function that is not far from the objective, and provides a gradient estimate for this alternative function (cf. (Nesterov and Spokoiny, 2017; Ghadimi and Lan, 2013)). We shall refer to this as the GS approach, as it involves smoothing using a Gaussian distribution. In the following sections, we present a brief survey of existing gradient and/or Hessian estimation schemes.

2.2 Finite Difference Stochastic Approximation (FDSA)

One of the oldest algorithms for estimating gradients using noisy function measurements is the finite difference stochastic approximation (FDSA) by (Kiefer and Wolfowitz, 1952), also known as Kiefer Wolfowitz algorithm. FDSA perturbs the value of each component of x separately while holding the other components at the nominal value.

$$\text{Let} \quad y_{ki}^{+} = f\left(x_k + \eta_k e_i\right) + \xi_{ki}^{+} \quad \text{and} \quad y_{ki}^{-} = f\left(x_k - \eta_k e_i\right) + \xi_{ki}^{-}, \quad \text{for } i = 1, \ldots, d,$$

where the perturbation constant $\eta_k \to 0$ as $k \to \infty$, $\xi_{ki}^{+}, \xi_{ki}^{-}$ are independent and identically distributed (i.i.d.), and e_i is the unit vector with a 1 in the i^{th} place. FDSA based gradient estimate is of the following form:

$$\widehat{\nabla} f\left(x_k\right) = \begin{pmatrix} \frac{y_{k1}^{+} - y_{k1}^{-}}{2\eta_k} \\ \vdots \\ \frac{y_{kd}^{+} - y_{kd}^{-}}{2\eta_k} \end{pmatrix}. \tag{2.6}$$

The convergence of the FDSA based gradient and Hessian estimators are based on the assumption that the noise vector $\left(\xi_{ki}^{+} - \xi_{ki}^{-}, i = 1, 2, \ldots, d\right)^{\top}$ is a martingale difference sequence for every $k \geq 0$, the step-sizes γ_k and perturbation constants η_k are positive for all k, and satisfy

$$\gamma_k, \eta_k \to 0 \text{ as } k \to \infty, \sum_k \gamma_k = \infty \text{ and } \sum_k \left(\frac{\gamma_k}{\eta_k}\right)^2 < \infty. \tag{2.7}$$

Detailed convergence analysis of this algorithm can be seen in (Kiefer and Wolfowitz, 1952). Note that the FDSA based gradient estimation scheme requires $2d$ noisy function measurements, where d is the dimension of the parameter vector x. Further, (Fabian, 1971) presented Hessian estimation scheme using $\mathcal{O}(d^2)$ noisy function measurements. This is the main drawback of this algorithm as it is computationally expensive for high dimensional problems. Therefore, simultaneous perturbation methods such as SPSA and RDSA which uses a constant number of function measurements for estimating gradient and Hessian, irrespective of the parameter dimension, have been analyzed.

2.3 Simultaneous Perturbation Stochastic Approximation (SPSA)

Simultaneous perturbation (SP) refers to a class of algorithms that can provide biased gradient/Hessian information, albeit with a bias that can be controlled, usually at the cost of increased variance in the gradient/Hessian estimate, using noisy function measurements. SP methods are a popular and efficient approach for estimating gradient/Hessian from function samples, especially in high dimensional problems as the number of function measurements needed to form an estimator of the gradient/Hessian is independent of the dimension of the parameter vector. The reader is referred to (Bhatnagar *et al.*, 2013) for a rigorous introduction to the class of simultaneous perturbation methods.

SPSA is a popular SP method. In a landmark paper, (Spall, 1992) introduced the first-order simultaneous perturbation stochastic approximation algorithm, henceforth referred to as 1SPSA. The 1SPSA scheme requires only *two* function measurements to estimate the gradient, regardless of the dimension of the parameter vector. The idea here is to simultaneously perturb all components of the parameter randomly.

Gradient estimate: Let $y_k^+ = f\left(x_k + \eta_k \Delta_k\right) + \xi_k^+$, and $y_k^- = f\left(x_k - \eta_k \Delta_k\right) + \xi_k^-$, where ξ_k^+, ξ_k^- are i.i.d. random vectors in $\mathbb{R}^d$, $\Delta_k = \left(\Delta_k^1, \ldots, \Delta_k^d\right)^\top$ is any vector consisting of i.i.d., zero-mean, symmetric random variables whose inverse second moments are bounded. The most commonly used perturbations within this category are the sym-

metric, ± 1-valued, Bernoulli random variables. SPSA based gradient estimate is of the following form:

$$\widehat{\nabla} f(x_k) = \begin{pmatrix} \frac{y_k^+ - y_k^-}{2\eta_k \Delta_k^1} \\ \vdots \\ \frac{y_k^+ - y_k^-}{2\eta_k \Delta_k^d} \end{pmatrix}. \tag{2.8}$$

Sketch of the proof: By Taylor's series expansions of $f(x_k + \eta_k \Delta_k)$ and $f(x_k - \eta_k \Delta_k)$, we obtain,

$$f(x_k + \eta_k \Delta_k) = f(x_k) + \eta_k \Delta_k^\top \nabla f(x_k) + \frac{\eta_k^2}{2} \Delta_k^\top \nabla^2 f(x_k) \Delta_k + \mathcal{O}\left(\eta_k^3\right),$$
$$f(x_k - \eta_k \Delta_k) = f(x_k) - \eta_k \Delta_k^\top \nabla f(x_k) + \frac{\eta_k^2}{2} \Delta_k^\top \nabla^2 f(x_k) \Delta_k + \mathcal{O}\left(\eta_k^3\right).$$

Combining the above two equations with (2.8), the ith component of the gradient estimation is given by

$$\frac{f(x_k + \eta_k \Delta_k) - f(x_k - \eta_k \Delta_k)}{2\eta_k \Delta_k^i} = \nabla_i f(x_k) + \sum_{j=1, j \neq i}^{d} \frac{\Delta_k^j}{\Delta_k^i} \nabla_j f(x_k) + \mathcal{O}\left(\eta_k^2\right).$$

Then taking conditional expectation, we obtain

$$\mathbb{E}\left[\frac{f(x_k + \eta_k \Delta_k) - f(x_k - \eta_k \Delta_k)}{2\eta_k \Delta_k^i} | \mathcal{F}_k\right] = \nabla_i f(x_k) + \mathcal{O}\left(\eta_k^2\right), \tag{2.9}$$

where we used the fact that Δ_k^j is independent of Δ_k^i when $j \neq i$ and $\mathbb{E}[\Delta_k^j] = 0, \forall j$.

The convergence of the SPSA based gradient estimators are based on the assumption that the noise vector $\xi_k^+ - \xi_k^-, \forall k \geq 0$ is a martingale difference sequence, the step-sizes γ_k and perturbation constants η_k are positive for all k and satisfy (2.7). Detailed convergence analysis of this algorithm can be seen in (Spall, 1992, 2005).

Note that the 1SPSA algorithm requires only *two* function measurements to estimate the gradient, regardless of the dimension of the parameter vector. As a result, 1SPSA became popular because of its computational simplicity, as well as the convergence and rate guarantees that it provides.

Hessian estimate: Let $y_k^+ = f(x_k + \eta_k \Delta_k) + \xi_k^+$, $y_k^- = f(x_k - \eta_k \Delta_k) + \xi_k^-$, $y_k^{++} = f\left(x_k + \eta_k \Delta_k + \widehat{\eta}_k \widehat{\Delta}_k\right) + \xi_k^{++}$, and $y_k^{-+} = f\left(x_k - \eta_k \Delta_k + \widehat{\eta}_k \widehat{\Delta}_k\right) + \xi_k^{+-}$, where the noise terms $\xi_k^+, \xi_k^-, \xi_k^{++}, \xi_k^{-+}$ satisfy $\mathbb{E}\left[\xi_k^{++} - \xi_k^+ - \xi_k^{-+} - \xi_k^- | \mathcal{F}_k\right] = 0$, with

$\mathcal{F}_k = \sigma\left(x_m, m \leq k\right)$ denoting the underlying sigma-field. The perturbation sequence $\left\{\Delta_k^i, \widehat{\Delta}_k^i, i = 1, \ldots, d, k = 1, 2, \ldots\right\}$ is independent. The SPSA based Hessian estimate is of the following form (Spall, 2000):

$$\widehat{H}_k = \left(\Delta_k^{-1}\right)\left(\frac{y_k^{++} - y_k^{+} - y_k^{-+} - y_k^{-}}{2\eta_k\tilde{\eta}_k}\right)\left(\tilde{\Delta_k}^{-1}\right)^T.$$

Note that the number of function measurements required for estimating Hessian is just *four*, regardless of the parameter dimension d. This algorithm is also referred to as the second-order SPSA algorithm (2SPSA). The second-order SPSA algorithm performs an update iteration as follows:

$$x_{k+1} = x_k - \gamma_k \Upsilon\left(\bar{H}_k\right)^{-1} \widehat{\nabla} f\left(x_k\right), \tag{2.10}$$

$$\bar{H}_k = \frac{k}{k+1}\bar{H}_{k-1} + \frac{1}{k+1}\widehat{H}_k. \tag{2.11}$$

In the above, $\bar{H}_k$ is a smoothed version of $\widehat{H}_k$, Υ is an operator that projects a matrix onto the set of positive definite and symmetric matrices, and is crucial to ensure progress along a descent direction. The reader is referred to (Spall, 2000) for detailed convergence results.

2.4 Random Directions Stochastic Approximation (RDSA)

A close cousin of the SPSA is RDSA. The gradient estimate in RDSA differs from SPSA, both in the construction and in the choice of random perturbations. (Kushner and Clark, 1978) proposed gradient search procedure, in which the perturbation variables are considered to be uniformly distributed over the surface of the unit sphere in $\mathbb{R}^d$, where d is the parameter dimension.

The RDSA procedure has recently been revisited in detail by (Prashanth *et al.*, 2017), and two novel gradient and Newton algorithms have been proposed by incorporating random perturbations based on the uniform distribution and a particular asymmetric Bernoulli distribution. RDSA is found to work nearly as well as SPSA in both theory and practice.

We now present the gradient and Hessian estimates using RDSA.

$$\text{Let } y_k^+ = f\left(x_k + \eta_k \Delta_k\right) + \xi_k^+, \quad y_k^- = f\left(x_k - \eta_k \Delta_k\right) + \xi_k^-, \quad \text{and} \quad y_k = f(x_k) + \xi_k, \tag{2.12}$$

where ξ_k^+, ξ_k^-, ξ_k is the measurement noise, η_k is a perturbation constant, and $\Delta_k = \left(\Delta_k^1, \ldots, \Delta_k^d\right)^\top$ are i.i.d. random perturbations.

In (Prashanth *et al.*, 2017), two choices for Δ_k are explored. The first is a uniform distribution, i.e., $\Delta_k^i, \forall i = 1, \ldots, d$, and $\forall k \geq 0$ are i.i.d. zero mean uniform random variables taking values in the interval $\mathbb{U}[-u, u]$ for some $u > 0$. Let y_k, y_k^+ and y_k^- be as defined in (2.12), then, the gradient and Hessian is estimated as follows:

$$\widehat{\nabla} f\left(x_k\right) = \frac{3}{u^2} \Delta_k \left[\frac{y_k^+ - y_k^-}{2\eta_k}\right], \quad \widehat{H}_k = \frac{9}{2u^4} M_k \left(\frac{y_k^+ + y_k^- - 2y_k}{\eta_k^2}\right), \quad \text{where,}$$

$$M_k = \begin{bmatrix} \frac{5}{2}\left(\left(\Delta_k^1\right)^2 - \frac{u^2}{3}\right) & \cdots & \Delta_k^1 \Delta_k^d \\ \Delta_k^2 \Delta_k^1 & \cdots & \Delta_k^2 \Delta_k^d \\ \Delta_k^d \Delta_k^1 & \cdots & \frac{5}{2}\left(\left(\Delta_k^d\right)^2 - \frac{u^2}{3}\right) \end{bmatrix}.$$

We shall refer to the RDSA with uniform perturbations as RDSA-Unif.

The second choice for perturbations is to employ a asymmetric Bernoulli distribution, i.e., $\forall i = 1, \ldots, d$, and $\forall k \geq 0$,

$$\Delta_k^i = \begin{cases} -1 & \text{w.p. } \frac{(1+\epsilon)}{(2+\epsilon)}, \\ 1+\epsilon & \text{w.p. } \frac{1}{(2+\epsilon)}, \end{cases}$$

for some constant $\epsilon > 0$. Let y_k, y_k^+ and y_k^- be as defined in (2.12), then, the gradient and Hessian estimates are formed as follows:

$$\widehat{\nabla} f\left(x_k\right) = \frac{1}{1+\epsilon} \Delta_k \left[\frac{y_k^+ - y_k^-}{2\eta_k}\right], \quad \widehat{H}_k = M_k \left(\frac{y_k^+ + y_k^- - 2y_k}{\eta_k^2}\right), \text{where,}$$

$$M_k = \begin{bmatrix} \frac{1}{\kappa}\left(\left(\Delta_k^1\right)^2 - (1+\epsilon)\right) & \cdots & \frac{1}{2(1+\epsilon)^2} \Delta_k^1 \Delta_k^d \\ \frac{1}{2(1+\epsilon)^2} \Delta_k^2 \Delta_k^1 & \cdots & \frac{1}{2(1+\epsilon)^2} \Delta_k^2 \Delta_k^d \\ \frac{1}{2(1+\epsilon)^2} \Delta_k^d \Delta_k^1 & \cdots & \frac{1}{\kappa}\left(\left(\Delta_k^d\right)^2 - (1+\epsilon)\right) \end{bmatrix}.$$

In the above, $\kappa = \tau\left(1 - \frac{(1+\epsilon)^2}{\tau}\right)$ and $\tau = \frac{(1+\epsilon)\left(1+(1+\epsilon)^3\right)}{(2+\epsilon)}$. We shall refer to the RDSA

with asymmetric Bernoulli perturbations as RDSA-AsymBer.

Similar to SPSA, RDSA variants such as RDSA-Unif and RDSA-AsymBer also requires only *two* function measurements (i.e., y_k^+ and y_k^-) for estimating gradient, while constructing a Hessian estimate would require a *third* function evaluation (i.e., y_k), and the second-order RDSA (2RDSA) algorithm performs an update iteration given by (2.10) and (2.11). Further, the convergence of the RDSA based gradient and Hessian estimators are based on assumptions similar to those in SPSA. The reader is referred to (Prashanth *et al.*, 2017) for detailed convergence results.

In Chapter 3, we propose a variant of RDSA that loops through a deterministic sequence to cancel out the bias in the gradient estimate – a property that regular RDSA achieves in expectation through a zero-mean random perturbation. We propose two new choices for deterministic perturbations, the first choice is based on a semi-lexicographic sequence, while the second employs permutation matrices.

2.5 Gaussian Smoothing (GS)

We consider a smooth approximation of the objective function f. It is well-known (Conn *et al.*, 2009) that the convolution of f with any nonnegative, measurable and bounded function $\psi : \mathbb{R}^d \to \mathbb{R}$ satisfying $\int_{\mathbb{R}^d} \psi(u)du = 1$ is an approximation of f which is at least as smooth as f. Let $\Delta_k \sim \mathcal{N}(0, \mathcal{I}_d)$ be a standard Gaussian random vector. For some $\eta_k > 0$, a smooth approximation of f is defined as:

$$f_{\eta_k}(x_k) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int f(x_k + \eta_k \Delta_k) e^{-\frac{1}{2}\|\Delta_k\|^2} d\Delta_k = \mathbb{E}_{\Delta_k}[f(x_k + \eta_k \Delta_k)].$$

In (Nesterov and Spokoiny, 2017), the authors establish that:

$$\begin{aligned}
\nabla f_{\eta_k}(x_k) &= \mathbb{E}_{\Delta_k}\left[\frac{f(x_k + \eta_k\Delta_k)}{\eta_k}\Delta_k\right] \\
&= \mathbb{E}_{\Delta_k}\left[\frac{f(x_k + \eta_k\Delta_k) - f(x_k)}{\eta_k}\Delta_k\right] \\
&= \frac{1}{(2\pi)^{d/2}} \int \frac{f(x_k + \eta_k\Delta_k) - f(x_k)}{\eta_k} \Delta_k e^{-\frac{\|\Delta_k\|^2}{2}} d\Delta_k.
\end{aligned}$$

This relation implies that we can estimate a gradient of f_{η_k} by only using evaluations of f. The Gaussian smoothing (GS) approach uses Gaussian distribution in the convolution

to find an alternative (smooth) function that is not far from the objective, and then provides a gradient estimate for this alternative function (cf. (Nesterov and Spokoiny, 2017; Ghadimi and Lan, 2013)).

$$\text{Let } y_k^+ = f\left(x_k + \eta_k \Delta_k\right) + \xi_k^+, \quad y_k^- = f\left(x_k - \eta_k \Delta_k\right) + \xi_k^- \quad \text{and} \quad y_k = f(x_k) + \xi_k,$$

where Δ_k is a d-dimensional Gaussian vector composed of standard normal r.v.s., i.e., $\Delta_k \sim N\left(0, I_d\right)$. The GS-based gradient and Hessian estimate is of the following form:

$$\widehat{\nabla} f\left(x_k\right) = \Delta_k \left[\frac{y_k^+ - y_k}{\eta_k}\right] \quad \text{and} \quad \widehat{H}_k = \left[\frac{y_k^+ + y_k^- - 2y_k}{2\eta_k^2}\right] \left(\Delta_k \Delta_k^T - I_d\right).$$

Similar to SPSA and RDSA, GS approach also requires only *two* function measurements for estimating gradient, and *three* function measurements for estimating Hessian. The reader is referred to (Nesterov and Spokoiny, 2017) for detailed convergence results.

CHAPTER 3

Random Directions Stochastic Approximation with Deterministic Perturbations

We introduce deterministic perturbation schemes for the recently proposed RDSA (Prashanth *et al.*, 2017), and propose new first-order and second-order algorithms. In the latter case, these are the first second-order algorithms to incorporate deterministic perturbations. We show that the gradient and/or Hessian estimates in the resulting algorithms with deterministic perturbations are asymptotically unbiased, so that the algorithms are provably convergent. Furthermore, we derive convergence rates to establish the superiority of the first-order and second-order algorithms, for the special case of a convex and quadratic optimization problem, respectively. Numerical experiments are used to validate the theoretical results.

3.1 Introduction

Recall from Chapter 2 that we consider the following problem:

$$\text{Find } x^* = \arg\min_{x \in \mathbb{R}^d} f(x). \tag{3.1}$$

We operate in a setting in which the analytical form of the objective function f is not known, but noisy measurements of the function can be obtained. Furthermore, noisy estimates of the objective function gradient are not directly available, so the function gradient needs to be estimated using the aforementioned noisy measurements. Simultaneous perturbation (Bhatnagar *et al.*, 2013) refers to a class of algorithms that can provide biased gradient information, albeit with a bias that can be controlled, usually at the cost of increased variance in the gradient/Hessian estimate, using noisy function measurements.

In this chapter, we are concerned with developing deterministic perturbation variants of first and second-order RDSA algorithms, henceforth referred to as RDSA-DP

family of algorithms, with 1RDSA-DP (resp. 2RDSA-DP) denoting first (resp. second) order variants. The principal aim is to incorporate deterministic perturbation sequences into RDSA, such that the resulting gradient estimates are still asymptotically unbiased and the overall stochastic gradient algorithm converges, preferably at the same rate as that of the random perturbation RDSA counterparts. We consider two novel choices for deterministic perturbations - a semi-lexicographic sequence and a permutation matrix-based sequence. We combine the two sequences with first and second-order RDSA.

In the case of 1RDSA-DP, the resulting algorithms, under both choices for deterministic perturbations, possess theoretical guarantees that are comparable to those of their random perturbation counterparts. This statement is true when we consider the asymptotic unbiasedness of the gradient estimation and asymptotic convergence of the overall 1RDSA-DP family of algorithms. Moreover, from a non-asymptotic bound that we derive for the special case of strongly-convex objective functions, we observe that the permutation matrix-based perturbations perform best, and even match the rate of a first-order method, whose gradients are directly available.

In the case of second-order RDSA, we incorporate both perturbation sequences to arrive at two variants of 2RDSA-DP, say 2RDSA-Lex-DP and 2RDSA-Perm-DP. However, the theoretical guarantees for the two variants differ significantly. For 2RDSA-Lex-DP, the asymptotic unbiasedness claim holds for the full Hessian, while a similar claim holds only for the Jacobi variant of 2RDSA-Perm-DP involving a diagonal matrix with diagonal elements being those of the Hessian. Furthermore, for the special case of a quadratic optimization problem in the noise-free regime, 2RDSA-Lex-DP is shown to exhibit a convergence rate that is comparable to that of 2SPSA with an adaptive feedback sequence that was proposed in (Spall, 2009). Note that a similar rate result does not exist for regular 2RDSA, and we believe, cannot be established. In any case, to the best of our knowledge, no deterministic perturbation sequences exist for the class of second-order simultaneous perturbation algorithms, including the popular 2SPSA (Spall, 1997).

In comparison to (Bhatnagar *et al.*, 2003), which is the closest related work, we remark that (i) we propose a novel deterministic perturbation scheme and combine it with first-order and second-order RDSA, while the deterministic perturbation schemes in (Bhatnagar *et al.*, 2003) are only for first-order SPSA; (ii) unlike (Bhatnagar *et al.*,

2003), we provide asymptotic normality results that quantify the convergence rate; and (iii) the permutation matrix-based perturbations that we propose are much easier to implement and require much less computational memory in comparison to the deterministic perturbation sequences proposed in (Bhatnagar *et al.*, 2003). In particular, the permutation matrices have a linear dependence on the dimension d, while the lexicographic/Hadamard matrix-based perturbations in (Bhatnagar *et al.*, 2003) scale exponentially with d.

The rest of this chapter is organized as follows: Section 3.2 presents the first-order RDSA variants with two deterministic perturbation sequences, and Section 3.3 describes deterministic perturbation variants of the second-order RDSA algorithm. The main theoretical guarantees for 1RDSA-DP and 2RDSA-DP algorithms are presented in Sections 3.2–3.3, while Section 3.4 provides detailed convergence proofs. Section 3.5 presents simulation experiments that compare the performance of the DP variants of RDSA with several algorithms that employ the simultaneous perturbation technique. Finally, Section 3.6 summarizes the results.

3.2 First-order RDSA with deterministic perturbations

Recall from 2.1 that a first-order method, given the gradient $\nabla f(\cdot)$, would feature an incremental update as follows:

$$x_{k+1} = x_k - \gamma_k \nabla f(x_k). \tag{3.2}$$

In the simulation optimization setting, we are given noisy function measurements, from which the gradient has to be estimated. The simultaneous perturbation method (Bhatnagar *et al.*, 2013) is a popular approach for obtaining such gradients. Recall that the RDSA based gradient estimate is of the following form:

$$\widehat{\nabla} f(x_k) = \frac{1}{1+\epsilon} \Delta_k \left[\frac{y_k^+ - y_k^-}{2\eta_k} \right], \tag{3.3}$$

where $y_k^{\pm} = f(x_k \pm \eta_k \Delta_k) + \xi$, ξ is the measure noise and η_k is a perturbation constant. Further, $\Delta_k = (\Delta_k^1, \ldots, \Delta_k^d)^\intercal$ is the random perturbation vector, with Δ_k^i, $i = 1, \ldots, d$ chosen using the asymmetric Bernoulli distribution, i.e., $\Delta_k^i = -1$ with probability

(w.p.) $\dfrac{(1+\epsilon)}{(2+\epsilon)}$ and $1+\epsilon$ w.p. $\dfrac{1}{(2+\epsilon)}$ for some $\epsilon > 0$.

In this chapter, we propose a variant of 1RDSA that loops through a deterministic sequence to cancel out the bias in the gradient estimate – a property that regular RDSA achieves in expectation through a zero-mean random perturbation. We consider two deterministic constructions for the perturbations Δ_k. The first choice is based on a semi-lexicographic sequence, while the second employs permutation matrices. In both cases, we perform gradient descent similar to (3.2), with a gradient estimate inspired from that of 1RDSA. However, unlike (3.3) that has a random source for perturbations Δ_k, we loop through a deterministic sequence (cf. Tables 3.1 and 3.2 below).

Table 3.1: Illustration of the deterministic perturbation sequence construction for two-dimensional and three-dimensional settings.

(a) Case $d = 2$

Inner loop counter m	D_2^1	D_2^2
0	-1	-1
1	-1	-1
2	-1	2
3	-1	-1
4	-1	-1
5	-1	2
6	2	-1
7	2	-1
8	2	2

(b) Case $d = 3$

Inner loop counter m	D_3^1	D_3^2	D_3^3	Inner loop counter m	D_3^1	D_3^2	D_3^3	Inner loop counter m	D_3^1	D_3^2	D_3^3
0	-1	-1	-1	9	-1	-1	-1	18	2	-1	-1
1	-1	-1	-1	10	-1	-1	-1	19	2	-1	-1
2	-1	-1	2	11	-1	-1	2	20	2	-1	2
3	-1	-1	-1	12	-1	-1	-1	21	2	-1	-1
4	-1	-1	-1	13	-1	-1	-1	22	2	-1	-1
5	-1	-1	2	14	-1	-1	2	23	2	-1	2
6	-1	2	-1	15	-1	2	-1	24	2	2	-1
7	-1	2	-1	16	-1	2	-1	25	2	2	-1
8	-1	2	2	17	-1	2	2	26	2	2	2

3.2.1 Semi-lexicographic sequence-based perturbations

Algorithm 1 presents the pseudocode for the 1RDSA-Lex-DP algorithm that employs a semi-lexicographic sequence for perturbations.

Algorithm 1 1RDSA-Lex-DP

Input: initial parameter $x_0 \in \mathbb{R}^d$, perturbation constants $\eta_k > 0$, step-sizes γ_k, deterministic perturbations $\{\Delta_0, \ldots, \Delta_{3^d-1}\}$.

for $k = 0, 1, 2, \ldots$ **do**

▷ *Fix x_k and loop through the rows of matrix D_d for perturbations Δ_m.*

for $m = 0, 1, 2, \ldots, 3^d - 1$ **do**

Obtain function values $y_m^+ = f(x_k + \eta_{k3^d+m}\Delta_m) + \xi$ and $y_m^- = f(x_k - \eta_{k3^d+m}\Delta_m) + \xi$, where ξ is the measure noise.

Set $g_m = \Delta_m \left[\dfrac{y_m^+ - y_m^-}{2\eta_{k3^d+m}}\right]$.

end for

$$\text{Gradient estimate:} \qquad \widehat{\nabla} f(x_k) = \frac{1}{2 \times 3^d} \sum_{m=0}^{3^d-1} g_m. \tag{3.4}$$

$$\text{Parameter update:} \qquad x_{k+1} = x_k - \gamma_k \widehat{\nabla} f(x_k). \tag{3.5}$$

end for

Return x_k.

Our proposed construction for perturbations Δ_m is illustrated for the case when $d = 2$ and $d = 3$ in Tables 3.1a and 3.1b, respectively. Letting $\mathcal{I}_d$ denote the $d \times d$ identity matrix, for $d = 2$, we have

$$\sum_{m=0}^{3^2-1} \Delta_m \Delta_m^\intercal = \begin{bmatrix} 18 & 0 \\ 0 & 18 \end{bmatrix} \Longrightarrow \frac{1}{2 \times 3^2} \sum_{m=0}^{3^2-1} \Delta_m \Delta_m^\intercal = \mathcal{I}_2.$$

In a similar fashion, for $d = 3$, we have

$$\sum_{m=0}^{3^3-1} \Delta_m \Delta_m^\intercal = \begin{bmatrix} 54 & 0 & 0 \\ 0 & 54 & 0 \\ 0 & 0 & 54 \end{bmatrix} \Longrightarrow \frac{1}{2 \times 3^3} \sum_{m=0}^{3^3-1} \Delta_m \Delta_m^\intercal = \mathcal{I}_3.$$

For any d, we require that $\frac{1}{2\times 3^d} \sum_{m=0}^{3^d-1} \Delta_m \Delta_m^\intercal = \mathcal{I}_d$ to ensure that the gradient estimate $\widehat{\nabla} f(x_k)$ (see (3.4) in Algorithm 1) is asymptotically unbiased. The crucial ingredient in the asymptotic-unbiasedness proof, presented later in Lemma 2, is the following step that uses suitable Taylor's series expansions:

$$\Delta_m \left[\frac{f(x_k + \eta_k \Delta_m) - f(x_k - \eta_k \Delta_m)}{2\eta_k}\right] = \Delta_m \Delta_m^\intercal \nabla f(x_k) + O(\eta_k^2).$$

Hence, if the product $\Delta_m \Delta_m^\intercal$ sums to identity over a loop, then $\widehat{\nabla} f(x_k)$ would be

asymptotically unbiased.

We now present the deterministic perturbation sequence for a general d. Set $D_1^1 = \begin{bmatrix} -1 \\ -1 \\ 2 \end{bmatrix}$ and apply the following recursion $d-1$ times to obtain D_d^i:

$$D_d^i = \begin{bmatrix} D_{d-1}^{i-1} \\ D_{d-1}^{i-1} \\ D_{d-1}^{i-1} \end{bmatrix}, \quad i = 2, \ldots, d. \tag{3.6}$$

The deterministic perturbation sequence loops through the rows in the matrix, say D_d, with columns D_d^i. Notice that each column $D_d^i, i = 1, \ldots, d$ in D_d is of length 3^d. Further, in the first column of D_d, the first $2 \times 3^{d-1}$ elements are -1 and the remaining 3^{d-1} elements are 2. On the other hand, the columns 2 through d in D_d are obtained from $D_{d-1}^1, \ldots, D_{d-1}^{d-1}$, respectively by concatenating the D_{d-1}^i columns thrice.

3.2.2 Permutation matrix-based perturbations

While the semi-lexicographic sequence-based perturbations result in a gradient estimate that is asymptotically unbiased, the inner loop for the perturbations becomes exponentially longer as a function of the dimension d. This exponential dependence on d is problematic, because the descent in parameter x_k occurs at the end of the inner loop (see Algorithm 1), and hence, a long inner loop would imply slow updates (and slow convergence) to x_k.

In this section, we propose an efficient alternative to the semi-lexicographic deterministic sequence; the approach is based on permutation matrices. A permutation matrix is a matrix whose rows are the rows of an identity matrix in some order. For instance, the permutation matrices in two dimension are

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

In three dimensions, there are 6 permutation matrices. In general, there are $d!$ permutation matrices in dimension d.

In the case of permutation matrix-based deterministic perturbations, the overall algorithm follows the template provided in Algorithm 1, except that the perturbations are generated using a permutation matrix in d-dimensions, the inner loop for m runs from 0 to $d-1$ and the gradient estimate in (3.4) is replaced by

$$\widehat{\nabla} f(x_n) = \sum_{m=0}^{d-1} g_m. \tag{3.7}$$

Table 3.2 illustrates the perturbations Δ_m used in Algorithm 1, for $d = 2$ and $d = 3$. In a nutshell, the sequence shown in Table 3.2 loops through the rows of the identity matrix in some order.

Table 3.2: Illustration of the permutation matrix-based deterministic perturbation sequence construction for two-dimensional and three-dimensional settings.

(a) Case $d = 2$

Inner loop counter m	D_2^1	D_2^2
0	1	0
1	0	1

(b) Case $d = 3$

Inner loop counter m	D_3^1	D_3^2	D_3^3
0	0	1	0
1	0	0	1
2	1	0	0

Remark 1. *The classic Kiefer-Wolfowitz (K-W) algorithm (Kiefer and Wolfowitz, 1952) obtains $2d$ function samples per iteration, corresponding to parameters $x_k \pm \eta_k e_i$, $i = 1, \ldots, d$ and updates the parameter as follows:*

$$x_{k+1}^i = x_k^i - \gamma_k \left(\frac{y_k^{i+} - y_k^{i-}}{2\eta_k} \right),$$

where $y_k^{i\pm} = f(x_k \pm \eta_k e_i)$, $i = 1, \ldots, d$.

The 1RDSA-Perm-DP algorithm that we propose resembles K-W in the sense that the inner loop obtains $2d$ samples before updating the parameter x_k. However, the gradient estimate features a product with the perturbation vector Δ_m and this is unlike K-W, where the individual coordinates are independently updated.

3.2.3 Main results

Let $D_d^1, \ldots, D_d^d$ denote the d columns of the semi-lexicographic perturbation variables. Consider the matrix

$$M_d = \begin{bmatrix} (D_d^1)^\intercal D_d^1 & (D_d^1)^\intercal D_d^2 & \ldots & (D_d^1)^\intercal D_d^d \\ (D_d^2)^\intercal D_d^1 & (D_d^2)^\intercal D_d^2 & \ldots & (D_d^2)^\intercal D_d^d \\ \vdots & \vdots & & \vdots \\ (D_d^d)^\intercal D_d^1 & (D_d^d)^\intercal D_d^2 & \ldots & (D_d^d)^\intercal D_d^d \end{bmatrix}.$$

Lemma 1. *For 1RDSA-Lex-DP $M_d = 2 \times 3^d \mathcal{I}_d$, and for 1RDSA-Perm-DP $M_d = \mathcal{I}_d$.*

Proof. See Section 3.4.1. □

Before providing the convergence claims for 1RDSA-DP with either perturbation choice, we outline the necessary assumptions below.

(A1) $f : \mathbb{R}^d \to \mathbb{R}$ is three-times continuously differentiable[1] with $\left|\nabla^3_{i_1 i_2 i_3} f(x)\right| < \alpha_0 < \infty$, for $i_1, i_2, i_3 = 1, \ldots, d$ and for all $x \in \mathbb{R}^d$.

(A2) $\{\xi_m^+, \xi_m^-, m = 0, \ldots, P, k = 1, 2, \ldots\}$ satisfy $\mathbb{E}\left[\xi_m^+ - \xi_m^- \middle| \mathcal{F}_k\right] = 0$, where $P = 3^d - 1$ for semi-lexicographic 1RDSA-DP and $P = d - 1$ for permutation matrix-based 1RDSA-DP.

(A3) For some $\alpha_0, \alpha_1 > 0$ and for all m, k, $\mathbb{E}\left|\xi_m^\pm\right|^2 \leq \alpha_0$, $\mathbb{E}\left|f(x_k \pm \eta\Delta_m)\right|^2 \leq \alpha_1$ for any $\eta > 0$ and Δ_m, $m = 0, \ldots, P$.

(A4) The step-sizes γ_k and perturbation constants η_k are positive, for all k and satisfy

$$\gamma_k, \eta_k \to 0 \text{ as } k \to \infty, \sum_k \gamma_k = \infty \text{ and } \sum_k \left(\frac{\gamma_k}{\eta_k}\right)^2 < \infty.$$

(A5) $\sup_k \|x_k\| < \infty$ w.p. 1.

The assumptions above are common to the analysis of simultaneous perturbation methods, and can be found, for instance, in the context of 1SPSA (Spall, 1992) – see also (Bhatnagar *et al.*, 2013) for the analysis of other simultaneous perturbation schemes

[1] Here $\nabla^3 f(x) = \dfrac{\partial^3 f(x)}{\partial x^\intercal \partial x^\intercal \partial x^\intercal}$ denotes the third derivative of f at x and $\nabla^3_{i_1 i_2 i_3} f(x)$ denotes the $(i_1 i_2 i_3)$th entry of $\nabla^3 f(x)$, for $i_1, i_2, i_3 = 1, \ldots, d$.

that employ similar assumptions. The first two are necessary to establish asymptotic unbiasedness of the 1RDSA-DP gradient estimate through a Taylor series expansion facilitated by (A1), while ignoring the noise owing to (A2). The third and fourth assumptions are necessary to ignore the effects of noise on the convergence behavior of x_k. The final assumption requiring boundedness of the iterates x_k can be ensured by constraining the iterates x_k to evolve in a certain compact region and projecting them back each time they go out of the region, see (Kushner and Clark, 1978) (Chapter 5). If the projected region contains the optima, then the stochastic gradient algorithms (RDSA or SPSA) would converge to this point, and in the complementary case, the algorithm would get stuck on the boundary of the projection region. In the literature, there also exist approaches to overcome the latter case, by either growing the projection region (Chen *et al.*, 1987), or performing sparse projections (Dalal *et al.*, 2018) (i.e., at time instants that grow exponentially to infinity, while not projecting the iterates at the remaining time instants).

Lemma 2. ***(Asymptotic unbiasedness of 1RDSA-DP gradient estimate)*** *Under (A1)-(A5),*

(i) *for $\widehat{\nabla} f(x_k)$ defined according to* (3.4)*, we have a.s. that*[2]

$$\left| \mathbb{E}\left[\widehat{\nabla}_i f(x_k) \middle| \mathcal{F}_k \right] - \nabla_i f(x_k) \right| = C_0 \eta_{k3^d}^2,$$

for $i = 1, \ldots, d$, where $C_0 = \alpha_0 d^3 3^{d-1}$ and $\mathcal{F}_k = \sigma(x_n, n \leq k)$, $k \geq 1$.

(ii) *for $\widehat{\nabla} f(x_k)$ defined according to* (3.7)*, we have a.s. that*

$$\left| \mathbb{E}\left[\widehat{\nabla}_i f(x_k) \middle| \mathcal{F}_k \right] - \nabla_i f(x_k) \right| = C_0 \eta_{kd}^2, \tag{3.8}$$

for $i = 1, \ldots, d$, where $C_0 = \alpha_0 d^3/6$.

Proof. See Section 3.4.1. □

The advantage of the permutation matrix approach is that the dependence on the dimension d is linear, whereas the semi-lexicographic sequence has an exponential dependence on d.

[2]Here $\widehat{\nabla}_i f(x_k)$ and $\nabla_i f(x_k)$ denote the ith coordinates in the gradient estimate $\widehat{\nabla} f(x_k)$ and true gradient $\nabla f(x_k)$, respectively.

We now have an asymptotic convergence claim for x_k updated according to (3.5); the claim is verbatim from Theorem 2 of (Prashanth *et al.*, 2017).

Theorem 3. ***(Strong Convergence)*** *Let x^* be an asymptotically stable equilibrium of the following ordinary differential equation (ODE): $\dot{x}_t = -\nabla f(x_t)$, with domain of attraction $D(x^*)$, i.e., $D(x^*) = \{x_0 \mid \lim_{t\to\infty} x(t \mid x_0) = x^*\}$, where $x(t \mid x_0)$ is the solution to the ODE with initial condition x_0. Assume (A1)-(A5), and also that there exists a compact subset $\mathcal{D}$ of $D(x^*)$ such that $x_k \in \mathcal{D}$ infinitely often. Let x_k be governed by* (3.5)*, with the gradient estimate $\widehat{\nabla} f(x_k)$ defined either according to* (3.4) *or* (3.7)*. Then,*

$$x_k \to x^* \text{ a.s. as } k \to \infty.$$

Proof. See Section 3.4.1. □

For the special case when the objective f is strongly-convex, we present a non-asymptotic bound for 1RDSA-DP with permutation matrix-based perturbations. More precisely, we assume the objective function f satisfies the following assumption:

(**A1'**) For any x, x', we have

$$(\nabla f(x) - \nabla f(x'))^{\mathsf{T}}(x - x') \geq \mu \left\|x - x'\right\|_2^2,$$

for some $\mu > 0$.

Theorem 4. ***(Non-asymptotic bound)*** *Under (A1') and (A2)-(A5), we have,*

$$\mathbb{E}\left\|x_{k+1} - x^*\right\|_2 \leq \underbrace{\sqrt{2}\exp(-\mu\Gamma_k)\left\|x_0 - x^*\right\|_2}_{\textit{initial error}} + \left(\underbrace{3\sum_{n=1}^{k}\gamma_n^2\exp(-2\mu(\Gamma_k - \Gamma_n))C_0^2\eta_n^4}_{\textit{bias error}} + \underbrace{2\sum_{n=1}^{k}\gamma_n^2\exp(-2\mu(\Gamma_k - \Gamma_n))C_1\eta_n^{-2}}_{\textit{sampling error}}\right)^{\frac{1}{2}}, \tag{3.9}$$

where x^* is the global minimizer of f, $\Gamma_k := \sum_{i=1}^{k}\gamma_i$, C_0 is as defined in Lemma 2, and $C_1 = \alpha_1 d/2$.

Proof. See Section 3.4.1. □

The initial error depends on the starting point x_0 of the algorithm. The sampling error relates to a martingale difference sequence, which arises due to the fact that only noisy measurements of the objective function are available. The bias error arises out of the need to estimate gradients from function measurements and quantifies the error in gradient estimation. The initial and sampling error components are common to classic stochastic convex optimization settings, while the bias error is specific to the simulation optimization framework, i.e., a setting where gradients are not directly available and have to be estimated from noisy function measurements.

Now we specialize the result above by choosing the step-size γ_k and perturbation constant η_k to obtain an order $\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ bound in expectation on the optimization error of the algorithm.

Corollary 5. *Let $\gamma_k = c/k$ and $\eta_k = \eta_0/k^\eta$. Then under (A1'), (A2) and (A3),*

$$\mathbb{E}\left\|x_k - x^*\right\|_2 \leq \frac{\sqrt{2}\left\|x_0 - x^*\right\|_2}{k^{\mu c}} + \frac{\sqrt{3}cC_0\eta_0^2}{\sqrt{2\mu c - 4\eta - 1}}k^{-\frac{(1+4\eta)}{2}} + \frac{\sqrt{3C_1}c}{\eta_0\sqrt{2\mu c + 2\eta - 1}}k^{\eta-\frac{1}{2}}.$$

Proof. See Section 3.4.1. □

Choosing $\eta = 0$, one can recover the optimal rate of the order $\mathcal{O}\left(k^{-1/2}\right)$ for simultaneous perturbation schemes. Further, choosing c such that $\mu c > 1/2$, it is easy to observe that the initial error is forgotten faster than the other error components. In contrast, for the more general case of non-convex objective f, the authors in (Spall, 1992; Chin, 1997) are able to establish a rate of $\mathcal{O}\left(k^{-1/3}\right)$ obtained from an asymptotic mean square error analysis using the second moment of the limiting normal distribution. More recently, for the case of convex (and not necessarily strongly-convex) objective f, an error of the order $\mathcal{O}\left(k^{-1/3}\right)$ is unavoidable from an information-theoretic (or minimax) viewpoint – see (Hu *et al.*, 2016) for further details.

3.3 Second-order RDSA with deterministic perturbations

Recall from 2.1 that second-order methods provide many advantages over their first-order counterparts. Using the two deterministic sequences, i.e., semi-lexicographic and

permutation matrix-based choices, presented in the previous section, we provide two variants of the 2RDSA algorithm proposed in (Prashanth *et al.*, 2017).

3.3.1 Semi-lexicographic sequence-based perturbations

Algorithm 2 2RDSA-Lex-DP

Input: initial parameter $x_0 \in \mathbb{R}^d$, perturbation constants $\eta_k > 0$, step-sizes $\{\gamma_k, \beta_k\}$, matrix projection operator Υ. The deterministic perturbation $\{\Delta_m\}$ sequence is chosen in the same manner as in 1RDSA-DP.

for $k = 0, 1, 2, \ldots$ **do**

Obtain function value $y_k = f(x_k) + \xi$, where ξ is the measure noise.

▷ As in 1RDSA, fix x_k and loop through the rows of matrix D_d for perturbations Δ_m.

for $m = 0, 1, 2, \ldots, 3^d - 1$ **do**

Obtain function values $y_m^+ = f(x_k + \eta_{k3^d+m}\Delta_m) + \xi$ and $y_m^- = f(x_k - \eta_{k3^d+m}\Delta_m) + \xi$, where ξ is the measure noise.

$$\text{Set} \quad g_m = \Delta_m \left[\frac{y_m^+ - y_m^-}{2\eta_{k3^d+m}} \right], \tag{3.10}$$

$$\text{Set} \quad \tilde{H}_m = M_m \left[\frac{y_m^+ + y_m^- - 2y_k}{\eta_{k3^d+m}^2} \right], \text{ where} \tag{3.11}$$

$$M_m = \begin{bmatrix} \kappa\left((\Delta_m^1)^2 - 2 \times 3^d\right) & \cdots & \Delta_m^1\Delta_m^d \\ \Delta_m^2\Delta_m^1 & \cdots & \Delta_m^2\Delta_m^d \\ \vdots & \vdots & \vdots \\ \Delta_m^d\Delta_m^1 & \cdots & \kappa\left((\Delta_m^d)^2 - 2 \times 3^d\right) \end{bmatrix},$$

$$\text{and } \kappa = \left(\frac{1}{2 \times 3^{d-1}} - 1 \right)^{-1}.$$

end for

$$\text{Gradient estimate:} \quad \widehat{\nabla} f(x_k) = \frac{1}{2 \times 3^d} \sum_{m=0}^{3^d-1} g_m. \tag{3.12}$$

$$\text{Hessian estimate:} \quad \widehat{H}_k = \frac{1}{(2 \times 3^d)^2} \sum_{m=0}^{3^d-1} \tilde{H}_m. \tag{3.13}$$

$$\text{Hessian update:} \quad \overline{H}_k = (1 - \beta_k)\overline{H}_{k-1} + \beta_k \widehat{H}_k. \tag{3.14}$$

$$\text{Parameter update:} \quad x_{k+1} = x_k - \gamma_k \Upsilon(\overline{H}_k)^{-1} \widehat{\nabla} f(x_k). \tag{3.15}$$

end for

Return x_k.

Algorithm 2 presents the pseudocode for the 2RDSA-Lex-DP algorithm that employs a semi-lexicographic sequence for perturbations illustrated in Table 3.1. The

reason such deterministic choices for perturbations work in the context of 2RDSA can be seen as follows: Using suitable Taylor's series expansions (see Lemma 6 below), we have

$$\mathbb{E}[\widehat{H}_k \mid \mathcal{F}_k] = \frac{1}{2 \times 3^d} \sum_{m=0}^{3^d-1} \left(M_m \sum_{i=1}^{d} (\Delta_m^i)^2 \nabla_{ii}^2 f(x_k) + 2 \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} \Delta_m^i \Delta_m^j \nabla_{ij}^2 f(x_k) + O(\eta_k^2) \right). \tag{3.16}$$

From Lemma 1, it can be seen that

$$\frac{1}{2 \times 3^d} \sum_{m=0}^{3^d-1} \sum_{i=1}^{d} (\Delta_m^i)^2 = 1, \text{ and } \sum_{m=0}^{3^d-1} \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} \Delta_m^i \Delta_m^j = 0.$$

The above equality can be directly verified for the case when $d = 2$ and $d = 3$ using Table 3.1.

Plugging the fact above into (3.16) followed by a tedious calculation (see Lemma 6 below), we obtain

$$\mathbb{E}[\widehat{H}_k(i,j) \mid \mathcal{F}_k] = \nabla_{ij}^2 f(x_k) + O(\eta_{k3^d}^2).$$

Remark 2. ***(Jacobi variant)*** *If the Hessian is known to be in a diagonal form, i.e., if the requirement is for an algorithm to estimate $\nabla_{ii}^2 f(\cdot)$, then the estimate of Algorithm 2 can be replaced by the following:*

$$\widehat{H}_k = \frac{1}{2 \times 3^d} \sum_{m=0}^{3^d-1} \tilde{H}_m,$$

with the inner-loop Hessian estimate given by

$$\tilde{H}_m = \left[\frac{y_m^+ + y_m^- - 2y_k}{\eta_{k3^d+m}^2} \right].$$

Notice that, unlike Algorithm 2, the scheme above (the so-called Jacobi variant of stochastic Newton algorithms - cf. (Bhatnagar, 2005)) can estimate the diagonal entries of the Hessian, and, more importantly, cannot estimate the off-diagonal entries of the Hessian, as the off-diagonal perturbation terms of interest zero out over the inner

loop, i.e., $\sum_{m=0}^{3^d-1}\sum_{i=1}^{d-1}\sum_{j=i+1}^{d}\Delta_m^i\Delta_m^j = 0$.

3.3.2 Permutation matrix-based perturbations

In the case of permutation matrix-based deterministic perturbations, it is not possible to estimate the off-diagonal entries of the Hessian. This is because $\sum_{m=0}^{d-1}\Delta_m^i\Delta_m^j = 0,\ i \neq j$ for permutation matrix-based perturbations. While a similar property holds for semi-lexicographic perturbations as well, we could add correction factors through the M_m matrix (see Algorithm 2) to produce an estimate for all the entries in the Hessian matrix. A similar correction factor is not feasible for the case of permutation matrix-based perturbations, because each column of a permutation matrix contains only one positive ($= 1$) entry, while the rest are zero. In other words, in (3.16), the second term inside the brackets always sums to zero when the outside summation for m goes up to $d-1$, irrespective of the choice for M_m.

However, using a permutation matrix, it is possible to estimate the diagonal entries of the Hessian. In this case, the overall algorithm follows the template provided in Algorithm 2, except that the perturbations are generated using a permutation matrix in d-dimensions, the inner loop for m would run from 0 to $d-1$, and the gradient/Hessian estimates in Algorithm 2 are replaced by

$$\text{Gradient estimate:}\quad \widehat{\nabla} f(x_k) = \sum_{m=0}^{d-1} g_m, \tag{3.17}$$

$$\text{Hessian estimate:}\quad \widehat{H}_k = \sum_{m=0}^{d-1} \tilde{H}_m, \tag{3.18}$$

where

$$\tilde{H}_m = \left[\frac{y_m^+ + y_m^- - 2y_k}{\eta_{kd+m}^2}\right].$$

Alternative using two permutation matrices

Let D_d and $\hat{D}_d$ be two d-dimensional permutation matrices that are not identical. Let $y_m^+ = f(x_k + \eta_k\Delta_m + \eta_k\hat{\Delta}_m) + \xi$ and $y_m^- = f(x_k - \eta_k\Delta_m - \eta_k\hat{\Delta}_m) + \xi$ be function measurements, where ξ is the measure noise, Δ_m and $\hat{\Delta}_m$ are sourced from D_d and $\hat{D}_d$, respectively. In other words, Δ_m and $\hat{\Delta}_m$ would loop through the rows of D_d and $\hat{D}_d$,

respectively. Consider the following estimate for the Hessian, in place of (3.13):

$$\text{Hessian estimate:} \quad \widehat{H}_k = \sum_{m=0}^{d-1} \tilde{H}_m, \tag{3.19}$$

where

$$\tilde{H}_m = \left[\frac{y_m^+ + y_m^- - 2y_k}{\eta_{kd+m}^2} \right].$$

3.3.3 Main results

The analysis of 2RDSA-DP is under assumptions that match those employed for studying the convergence behavior of regular second-order SPSA and RDSA algorithms (i.e., with random perturbations), and we list them below for the sake of completeness.

(**C1**) The function f is four-times differentiable[3] with $\left|\nabla^4_{i_1 i_2 i_3 i_4} f(x)\right| < \infty$, for $i_1, i_2, i_3, i_4 = 1, \ldots, d$ and for all $x \in \mathbb{R}^d$.

(**C2**) For each k and all x, there exists a $\rho > 0$ not dependent on k and x, such that $(x - x^*)^\intercal \bar{f}_k(x) \geq \rho \left\| x_k - x \right\|$, where $\bar{f}_k(x) = \Upsilon(\overline{H}_k)^{-1} \nabla f(x)$.

(**C3**) $\{\xi_m, \xi_m^+, \xi_m^-, m = 0, \ldots, P, k = 1, 2, \ldots\}$ satisfy $\mathbb{E}\left[\xi_m^+ + \xi_m^- - 2\xi_m \middle| \mathcal{F}_k\right] = 0$, where $P = 3^d - 1$ for semi-lexicographic 2RDSA-DP and $P = d - 1$ for permutation matrix-based 2RDSA-DP.

(**C4**) Same as (A4).

(**C5**) For each $i = 1, \ldots, d$ and any $\rho > 0$, $P(\{\bar{f}_{ki}(x_k) \geq 0 \text{ i.o}\} \cap \{\bar{f}_{ki}(x_k) < 0 \text{ i.o}\} \mid \{|x_{ki} - x_i^*| \geq \rho, \forall k\}) = 0$.

(**C6**) The operator Υ satisfies $\eta_k^2 \Upsilon(H_k)^{-1} \to 0$ a.s. and $\mathbb{E}(\left\|\Upsilon(H_k)^{-1}\right\|^{2+\zeta}) \leq \rho$ for some $\zeta, \rho > 0$.

(**C7**) For any $\varsigma > 0$ and nonempty $S \subseteq \{1, \ldots, d\}$, there exists a $\rho'(\varsigma, S) > \varsigma$ such that

$$\limsup_{k \to \infty} \left| \frac{\sum_{i \notin S} (x - x^*)_i \bar{f}_{ki}(x)}{\sum_{i \in S} (x - x^*)_i \bar{f}_{ki}(x)} \right| < 1 \text{ a.s.}$$

for all $|(x - x^*)_i| < \varsigma$ when $i \notin S$ and $|(x - x^*)_i| \geq \rho'(\varsigma, S)$ when $i \in S$.

[3]Here $\nabla^4 f(x) = \dfrac{\partial^4 f(x)}{\partial x^\intercal \partial x^\intercal \partial x^\intercal \partial x^\intercal}$ denotes the fourth derivative of f at x and $\nabla^4_{i_1 i_2 i_3 i_4} f(x)$ denotes the $(i_1 i_2 i_3 i_4)$th entry of $\nabla^4 f(x)$, for $i_1, i_2, i_3, i_4 = 1, \ldots, d$.

(**C8**) For some $\alpha_0, \alpha_1 > 0$ and for all m, k, $\mathbb{E}|\xi_m|^2 \leq \alpha_0$, $\mathbb{E}|\xi_m^{\pm}|^2 \leq \alpha_0$, $\mathbb{E}|f(x_k)|^2 \leq \alpha_1$ and $\mathbb{E}|f(x_k \pm \eta\Delta_m)|^2 \leq \alpha_1$, for any $\eta > 0$.

(**C9**) $\sum_k \frac{1}{(k+1)^2\eta_k^4} < \infty$.

Comments on assumptions (C1)-(C9): (C1) and (C2) are basic assumptions about the smoothness and steepness of the function f. (C1) holds if f is twice continuously differentiable with a bounded second derivative on $\mathbb{R}^d$ and (C2) ensures the function f has enough curvature. (C3) and (C4) are common martingale-difference noise and step-sizes conditions and can be motivated in a similar manner as in the case of 1RDSA-DP (see Section II-C). (C5) says that if x_k is uniformly bounded away from x^*, then x_k cannot be bouncing around causing the change in signs of the normalized gradient elements an infinite number of times. (C6) can be ensured by having $\Upsilon(A)$ defined as performing an eigen-decomposition of A followed by projecting the eigenvalues to the positive side by adding a large enough scalar. (C7) ensures that, after sufficiently large iterations, each element of $\bar{f}_k(x)$ tends to make a non-negligible contribution to products of the form $(x - x^*)^T \bar{f}_k(x)$ (see C2). (C5) and (C7) are not necessary if the iterates are bounded, i.e., $\sup_k \|x_k\| < \infty$ a.s. Finally, (C8) and (C9) are necessary to ensure convergence of the Hessian recursion, in particular, to invoke a martingale convergence result (see Theorem 7 and the proof of (Spall, 2000, Theorem 2a)). For a more detailed interpretation of the above conditions, the reader is referred to Section III and Appendix B of (Spall, 2000).

The main claim that establishes the asymptotic unbiasedness of the Hessian estimate in the DP variants of 2RDSA that we propose is given below.

Lemma 6. ***(Asymptotic unbiasedness of 2RDSA-DP Hessian estimate)*** *Under (C1)-(C9),*

(i) *for $\widehat{H}_k$ defined according to* (3.13)*, we have a.s. that*[4]*, for $i, j = 1, \ldots, d$,*

$$\left| \mathbb{E}\left[\widehat{H}_k(i,j) \middle| \mathcal{F}_k \right] - \nabla^2_{ij} f(x_k) \right| = O(\eta^2_{k3^d}). \tag{3.20}$$

(ii) *for $\widehat{H}_k$ defined according to* (3.18)*, we have a.s. that*

$$\left| \mathbb{E}\left[\widehat{H}_k(i,i) \middle| \mathcal{F}_k \right] - \nabla^2_{ii} f(x_k) \right| = O(\eta^2_{kd}), \tag{3.21}$$

[4]Here $\widehat{H}_k(i,j)$ and $\nabla^2_{ij} f(\cdot)$ denote the (i,j)th entry in the Hessian estimate $\widehat{H}_k$ and the true Hessian $\nabla^2 f(\cdot)$, respectively.

for $i = 1, \ldots, d$.

Proof. See Section 3.4.2. □

Once we have the asymptotic unbiasedness for the 2RDSA-DP Hessian estimate, the convergence of the Hessian recursion is immediate and is given below for the sake of completeness.

Theorem 7. ***(Strong Convergence)*** *Assume (C1)-(C9). Then* $x_k \to x^*$ *a.s. as* $k \to \infty$, *where* x_k *is given by* (3.15). *For 2RDSA-Lex-DP* $\overline{H}_k \to \nabla^2 f(x^*)$ *a.s. as* $k \to \infty$, *furthermore, if the true Hessian is diagonal, then even for 2RDSA-Perm-DP* $\overline{H}_k \to \nabla^2 f(x^*)$ *a.s. as* $k \to \infty$, *where* $\overline{H}_k$ *is governed by* (3.14).

Proof. See Section 3.4.2. □

We next present a convergence rate result for the special case of a quadratic objective function under the following additional assumptions, which have been used to establish a rate result for a variant of 2SPSA in (Spall, 2009):

(**C10**) f is quadratic and $\nabla^2 f(x^*) > 0$.

(**C11**) The operator Υ is chosen such that $\mathbb{E} \parallel \Upsilon(\overline{H}_k) - \overline{H}_k \parallel^2 = \mathcal{O}(e^{-2b_0 k^{1-r}/(1-r)})$ and $\parallel \Upsilon(H) - H \parallel^2 / (1 + \parallel H \parallel^2)$ is uniformly bounded.

The assumptions (C10) and (C11) are much stronger compared to (C1) and (C6), respectively. In a noise-free setting, after suitable Taylor series expansions, the Hessian estimate of 2SPSA can be written as

$$\widehat{H}_k(2SPSA) = \nabla^2 H(x_k) + \Psi(H(x_k)),$$

where $H(x_k)$ is the Hessian at iterate x_k and $\Psi(H(x_k))$ is a function that involves the Hessian at x_k and random perturbations used in 2SPSA. More importantly, $\Psi(H(x_k))$ is zero-mean. In (Spall, 2009), since the true Hessian $H(x_k)$ is not known in practice, in place of $\widehat{H}_k$ in the Hessian update in (3.14), the author uses the following improved Hessian estimate: $\widehat{H}_k(2SPSA) - \Psi(\overline{H}_k)$. The rationale underlying this replacement is that subtracting the $\Psi(\overline{H}_k)$ term reduces the error in Hessian estimation. In (Reddy *et al.*,

2016), the authors use a similar trick to reduce the Hessian estimation error in regular 2RDSA. We claim such a feedback term is not necessary in the context of 2RDSA-DP and this can be argued as follows: From the proof passage leading to (3.37), it is easy to infer the following after ignoring the noise terms:

$$\widehat{H}_k(i,j) = \nabla^2_{ij} f(x_k) + O(\eta^2_{k3^d}).$$

The principal difference with feedback variants of 2SPSA/2RDSA is that the equation above implies that there are no zero-mean terms involving the perturbations that one needs to worry about in case of 2RDSA-DP.

To substantiate the claim that the feedback terms are not necessary for 2RDSA-DP, we provide a rate result that parallels a corresponding result for 2SPSA with feedback (Spall, 2009). The proof given later is also much simpler than the corresponding version for 2SPSA, due to the fact that there are no extra terms involving perturbations to handle.

Theorem 8. *(**Non-asymptotic bound**) Assume (C9), (C10) and (C11), and also that the setting is noise-free. Let* $b_k = b_0/k^r$, $k = 1, 2, \ldots, n$, *where* $1/2 < r < 1$ *and* $0 < b_0 \leq 1$. *Let* $H^* = \nabla^2 f(x)$ *for any* x, *and* $\Lambda_n = \overline{H}_n - H^*$. *Then, we have*

$$\mathit{trace}[\mathbb{E}(\Lambda_k^{\mathsf{T}} \Lambda_k)] = \mathcal{O}(e^{-2b_0 k^{1-r}/(1-r)}). \tag{3.22}$$

Proof. See Section 3.4.2. □

3.4 Convergence proofs

3.4.1 Proofs for 1RDSA variants with deterministic perturbations

Proof of Lemma 1

Proof.

Case 1: Semi-lexicographic sequence-based perturbations

We prove the claim by induction. Consider first the case of $d = 1$. Now $D_1^1 =$

$\begin{bmatrix} -1 \\ -1 \\ 2 \end{bmatrix}$. Note that $M_1 = (D_1^1)^\intercal D_1^1 = 6$. Thus the result holds for $d = 1$.

Suppose that the claim above holds for $d = k$, i.e., $M_k = 2 \times 3^k \mathcal{I}_k$. Consider the case of $d = k + 1$. We make the following observations:

1. The size of each column $D_{k+1}^i, i = 1, \ldots, k+1$ is 3^{k+1}.

2. The columns $D_{k+1}^2, \ldots, D_{k+1}^{k+1}$ are obtained from $D_k^1, \ldots, D_k^k$, respectively by concatenating the D_k^i columns thrice, i.e.,

$$D_{k+1}^i = \begin{bmatrix} D_k^{i-1} \\ D_k^{i-1} \\ D_k^{i-1} \end{bmatrix}, \quad i = 2, \ldots, k+1. \tag{3.23}$$

3. In the first column, i.e., D_{k+1}^1, the first 2×3^k elements are -1 and the remaining 3^k elements are 2.

4. The off-diagonal terms in M_{k+1} are all zero. This is argued as follows: For $i \neq j$, $i, j \in \{2, \ldots, k+1\}$, we have that $(D_{k+1}^i)^\intercal D_{k+1}^j = 0$. This follows from (3.23) because $(D_{k+1}^i)^\intercal D_{k+1}^j = 3(D_k^{i-1})^\intercal D_k^{j-1} = 0$ (by induction hypothesis).

   For off-diagonal terms of the type $(D_{k+1}^1)^\intercal D_{k+1}^i$, where $i \in \{2, \ldots, k+1\}$, we first re-write D_{k+1}^1 and D_{k+1}^i as follows:

$$D_{k+1}^1 = \begin{bmatrix} \mathbf{-1} \\ \mathbf{-1} \\ \mathbf{2} \end{bmatrix}, \tag{3.24}$$

   where $\mathbf{-1}$ and $\mathbf{2}$ are $k \times 1$ vectors with each entry -1 and 2, respectively. Thus, from (3.23) and the foregoing,

$$(D_{k+1}^1)^\intercal D_{k+1}^i = -\mathbf{1}^\intercal D_k^{i-1} - \mathbf{1}^\intercal D_k^{i-1} + \mathbf{2}^\intercal D_k^{i-1} = 0.$$

5. Finally, the diagonal terms of M_{k+1} are $2 \times 3^{k+1}$. This can be seen as follows: For $i = 1, \ldots, k+1$,

$$(D_{k+1}^i)^\intercal D_{k+1}^i = \frac{2}{3} \times (-1)^2 \times 3^{k+1} + \frac{1}{3} \times (2)^2 \times 3^{k+1}$$

$$= 2 \times 3^{k+1}.$$

In the above, we have used the fact that D^i_{k+1} is a 3^{k+1}-dimensional vector with two-third entries -1 and the remaining one-third entries as 2.

Thus, $M_{k+1} = 2 \times 3^{k+1} \mathcal{I}_{k+1}$ and the claim follows for semi-lexicographic perturbations.

Case 2: Permutation matrix-based perturbations

Let D_d, the columns D^i_d, be the d-dimensional permutation matrix. Recall that 1RDSA-Perm-DP loops through the rows in D_d and the columns D^i_d in D_d are in d-dimensions. It is well-known that $D_d^{-1} = D_d^{\intercal}$ for a permutation matrix D_d, cf. Proposition 2.7.21 of (Goodaire, 2013) for a detailed proof. Hence, the claim that $M_d = \mathcal{I}_d$ follows for the case of permutation matrix based deterministic perturbation sequences. □

Proof of Lemma 2

Proof. We prove the claim for semi-lexicographic perturbations; the claim for the case of permutation matrix-based perturbations follows by an analogous argument.

By Taylor's series expansions, we obtain, a.s.,

$$f(x_k \pm \eta_k \Delta_m) = f(x_k) \pm \eta_k \Delta_m^T \nabla f(x_k) + \frac{\eta_k^2}{2} \Delta_m^T \nabla^2 f(x_k) \Delta_m \pm \frac{\eta_k^3}{6} \nabla^3 f(\tilde{x}_k^+)(\Delta_m \otimes \Delta_m \otimes \Delta_m)$$

where $\otimes$ denotes the Kronecker product and $\tilde{x}_k^+$ (respectively, $\tilde{x}_k^-$) are on the line segment between x_k and $(x_k + \eta_k \Delta_m)$ (respectively, $(x_k - \eta_k \Delta_m)$). Then

$$\begin{aligned}
&\mathbb{E}\left[\widehat{\nabla} f(x_k) \middle| \mathcal{F}_k\right] \\
&= \frac{1}{2 \times 3^d} \sum_{m=0}^{3^d-1} \mathbb{E}\left[g_m \middle| \mathcal{F}_k\right] \\
&= \frac{1}{2 \times 3^d} \sum_{m=0}^{3^d-1} \mathbb{E}\left[\frac{\Delta_m}{2\eta_{(k3^d+m)}} \times \left(f(x_k + \eta_{(k3^d+m)}\Delta_m) - f(x_k - \eta_{(k3^d+m)}\Delta_m)\right) \middle| \mathcal{F}_k\right] \\
&= \frac{1}{2 \times 3^d} \sum_{m=0}^{3^d-1} \left[\Delta_m \Delta_m^{\intercal} \nabla f(x_k) + \frac{\eta^2_{(k3^d+m)}}{12} \Delta_m (\nabla^3 f(\tilde{x}_k^+)\right.
\end{aligned}$$

$$+ \nabla^3 f(\tilde{x}_k^-))(\Delta_m \otimes \Delta_m \otimes \Delta_m)\Big]$$

$$= \nabla f(x_k) + C_0 \eta_{k3^d}^2. \tag{3.25}$$

The first term on the RHS of (3.25) follows from Lemma 1. Now, the lth coordinate of the second term in the RHS of (3.25) can be upper-bounded as follows:

$$\begin{aligned}
&\sum_{m=0}^{3^d-1} \frac{\eta_{(k3^d+m)}^2}{12} \Delta_m (\nabla^3 f(\tilde{x}_k^+) + \nabla^3 f(\tilde{x}_k^-))(\Delta_m \otimes \Delta_m \otimes \Delta_m) \\
&\leq \frac{\alpha_0 \eta_{k3^d}^2}{6} \sum_{m=0}^{3^d-1} \sum_{i_1=1}^{d} \sum_{i_2=1}^{d} \sum_{i_3=1}^{d} (\Delta_m^l \Delta_m^{i_1} \Delta_m^{i_2} \Delta_m^{i_3}) \\
&\leq \frac{\alpha_0 \eta_{k3^d}^2 d^3 (2 \times 3^d)^2}{6}
\end{aligned}$$

The first inequality above follows from (A1), while the second inequality follows from the fact that $\sum_{m=0}^{3^d-1} (\Delta_m^l \Delta_m^{i_1} \Delta_m^{i_2} \Delta_m^{i_3})$ is non-zero only if either $l = i_2$ and $i_1 = i_3$ or vice-versa and in this case, we have

$$\sum_{m=0}^{3^d-1} (\Delta_m^l)^2 (\Delta_m^{i_1})^2 = (2 \times 3^d)^2.$$

The equality above can be easily inferred using induction arguments similar to proof of Lemma 1 and we omit the details. □

Proof of Theorem 3

Proof. Follows in exactly the same manner as the proof of Theorem 2 of (Prashanth *et al.*, 2017), given the asymptotic unbiasedness result in Lemma 2. □

Proof of Theorem 4

Proof. Fixing η_k for the inner loop and using arguments from the proof of Lemma 2, we obtain

$$\widehat{\nabla} f(x_k) = \nabla f(x_k) + C_0 \eta_k^2 + \sum_{m=0}^{d-1} \Delta_m \left(\frac{\xi_m^+ - \xi_m^-}{2\eta_k} \right).$$

Letting $\zeta_k = \sum_{m=0}^{d-1} \Delta_m \left(\frac{\xi_m^+ - \xi_m^-}{2\eta_k} \right)$, the update in (3.5) is equivalent to

$$x_{k+1} = x_k - \gamma_k \left[\nabla f(x_k) + C_0 \eta_k^2 + \zeta_k \right]. \tag{3.26}$$

Since $\nabla f(x^*) = 0$, we have the following from the fundamental theorem of calculus:

$$\left(\int_0^1 \nabla^2 f(x^* + \lambda(x_k - x^*)) d\lambda \right) z_k = \nabla f(x_k).$$

Here $z_k := x_k - x^*$ denotes the optimization error at instant k of 1RDSA-DP. Then, using (3.26), we have the following recursive update form for z_k:

$$\begin{aligned} z_{k+1} =& (I - \gamma_k J_k) z_k + \gamma_k \left(C_0 \eta_k^2 + \zeta_k \right) \\ =& \Pi_k z_0 + \sum_{n=1}^{k} \gamma_n \Pi_k \Pi_n^{-1} (C_0 \eta_n^2 + \zeta_n), \end{aligned} \tag{3.27}$$

where $J_k := \int_0^1 \nabla^2 f(x^* + \lambda(x_k - x^*)) d\lambda$ and $\Pi_k := \prod_{n=1}^{k} (I - \gamma_n J_n)$. A similar unrolling of a general stochastic approximation recursion can be found in (Frikha and Menozzi, 2012). However, our setting involves biased gradient estimates and the non-asymptotic bounds require a careful handling of the perturbation constant η_k, so that the overall convergence rate is of the order $O\left(1/\sqrt{k}\right)$. Moreover, we make all the constants explicit in the final bound.

Now, for the square of the error $\|z_{k+1}\|_2^2$, we use (3.27) and Jensen's inequality to obtain

$$\begin{aligned} &\left(\mathbb{E} \|z_{k+1}\|_2\right)^2 \leq \mathbb{E}(\langle z_k, z_k \rangle) \\ &= \mathbb{E}\Bigg[\|\Pi_k z_0\|_2^2 + \left\| \sum_{n=1}^{k} \gamma_n \Pi_k \Pi_n^{-1} C_0 \eta_n^2 \right\|_2^2 + \left\| \sum_{n=1}^{k} \gamma_n \Pi_k \Pi_n^{-1} \zeta_n \right\|_2^2 \\ &\quad + \left\langle \Pi_k z_0, \sum_{n=1}^{k} \gamma_n \Pi_k \Pi_n^{-1} C_0 \eta_n^2 \right\rangle + \left\langle \Pi_k z_0, \sum_{n=1}^{k} \gamma_n \Pi_k \Pi_n^{-1} \zeta_n \right\rangle \\ &\quad + \left\langle \sum_{n=1}^{k} \gamma_n \Pi_k \Pi_n^{-1} C_0 \eta_n^2, \sum_{n=1}^{k} \gamma_n \Pi_k \Pi_n^{-1} \zeta_n \right\rangle \Bigg] \\ &\quad \leq 2 \|\Pi_k z_0\|_2^2 + 3 \sum_{n=1}^{k} \gamma_n^2 \left\|\Pi_k \Pi_n^{-1}\right\|_2^2 C_0^2 \eta_n^4 + 2 \sum_{n=1}^{k} \gamma_n^2 \left\|\Pi_k \Pi_n^{-1}\right\|_2^2 \mathbb{E} \|\zeta_n\|_2^2. \end{aligned} \tag{3.28}$$

For the last inequality, we have used the fact that ζ_n is zero-mean (see (A2)) in order

to ignore a cross term. For the other two cross terms, we have used Cauchy-Schwarz to conclude $\langle a, b\rangle \leq \max(\|a\|_2^2, \|b\|_2^2)$ and hence, the first and last square terms can appear at most twice, while the second square term can appear at most thrice.

The second moment of the noise factor ζ_k can be bounded as follows:

$$\mathbb{E}\left\|\zeta_k\right\|_2^2 \leq \sum_{m=0}^{d-1} \|\Delta_m\|_2^2 \,\mathbb{E}\left(\frac{\xi_m^+ - \xi_m^-}{2\eta_k}\right)^2 \leq \frac{d\alpha_1}{2\eta_k^2}, \tag{3.29}$$

where we have used the fact that for permutation matrix-based perturbation $\|\Delta_m\|_2 = 1$ and $\mathbb{E}(\xi_m^+ - \xi_m^-)^2 \leq 2\alpha_1$ from assumption (A3).

The term $\left\|\Pi_k \Pi_n^{-1}\right\|_2$ is bounded as follows:

$$\begin{aligned}
\left\|\Pi_k \Pi_n^{-1}\right\|_2 &= \left\| \prod_{j=n+1}^{k} (I - \gamma_j J_j) \right\|_2 \\
&= \prod_{j=n+1}^{k} \left\|(1-\gamma_j\mu)I - \gamma_j(J_j - \mu I)\right\|_2 \\
&\leq \prod_{j=n+1}^{k} \left\|(1-\gamma_j\mu)I\right\|_2 \leq \prod_{j=n+1}^{k} (1-\gamma_j\mu) \\
&\leq \exp\left(-\mu(\Gamma_k - \Gamma_n)\right).
\end{aligned} \tag{3.30}$$

The second inequality above follows by observing that $\|I - \gamma_k J_k\|_2 \leq \exp(-\mu\gamma_k)$, since $\nabla^2 f(x) - \mu I$ is positive semi-definite owing to strong-convexity of f (see (A1')).

The main claim now follows by plugging in the bounds in (3.29) and (3.30) into (3.28). □

Proof of Theorem 5

Proof. We bound each of the error terms on the RHS of (3.9) separately. For bounding the initial error, we use the following inequality:

$$\exp(-\mu\Gamma_k) \leq \exp(-\mu c \ln k) \leq k^{-\mu c}.$$

In arriving at the bound above, we have compared a sum with an integral.

Substituting $\gamma_k = c/k$ and $\eta_k = \eta_0/k^\eta$ into the bias error term in (3.9), we obtain

$$\begin{aligned}\sum_{n=1}^{k} \gamma_n^2 \exp(-2\mu(\Gamma_k - \Gamma_n)) C_0^2 \eta_n^4 &\leq \sum_{n=1}^{k} \frac{c^2}{n^2} k^{-2\mu c} n^{2\mu c} C_0^2 \frac{\eta_0^4}{k^{4\eta}} \\ &\leq c^2 k^{-2\mu c} C_0^2 \eta_0^4 \sum_{n=1}^{k} n^{2\mu c - 4\eta - 2} \\ &\leq \frac{c^2 C_0^2 \eta_0^4}{(2\mu c - 4\eta - 1)} k^{-1-4\eta}.\end{aligned}$$

Along similar lines, the sampling error term in (3.9) can be upper-bounded as follows:

$$\sum_{n=1}^{k} \gamma_n^2 \exp(-2\mu(\Gamma_k - \Gamma_n)) \frac{C_1}{\eta_n^2} \leq \frac{c^2 C_1}{\eta_0^2 (2\mu c - 4\eta - 1)} k^{-1+2\eta}.$$

The claim follows by combining the bounds derived above on each of the error terms on the RHS of (3.9). □

3.4.2 Proofs for 2RDSA variants with deterministic perturbations

Proof of Lemma 6

Proof.

Case 1: Semi-lexicographic sequence-based perturbations

As in the proof of Lemma 4 in (Prashanth *et al.*, 2017), we employ Taylor's series expansions of $f(\cdot)$ at $x_k \pm \eta_k \Delta_k$ to obtain

$$\begin{aligned}&\frac{f(x_k + \eta_{(k3^d+m)}\Delta_m) + f(x_k - \eta_{(k3^d+m)}\Delta_m) - 2f(x_k)}{\eta_{(k3^d+m)}^2} \\ =&\Delta_m^\mathsf{T} \nabla^2 f(x_k) \Delta_m + O(\eta_{(k3^d+m)}^2) \\ =&\sum_{i=1}^{d}\sum_{j=1}^{d} \Delta_m^i \Delta_m^j \nabla_{ij}^2 f(x_k) + O(\eta_{(k3^d+m)}^2) \\ =&\sum_{i=1}^{d} (\Delta_m^i)^2 \nabla_{ii}^2 f(x_k) + 2\sum_{i=1}^{d-1}\sum_{j=i+1}^{d} \Delta_m^i \Delta_m^j \nabla_{ij}^2 f(x_k) + O(\eta_{(k3^d+m)}^2). \qquad (3.31)\end{aligned}$$

Recall that the Hessian estimate is given by

$$\widehat{H_k} = \frac{1}{(2 \times 3^d)^2} \sum_{m=0}^{3^d-1} M_m \left(\frac{y_m^+ + y_m^- - 2y_k}{\eta_k^2} \right), \text{ where}$$

M_m is as defined in Algorithm 2. For the sake of simplicity, we ignore the zero-mean noise term $\xi_m^+ + \xi_m^- - 2\xi$ temporarily and analyze the following product inside the Hessian estimate:

$$\frac{1}{(2 \times 3^d)^2} \sum_{m=0}^{3^d-1} M_m \left(\sum_{i=1}^{d} (\Delta_m^i)^2 \nabla_{ii}^2 f(x_k) + 2 \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} \Delta_m^i \Delta_m^j \nabla_{ij}^2 f(x_k)) \right). \quad (3.32)$$

Off-diagonal terms in (3.32)

We now consider the (h, l)th term in (3.32): Assume without loss of generality, that $h < l$. Then,

$$\begin{aligned}
&\frac{1}{(2 \times 3^d)^2} \sum_{m=0}^{3^d-1} \left[\Delta_m^h \Delta_m^l \left(\sum_{i=1}^{d} (\Delta_m^i)^2 \nabla_{ii}^2 f(x_k) \ + 2 \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} \Delta_m^i \Delta_m^j \nabla_{ij}^2 f(x_k) \right) \right] \\
=&\frac{1}{(2 \times 3^d)^2} \sum_{i=1}^{d} \nabla_{ii}^2 f(x_k) \sum_{m=0}^{3^d-1} \left(\Delta_m^h \Delta_m^l (\Delta_m^i)^2 \right) \\
&+ \frac{1}{(2 \times 3^d)^2} \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} \nabla_{ij}^2 f(x_k) \sum_{m=0}^{3^d-1} \left(\Delta_m^h \Delta_m^l \Delta_m^i \Delta_m^j \right) \quad (3.33) \\
=&\nabla_{jl}^2 f(x_k). \quad (3.34)
\end{aligned}$$

The last equality follows from the fact that the first term in (3.33) is 0 since $h \neq l$ and

$$\sum_{m=0}^{3^d-1} \left(\Delta_m^h \Delta_m^l (\Delta_m^i)^2 \right) = 0 \text{ for any } i,$$

while the second term in (3.33) can be seen to be equal to $\nabla_{hl}^2 f(x_k)$ using induction arguments similar to proof of Lemma 2.

Diagonal terms in (3.32)

Consider the lth diagonal term inside the conditional expectation in (3.32):

$$\begin{aligned}
&\frac{\kappa}{(2\times 3^d)^2}\sum_{m=0}^{3^d-1}\left((\Delta_m^l)^2-(2\times 3^d)\right)\times\left(\sum_{i=1}^{d}(\Delta_m^i)^2\nabla_{ii}^2 f(x_k)\right.\\
&\qquad\left.+2\sum_{i=1}^{d-1}\sum_{j=i+1}^{d}\Delta_m^i\Delta_m^j\nabla_{ij}^2 f(x_k)\right)\\
=&\frac{\kappa}{(2\times 3^d)^2}\sum_{m=0}^{3^d-1}(\Delta_m^l)^2\sum_{i=1}^{d}(\Delta_m^i)^2\nabla_{ii}^2 f(x_k)\\
&\qquad+\frac{2\kappa}{(2\times 3^d)^2}\sum_{m=0}^{3^d-1}(\Delta_m^l)^2\sum_{i=1}^{d-1}\sum_{j=i+1}^{d}\Delta_m^i\Delta_m^j\nabla_{ij}^2 f(x_k)\\
&-\frac{\kappa}{(2\times 3^d)}\sum_{m=0}^{3^d-1}\sum_{i=1}^{d}(\Delta_m^i)^2\nabla_{ii}^2 f(x_k)-\frac{2\kappa}{(2\times 3^d)}\sum_{m=0}^{3^d-1}\sum_{i=1}^{d-1}\sum_{j=i+1}^{d}\Delta_m^i\Delta_m^j\nabla_{ij}^2 f(x_k).
\end{aligned}\tag{3.35}$$

Now, we analyze each of the four terms on the RHS above.

The first term on the RHS of (3.35) can be simplified as follows:

$$\begin{aligned}
&\frac{\kappa}{(2\times 3^d)^2}\sum_{m=0}^{3^d-1}(\Delta_m^l)^2\sum_{i=1}^{d}(\Delta_m^i)^2\nabla_{ii}^2 f(x_k)\\
=&\frac{\kappa}{(2\times 3^d)^2}\left(\sum_{m=0}^{3^d-1}(\Delta_m^l)^4\nabla_{ll}^2 f(x_k)+\sum_{m=0}^{3^d-1}\sum_{i=1,i\neq l}^{d}(\Delta_m^l)^2(\Delta_m^i)^2\nabla_{ii}^2 f(x_k)\right)\\
=&\frac{\kappa}{(2\times 3^d)^2}\left((2\times 3^{d+1})\nabla_{ll}^2 f(x_k)+(2\times 3^d)^2\sum_{i=1,i\neq l}^{d}\nabla_{ii}^2 f(x_k)\right).
\end{aligned}$$

For the second equality above, we have used the fact that $\sum_{m=0}^{3^d-1}(\Delta_m^l)^4 = 2\times 3^{d+1}$ (easy to infer this claim along the lines of point (5) in the proof of Lemma 1) and $\sum_{m=0}^{3^d-1}(\Delta_m^l)^2(\Delta_m^i)^2 = (2\times 3^d)^2, \forall l\neq i$.

The second term in (3.35) is zero because $\sum_{m=0}^{3^d-1}\left(\Delta_m^i\Delta_m^j(\Delta_m^l)^2\right) = 0$ for any l and $i\neq j$ – a claim that can be easily proved using an induction argument.

The third term in (3.35) without the negative sign can be simplified as follows:

$$\frac{\kappa}{(2\times 3^d)}\sum_{m=0}^{3^d-1}\sum_{i=1}^{d}(\Delta_m^i)^2\nabla_{ii}^2 f(x_k) = \frac{\kappa}{(2\times 3^d)}\sum_{i=1}^{d}\sum_{m=0}^{3^d-1}(\Delta_m^i)^2\nabla_{ii}^2 f(x_k)$$
$$= \kappa\sum_{i=1}^{d}\nabla_{ii}^2 f(x_k), \text{ a.s.}$$

Combining the above followed by some algebra, we obtain

$$\frac{\kappa}{(2\times 3^d)^2}\mathbb{E}\left[\left((\Delta_m^l)^2-(2\times 3^d)\right)\left(\sum_{i=1}^{d}(\Delta_m^i)^2\nabla_{ii}^2 f(x_k)\right.\right.$$
$$\left.\left.+2\sum_{i=1}^{d-1}\sum_{j=i+1}^{d}\Delta_m^i\Delta_m^j\nabla_{ij}^2 f(x_k)\right)\Bigg|\,\mathcal{F}_k\right]$$
$$=\nabla_{ll}^2 f(x_k). \tag{3.36}$$

The fourth term in (3.35) is zero from Lemma 1.

Denote the product in (3.32) by (A) and the (i,j)th term there by $(A)_{i,j}$. Then,

$$\mathbb{E}[\widehat{H}_k(i,j)\mid\mathcal{F}_k] = \mathbb{E}\left[(A)\mid\mathcal{F}_k\right]+\mathbb{E}[\xi_k^+ + \xi_k^- - 2\xi_k\mid\mathcal{F}_k] \tag{3.37}$$
$$=\nabla_{ij}^2 f(x_k)+O(\eta_{(k3^d)}^2).$$

The last equality above follows from (C3), while the term involving the factor (A) reduces to the true Hessian with a bias of $O(\eta_{(k3^d)}^2)$ due to (3.34) and (3.36).

The main claim follows.

Case 2: Permutation matrix-based perturbations

From the first step involving Taylor series expansions in Case 1, we have

$$\frac{f(x_k+\eta_{(kd+m)}\Delta_m)+f(x_k-\eta_{(kd+m)}\Delta_m)-2f(x_k)}{\eta_{(kd+m)}^2}$$
$$=\sum_{i=1}^{d}(\Delta_m^i)^2\nabla_{ii}^2 f(x_k)+2\sum_{i=1}^{d-1}\sum_{j=i+1}^{d}\Delta_m^i\Delta_m^j\nabla_{ij}^2 f(x_k)+O(\eta_{(kd+m)}^2). \tag{3.38}$$

Using the facts that $\sum_{m=0}^{d-1}\sum_{i=1}^{d}\Delta_m^i\Delta_m^j = 0$ for $i\neq j$ and $\sum_{m=0}^{d-1}(\Delta_m^i)^2 = 1$ for any i, we

obtain

$$\sum_{m=0}^{d-1}\left[\frac{f(x_k+\eta_{(kd+m)}\Delta_m)+f(x_k-\eta_{(kd+m)}\Delta_m)-2f(x_k)}{\eta^2_{(kd+m)}}\right]$$
$$=\sum_{i=1}^{d}\sum_{m=0}^{d-1}(\Delta_m^i)^2\nabla^2_{ii}f(x_k)+2\sum_{i=1}^{d-1}\sum_{j=i+1}^{d}\nabla^2_{ij}f(x_k)\sum_{m=0}^{d-1}\Delta_m^i\Delta_m^j+O(\eta^2_{(kd+m)}) \quad (3.39)$$
$$=\nabla^2_{ii}f(x_k)+O(\eta^2_{(kd)}).$$

Denote the term on LHS in (3.39) by (B), and following an argument similar to that used in simplifying the noise terms in (3.37), we obtain

$$\mathbb{E}[\widehat{H}_k(i,i)\mid\mathcal{F}_k]=\mathbb{E}\left[(B)\mid\mathcal{F}_k\right]+\mathbb{E}[\xi_k^+ +\xi_k^- -2\xi_k\mid\mathcal{F}_k]$$
$$=\nabla^2_{ii}f(x_k)+O(\eta^2_{(kd)}).$$

Hence proved. □

Proof of Theorem 7

Proof. Follows in a similar manner as that of the proofs of Theorem 5 and 6 in (Prashanth *et al.*, 2017). □

Proof of Theorem 8

Proof. From the quadratic assumption, we have that $H(x)$ is a constant, independent of x. From the proof leading up to (3.37), we have that $\widehat{H}_k=\nabla^2 H(x_k)=H^*$. Now, we follow the technique from (Spall, 2009) to derive the main claim.

Notice that for some $n\geq 1$,

$$\Lambda_n=(1-b_n)\Lambda_{n-1}+b_n(\overline{H}_n-H^*)=(1-b_n)\Lambda_{n-1}.$$

Unrolling the recursion above, we obtain

$$\Lambda_k=\left[\prod_{n=1}^{k}(1-b_n)\right]\Lambda_0.$$

Using the above, we can arrive at a simpler representation for the trace $\left[\mathbb{E}(\Lambda_k^T \Lambda_k)\right]$ as follows:

$$\mathbb{E}(\Lambda_k^T \Lambda_k) = \left[\prod_{n=1}^{k}(1-b_n)\right]^2 \mathbb{E}(\Lambda_0^T \Lambda_0), \text{ leading to}$$
$$\text{trace}\left[\mathbb{E}(\Lambda_k^T \Lambda_k)\right] = \left[\prod_{n=1}^{k}(1-b_n)\right]^2 \text{trace}\left[\mathbb{E}(\Lambda_0^T \Lambda_0)\right].$$

Simplifying further, using the fact that $1-b_n = e^{-b_n}(1-\mathcal{O}(b_n^2))$, with the $\mathcal{O}(b_n^2)$ being strictly positive as e^{-b_n} is convex, we have

$$\text{trace}\left[\mathbb{E}(\Lambda_k^T \Lambda_k)\right] = e^{-2b_{sum}(1,k)} c_{0k} \text{trace}\left[\mathbb{E}(\Lambda_0^T \Lambda_0)\right], \tag{3.40}$$

where $b_{sum}(i,j) = \sum_{n=i}^{j} b_n$ and $c_{nk} = \left[\prod_{i=n+1}^{k}(1-\mathcal{O}(b_i^2))\right]^2, n \geq 0$ and $c_{kk} = 1$. Since $0 < b_n < 1, \forall n \geq 2$, and $r > 1/2$, the c_{nk} are uniformly bounded in magnitude. Further,

$$b_{sum}(1,k) \geq \int_1^{k+1} \frac{b_0}{x^r} dx \geq \left(\frac{b_0}{1-r}\right)\left(k^{1-r}-1\right).$$

Using the bound derived above in (3.40), we obtain

$$\text{trace}\left[\mathbb{E}(\Lambda_k^T \Lambda_k)\right] \leq e^{-2b_0 k^{1-r}/(1-r)} e^{2b_0} c_{0k} \text{trace}\left[\mathbb{E}(\Lambda_0^T \Lambda_0)\right].$$

The main claim follows. □

3.5 Experiments

3.5.1 Implementation[5]

We consider the following problem:

$$\min_x \ \mathbb{E}_\xi \left[F(x,\xi)\right], \tag{3.41}$$

where $F(x,\xi)$ is the sample observation of the objective function $f(x)$ corrupted with zero mean noise ξ. In particular, the noise is $[x^T, 1]\xi$, where ξ is a multivariate Gaussian

[5]The implementation is available at `https://github.com/prashla/RDSA/archive/master.zip`.

distribution with mean zero and covariance $\sigma^2\mathcal{I}_{d+1}$. A similar noise structure has been used earlier in the implementation of both RDSA and SPSA algorithms in (Prashanth *et al.*, 2017) and (Spall, 2000), respectively. For all the experiments, we consider two different settings of noise: (i) low noise with $\sigma = 0.001$; and (ii) high noise with $\sigma = 0.1$.

We consider three different functional forms for $F(x, \xi)$, namely quadratic, fourth-order and Rastrigin, respectively, in both $d = 5$ and $d = 10$ dimensions, for evaluating our algorithms. Before describing the example functions, we present the details about the algorithms implemented.

We implement the first-order and second-order algorithms proposed in this chapter and compare them with several baselines that are based on the simultaneous perturbation method.

The first-order algorithms implemented include 1RDSA-Lex-DP, 1RDSA-Perm-DP and 1RDSA-KW-DP - the three deterministic perturbation variants of 1RDSA (see Section 3.2 for a detailed description). We compare these algorithms with the following baselines: RDSA with uniform and asymmetric Bernoulli perturbations proposed in (Prashanth *et al.*, 2017) and henceforth referred to as 1RDSA-Unif and 1RDSA-AsymBer, respectively, and SPSA with Bernoulli perturbations, henceforth referred to as 1SPSA.

The second-order algorithms implemented include 2RDSA-Lex-DP and 2RDSA-Perm-DP - the two deterministic perturbation variants of 2RDSA (see Section 3.3 for a detailed description). We compare these algorithms with the following baselines: second-order RDSA with uniform and asymmetric Bernoulli perturbations proposed in (Prashanth *et al.*, 2017) and henceforth referred to as 2RDSA-Unif and 2RDSA-AsymBer, respectively, and second-order SPSA with Bernoulli perturbations, henceforth referred to as 2SPSA.

The settings of the parameters η_k and γ_k for both first- and second-order algorithms are listed in Table 3.3, and a similar setting has been used in implementation of both RDSA and SPSA algorithms in (Prashanth *et al.*, 2017) and (Spall, 2000), respectively. The distribution parameter for RDSA variants is set as follows: $u = 1$ for RDSA-Unif, $\epsilon = 0.0001$ for 1RDSA-AsymBer, and $\epsilon = 1$ for 2RDSA-AsymBer, and a similar

setting has been used earlier for RDSA implementation in (Prashanth *et al.*, 2017). Each coordinate of the parameter is projected onto the set $[-2.048, 2.047]$, which helps to keep the iterates stable. All results are averages over 50 independent runs.

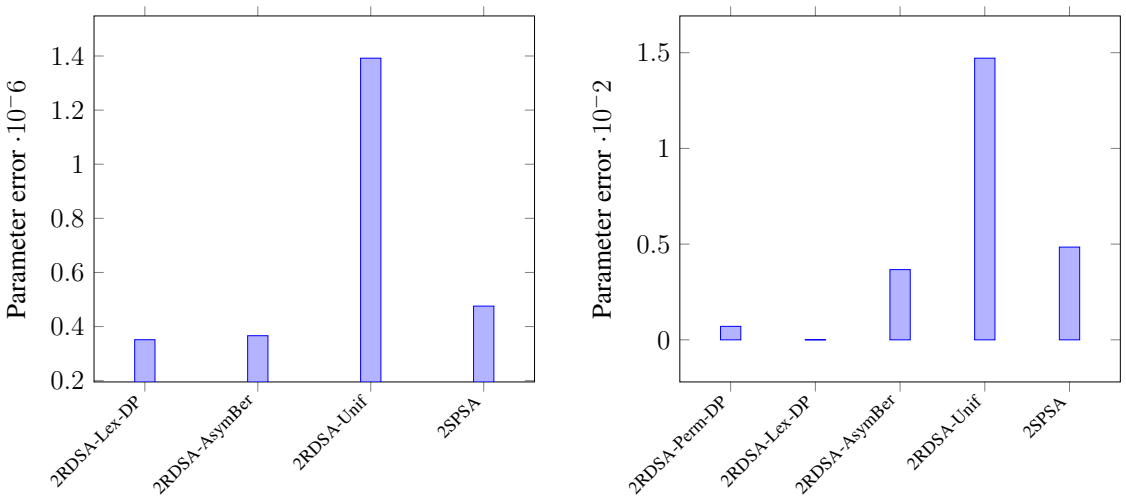
Table 3.3: Step-size and perturbation constant parameter settings, for first and second order algorithms.

Algorithms	η_k	γ_k
First-order	$1.9/k^{0.101}$	$1/(k+50)$
Second-order	$3.8/k^{0.101}$	$1/k^{0.6}$

To compare the algorithms' performance, we use parameter error as the performance metric. For a given simulaton budget, the parameter error measures the distance between the final iterate obtained after the final update iteration and the optimum parameter x^*. More precisely, we use the following form for the parameter error, after suitable normalization:

$$\text{Parameter error} = \frac{||x_\tau - x^*||^2}{||x_0 - x^*||^2},$$

where τ is the number of times x is updated until the end of the simulation. Notice that τ varies with the algorithm and the number of function measurements. For example, with a budget of 5000 measurements for $d = 10$, $\tau = 250$ for 1RDSA-Perm-DP and 1RDSA-KW-DP as they use $2d$ measurements per iteration. Further, $\tau = 2500$ for 1SPSA as well as for both variants of 1RDSA, as they use two measurements per iteration. Notice that 1RDSA-Lex-DP does not make much progress under low simulation budgets, as it requires 2×3^d measurements per iteration. On the other hand, for second-order algorithms, an initial 20% of the measurements were used up by the corresponding first-order algorithm for initialization. Thus, for $d = 10$ and a budget of 5000 measurements, the initial 1000 measurements are used for the first-order algorithm and the remaining 4000 are used by the second-order algorithm. As a consequence of the simulation budget split, the number of update iterations $\tau = 4000/30 \approx 133$ for 2RDSA-Perm-DP, $4000/4 = 1000$ for 2SPSA, and $4000/3 \approx 1333$ for 2RDSA algorithms. The 2RDSA-Lex-DP algorithm does not output a meaningful parameter with a low simulation budget, owing to its high inner loop length. The difference here is due to the fact that 2RDSA-Perm-DP uses $3d$ measurements per iteration, while 2RDSA-



(a) $\sigma = 0.001$ (b) $\sigma = 0.1$

Figure 3.1: Parameter error for various second-order algorithms under the quadratic objective (3.42) for a five-dimensional problem with a simulation budget of 50000 and $\sigma = 0.001$ and 0.1.

Lex-DP needs 3×3^d, 2RDSA needs 3 and 2SPSA needs 4.

3.5.2 Example 1: Quadratic objective

Let A be such that dA is an $d \times d$ upper triangular matrix with $a_{ij} = 1$ for $i \leq j$ and let b be an d-dimensional vector of ones. Then the quadratic objective function is defined as follows:

$$F(x, \xi) = x^T Ax + b^T x + \xi. \tag{3.42}$$

The initial point x_0 is set to the d-dimensional vector of ones and the optimal point x^* is dimension dependent. For instance, with $d = 10$, the optimal point x^*, is the 10-dimensional vector of -0.9091 for the choice of A and b described earlier. Note that $f(x^*) = \mathbb{E}_\xi[F(x^*, \xi)] = -4.55$.

Figures 3.1 and 3.2 present the parameter error for the first-order and second-order algorithms under the quadratic objective (3.42) for dimension 5 and $\sigma = 0.001$ and 0.1.

Among the first-order algorithms, for both settings of noise, 1RDSA-Perm-DP and 1RDSA-KW-DP exhibited similar performance and outperformed the other algorithms. The parameter error in 1RDSA-Perm-DP and 1RDSA-KW-DP is of the order of 10^{-5}, while for the others, the same is of the order of 10^{-3}. 1RDSA-Lex-DP (not shown in

the figure) showed a parameter error that was an order of magnitude higher than the other algorithms.

Among the second-order algorithms, for both settings of noise, 2RDSA-Lex-DP exhibited the best performance. Furthermore, it is interesting to see that for both settings of noise 2RDSA-Perm-DP, and 2RDSA-Lex-DP gave consistent performance with parameter error of the order of 10^{-4} and 10^{-7}, respectively, while the parameter error of 2RDSA-AsymBer, 2RDSA-Unif, and 2SPSA increased with noise. Further, the benefit of using second-order algorithms is more noticeable under the low noise setting.

For the low noise setting, the parameter error of 2RDSA-Perm-DP is not shown in the figure, for the sake of readability in comparing the errors of the other algorithms.

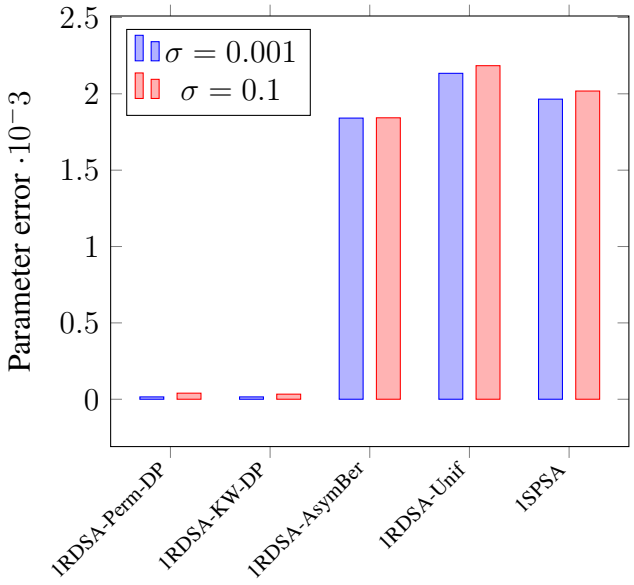


Figure 3.2: Parameter error for various first-order algorithms under the quadratic objective (3.42) for a five-dimensional problem with a simulation budget of 50000 and $\sigma = 0.001$ and 0.1.

Figure 3.3 compares the parameter error of 1RDSA-Perm-DP, 1RDSA-KW-DP, and both variants of 1RDSA and 1SPSA algorithms for the quadratic objective with dimension 10, $\sigma = 0.001$ and a simulation budget of 50000 function measurements. As in the case of the problem with dimension 5, 1RDSA-Perm-DP and 1RDSA-KW-DP performed best, while the result of 1RDSA-Lex-DP is not reported due to its high inner loop length.

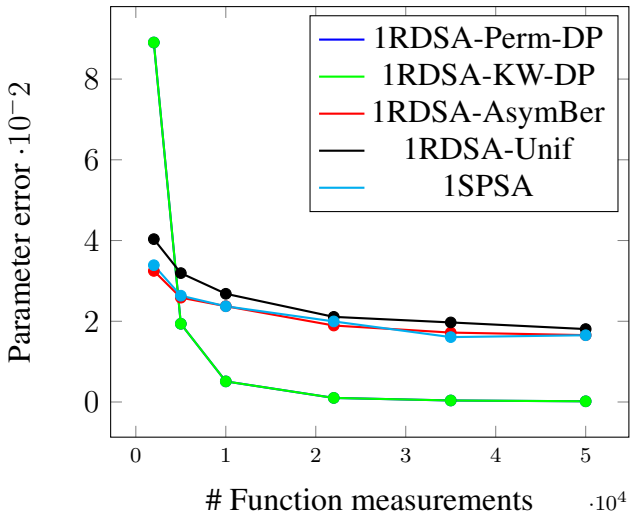


Figure 3.3: Evolution of the parameter error as the simulation budget is varied, for the first-order algorithms under the quadratic objective with $d = 10$ and $\sigma = 0.001$.

3.5.3 Example 2: Fourth-order objective

The function given below has been used for evaluating both RDSA and SPSA algorithms in (Prashanth *et al.*, 2017) and (Spall, 2000), respectively.

$$F(x, \xi) = x^T A^T A x + 0.1 \sum_{j=1}^{d} (Ax)_j^3 + 0.01 \sum_{j=1}^{d} (Ax)_j^4 + \xi, \tag{3.43}$$

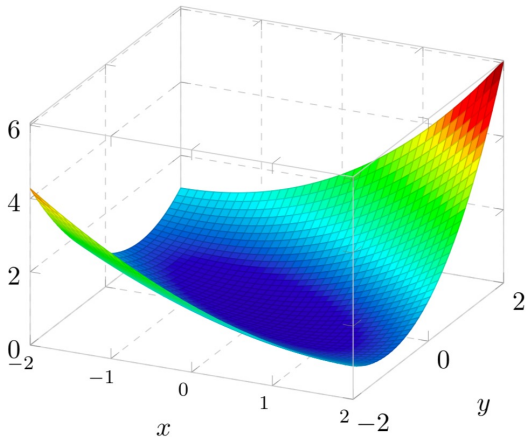
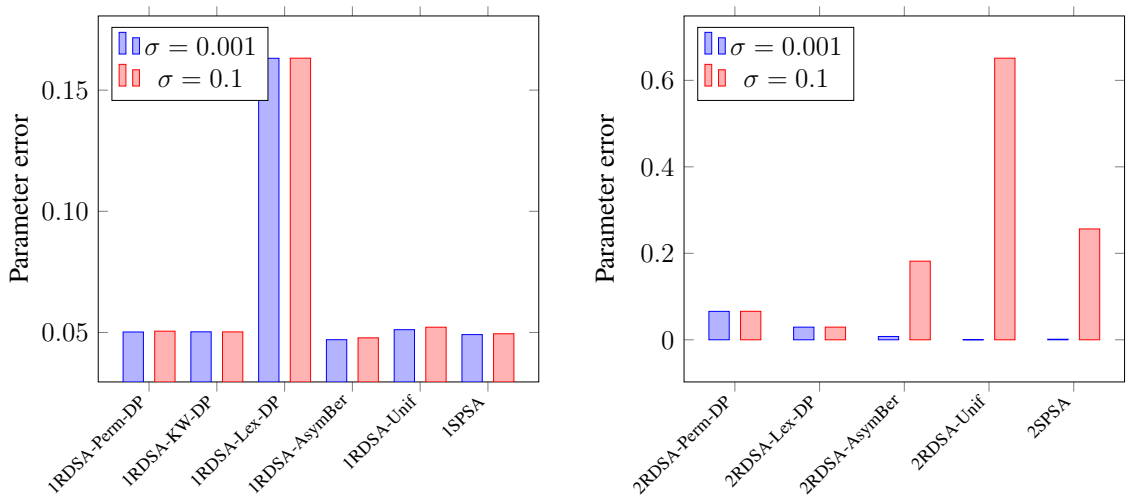


Figure 3.4: A plot of the fourth-order objective (3.43), $d = 2$.

where A and ξ are the same as in the quadratic objective. The initial point x_0 is set to the d-dimensional vector of ones and the optimal point x^* is the d-dimensional vector



(a) First-order algorithms

(b) Second-order algorithms

Figure 3.5: Parameter error for various algorithms under the fourth-order objective (3.43) for a five-dimensional problem with a simulation budget of 50000 and $\sigma = 0.001$ and 0.1.

of zeros, with $f(x^*) = \mathbb{E}_\xi[F(x^*, \xi)] = 0$. Figure 3.4 shows a plot of the fourth-order objective (3.43).

Figure 3.5 presents the parameter error for both first and second order algorithms in the case of the fourth-order objective (3.43) with dimension 5 and $\sigma = 0.001$ and 0.1. Among the first-order algorithms, for both settings of noise, all algorithms except 1RDSA-Lex-DP exhibited similar performance.

Among the second-order algorithms, similar to the quadratic case, for both settings of noise, 2RDSA-Perm-DP and 2RDSA-Lex-DP gave consistent performance with parameter error of the order of 10^{-2}. Under the low noise setting, i.e., $\sigma = 0.001$, 2RDSA-Unif exhibited the best performance, while under the high noise setting with $\sigma = 0.1$, 2RDSA-Lex-DP outperformed the other algorithms. Thus, we observe that 2RDSA-Lex-DP and 2RDSA-Perm-DP algorithms are more tolerant to the noise, as compared to their random counterparts 2RDSA-AsymBer, 2RDSA-Unif, and 2SPSA.

Since the fourth-order objective is more difficult to optimize than the quadratic one, we observe that under the low noise setting, the parameter error in the case of the fourth-order objective for the first-order algorithms is higher, of the order of 10^{-2} compared to 10^{-5} for the quadratic case. For the second-order algorithms, the same is also of the order of 10^{-2} compared to 10^{-6} for the quadratic case. A similar trend is observed in the high noise regime.

Figure 3.6 compares the parameter error of 2RDSA-Perm-DP, as well as both variants of 2RDSA and 2SPSA algorithms, for the fourth-order objective function with dimension 10, $\sigma = 0.001$ and a simulation budget of 50000 function measurements. As in the five-dimensional problem, 2RDSA-Unif and 2SPSA exhibited similar performance and outperformed the other algorithms. The results of the 2RDSA-Lex-DP algorithm are not displayed as it requires 3×3^{10} function measurements per iteration.

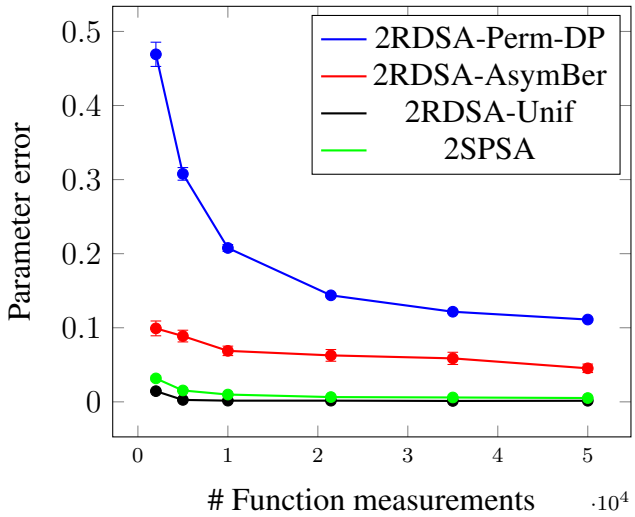


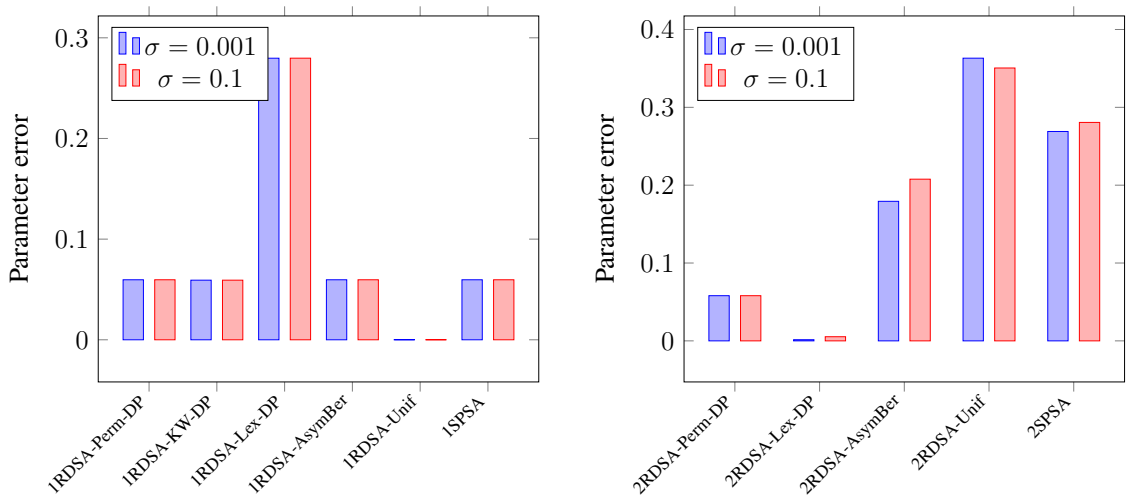
Figure 3.6: Evolution of the parameter error as the simulation budget is varied, for the second-order algorithms under the fourth-order objective with $d = 10$ and $\sigma = 0.001$.

3.5.4 Example 3: Rastrigin objective

The Rastrigin objective function is defined as follows:

$$F(x, \xi) = \sum_{i=1}^{d} (x_i^2 - 10\cos(2\pi x_i)) + 10d + 1 + \xi, \tag{3.44}$$

where ξ is the same as for the quadratic objective. The initial point x_0 is set to the d-dimensional vector of twos and the optimal point x^* is the d-dimensional vector of zeros, with $f(x^*) = \mathbb{E}_\xi[F(x^*, \xi)] = 1$. Figure 3.8 shows a plot of the Rastrigin objective (3.44), which has many local minima.



(a) First-order algorithms

(b) Second-order algorithms

Figure 3.7: Parameter error for various algorithms under the Rastrigin objective (3.44) for a five-dimensional problem and a simulation budget of 50000 and $\sigma = 0.001$ and 0.1.

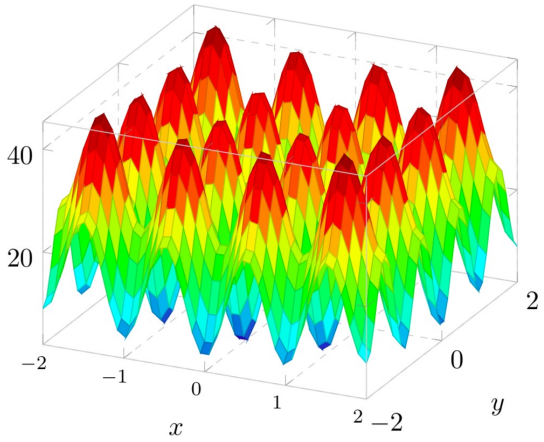


Figure 3.8: A plot of the Rastrigin objective (3.44), $d = 2$.

Figure 3.7 presents the parameter error for both first-order and second-order algorithms for the Rastrigin objective (3.44) with dimension 5 and $\sigma = 0.001$ and 0.1.

For both settings of noise, among the first-order algorithms, 1RDSA-Unif outperformed the other algorithms, while 2RDSA-Lex-DP performed the best among the second-order algorithms.

Similar to the quadratic and fourth-order objective, first-order algorithms, 2RDSA-Perm-DP and 2RDSA-Lex-DP gave consistent performance under both settings of noise.

In summary, for all the three objectives, among the first-order algorithms, we observed that 1RDSA-Perm-DP and 1RDSA-KW-DP performed best, while 1RDSA-Lex-DP showed poor performance. On the other hand, among the second-order algorithms, 2RDSA-Lex-DP exhibited the best performance.

3.6 Summary

We incorporated two novel deterministic perturbation (DP) schemes into the RDSA class of simultaneous perturbation algorithms. The proposed DP variants of first-order, as well as second-order, RDSA were shown to result in asymptotically unbiased gradient/Hessian estimates, thus resulting in provably convergent 1RDSA/2RDSA variants. We also performed numerical experiments to validate the theoretical findings.

CHAPTER 4

Non-Asymptotic Bounds for Zeroth-Order Stochastic Optimization

We consider the problem of optimizing an objective function with and without convexity in a simulation-optimization context, where only stochastic zeroth-order information is available. We consider two techniques for estimating gradient/Hessian, namely simultaneous perturbation (SP) and Gaussian smoothing (GS). We introduce an optimization oracle to capture a setting where the function measurements have an estimation error that can be controlled. Our oracle is appealing in several practical contexts where the objective has to be estimated from i.i.d. samples, and increasing the number of samples reduces the estimation error. In the stochastic non-convex optimization context, we analyze the zeroth-order variant of the randomized stochastic gradient (RSG) and quasi-Newton (RSQN) algorithms with a biased gradient/Hessian oracle, and with its variant involving an estimation error component. In particular, we provide non-asymptotic bounds on the performance of both algorithms, and our results provide a guideline for choosing the batch size for estimation, so that the overall error bound matches with the one obtained when there is no estimation error. Next, in the stochastic convex optimization setting, we provide non-asymptotic bounds that hold in expectation for the last iterate of a stochastic gradient descent (SGD) algorithm, and our bound for the GS variant of SGD matches the bound for SGD with unbiased gradient information. We perform simulation experiments on synthetic as well as real-world datasets, and the empirical results validate the theoretical findings.

4.1 Introduction

We consider the problem of minimizing a smooth objective function, when the optimization algorithm is provided with function measurements corrupted by zero-mean noise. Recall that this setting falls under the realm of simulation optimization and gradient-based methods are popular for solving such optimization problems. We study

stochastic gradient algorithms that incorporate either SP-based or GS-based gradient/Hessian estimates and provide non-asymptotic bounds in a setting where the objective is convex as well as one where it is not.

In (Hu *et al.*, 2016), the gradient estimation schemes motivated by SP and GS approaches have been formalized as biased gradient oracles. However, the aforementioned reference focused primarily on a convex objective, and derived an upper bound for a mirror-descent scheme. In contrast, we derive a matching upper bound, albeit with a regular stochastic gradient descent algorithm, with the added advantage that the step-size we employ does not require knowledge of the underlying smoothness parameter. More importantly, unlike (Hu *et al.*, 2016), we study stochastic non-convex optimization problems with the biased gradient oracles mentioned before.

We also propose a variant of the zeroth-order setting, where the objective function has to be estimated from i.i.d. samples, leading to an estimation error component. The latter model is applicable in a reinforcement learning (RL) context, where the objective is not perfectly observable, and has to be estimated from sample trajectories. We formalize this through an optimization oracle, that outputs biased gradient information, while taking in an additional input of the mini-batch size. Finally, we also consider an optimization oracle that provides a biased gradient as well as Hessian information, along with a variant that incorporates an estimation error component. We study the performance of gradient-based algorithms in the convex as well as non-convex regimes under the proposed oracle.

We summarize our contributions in the stochastic non-convex optimization context. We analyze the performance of the zeroth-order gradient as well as quasi-Newton algorithms by deriving non-asymptotic bounds. In particular, we study the randomized stochastic gradient (RSG) (Ghadimi and Lan, 2013), and randomized quasi-Newton (RSQN) (Wang *et al.*, 2017) algorithms. The case of unbiased gradient information is addressed in the aforementioned references. We consider the zeroth-order feedback model, i.e., a setting where only biased gradient information is available, and derive non-asymptotic bounds for zeroth-order variants of RSG and RSQN algorithms.

From our analysis in the stochastic non-convex optimization setting, we derive the following conclusions:

1. In the case of the zeroth-order setting without estimation error, we observe that the

overall rate for the SP method is $\mathcal{O}(N^{-1/3})$, which is weaker than the corresponding result for the GS method (i.e., $\mathcal{O}(N^{-1/2})$) (Ghadimi and Lan, 2013). This is not surprising, as the SP approach results in a gradient estimate whose variance scales inversely with the perturbation constant η, and this is unlike the Gaussian smoothing approach, where such an inverse scaling is absent.

2. In the zeroth-order setting with estimation errors, we observe that an order of $\mathcal{O}(N^{-1/2})$ (resp. $\mathcal{O}(N^{-1/3})$) bound can be obtained for GS method (resp. SP method), and this matches the rate in the model above, i.e., biased gradients without estimation error. An advantage with our approach is that, unlike (Ghadimi and Lan, 2013) approach for without estimation error setting, we do not require knowledge of the function value at the optima for choosing a smoothing parameter, which is employed in gradient estimation. Our results hold for a choice of a batch size that increases asymptotically, while a constant batch size would lead to sub-optimal rates.

3. The bounds for RSQN that are given biased gradient information with/without estimation errors, match the corresponding bounds for RSG up to constant factors. This is expected, since the net effect of RSG algorithm is that of iterate averaging in expectation. Such a finding is not surprising, and the reader is referred to an analysis of iterate averaging and second-order methods in Section 5 of (Dippon and Renz, 1997), albeit from an asymptotic convergence rate viewpoint, to see the parallel.

Next, we summarize our contributions in the stochastic convex optimization context. Using a proof technique that is similar to the one employed in the non-convex case, we provide a non-asymptotic bound for the RSG algorithm in a zeroth-order setting. A disadvantage with this approach is that it requires knowledge of the smoothness parameter for choosing stepsize. We overcome this dependency by employing a different algorithm that is based on the SGD scheme analyzed in (Jain *et al.*, 2019). We provide non-asymptotic bounds that hold in expectation for the final iterate of the stochastic gradient algorithm with biased gradient information. For the case of unbiased gradient information, the authors in (Jain *et al.*, 2019) provide a bound of the order $\mathcal{O}\left(N^{-1/2}\right)$, where N is the number of steps of the algorithm. We also provide a similar order bound, when the gradients are obtained using the GS approach. On the other hand, when SP-based gradient estimates are employed, the bound we obtain is of the order $\mathcal{O}\left(N^{-1/3}\right)$. The latter bound is not surprising, considering a matching information-theoretic lower

bound obtained in (Hu *et al.*, 2016).

Finally, we perform simulation experiments on synthetic as well as real-world data sets, and observe that: (i) RSG algorithm, when provided with unbiased gradient/Hessian information outperforms the other algorithms, and this is not surprising; and (ii) In the zeroth-order setting, among the variants of the RSG algorithm, where the variation is in the perturbation vectors used for gradient estimation, we observe that the GS method outperformed those using SP method. Among the RSQN variants, we observed that 2RDSA-Perm-DP, a recently proposed SP method that uses deterministic perturbations based on permutation matrices (Prashanth *et al.*, 2020), performed best. Moreover, RSQN variants outperformed the RSG variants.

The rest of this chapter is organized as follows: Section 4.2 presents the SP method based zeroth-order optimization oracles, Section 4.3 considers the stochastic non-convex optimization problem, and presents non-asymptotic bounds for both gradient and quasi-Newton algorithms, Section 4.4 considers the stochastic convex optimization problem and presents non-asymptotic bounds that hold in expectation for the random and last iterate of a stochastic gradient descent algorithm, Section 4.5 presents the non-asymptotic bounds using Gaussian smoothing method for both convex and non-convex objectives. Section 4.6 provides the proofs of all the bounds which are presented in this chapter, Section 4.7 describes the simulation experiments, and finally, Section 4.8 summarizes the results.

Notation: Throughout this chapter we assume $\|\cdot\| = \|\cdot\|_2$ and $\mathbf{1}_{m\times n}$ is an $m \times n$ matrix with each entry as one.

4.2 Zeroth-order optimization oracles

Recall from 2.1 that we consider the following stochastic optimization problem:

$$\min_{x\in\mathcal{W}} \{f(x) = \mathbb{E}_{\xi}[F(x,\xi)]\}, \tag{4.1}$$

where the function $f : \mathbb{R}^d \to \mathbb{R}$ is assumed to be smooth, ξ is the noise factor that captures stochastic nature of the problem, and $\mathcal{W}$ is a closed convex subset of $\mathbb{R}^d$. We operate in a *simulation optimization* setting (Fu, 2015), i.e., we are given noisy mea-

surements of the objective f. Gradient-based methods are very popular for solving the optimization problem formulated above, and we consider an iterative algorithm which obtains $\nabla f(\cdot)$ via subsequent calls to a stochastic zeroth-order oracle.

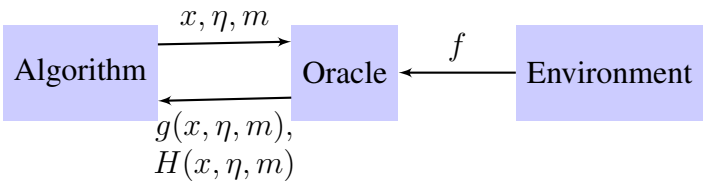


Figure 4.1: The interaction of the algorithms with a stochastic zeroth-order oracle that provides a gradient estimate $g(x, \eta, m)$ and/or a Hessian estimate $H(x, \eta, m)$ at the query point x, with perturbation parameter η and mini-batch size parameter m controlling the estimation error.

In this chapter, we consider two oracles: (i) a biased gradient oracle, and its variant involving an estimation error; and (ii) a biased gradient/Hessian oracle, with a variant involving an estimation error. Figure 4.1 shows the interaction of the algorithms with the gradient/Hessian estimation oracle and environment.

We define the oracle corresponding to (i) below.

(O1) Biased gradient oracle

Input: $x \in \mathbb{R}^d$ and perturbation parameter $\eta > 0$.

Output: a gradient estimate $g(x, \xi) \in \mathbb{R}^d$ that satisfies

(a) $\mathbb{E}_\xi \left[g\left(x, \xi\right) \right] \leq \nabla f\left(x\right) + c_1 \eta^2 \mathbf{1}_{d \times 1}$,

(b) $\mathbb{E}_\xi \left[\| g\left(x, \xi\right) - \mathbb{E}_\xi \left[g\left(x, \xi\right) \right] \|^2 \right] \leq \frac{c_2}{\eta^2}$,

for some constants $c_1, c_2 > 0$.

Gradient estimation through the simultaneous perturbation (SP) method is a popular approach (see (Bhatnagar *et al.*, 2013) for a textbook introduction), and the SP-based gradient estimates can be used to construct an oracle of type **(O1)**, assuming that the underlying function f is either three-times continuously differentiable or convex and smooth (cf. (Spall, 1992; Prashanth *et al.*, 2017, 2020; Bhatnagar *et al.*, 2013; Hu *et al.*, 2016) for a proof). Simultaneous perturbation stochastic approximation (SPSA) (Spall, 1992) and random directions stochastic approximation (RDSA) (Prashanth *et al.*, 2017) are two popular SP-based estimation schemes, and for these methods, we have $c_1 = \kappa_1 d^3$ and $c_2 = \kappa_2 d$, where $\kappa_1, \kappa_2 > 0$ are dimension-independent constants. The reader is referred to Section 4.2.1 for further details.

The second type of oracle, which is defined below, first estimates function value f as an average from m i.i.d. samples, and then uses the sample average to obtain the gradient information by using the SP method.

(**O2**) **Biased gradient oracle with estimation error**

Input: $x \in \mathbb{R}^d$, perturbation parameter $\eta > 0$, and mini-batch size $m > 0$.

Output: a gradient estimate $g(x, \xi, m) \in \mathbb{R}^d$, that satisfies

(a) $\mathbb{E}_\xi \left[g\left(x, \xi, m\right) \right] \leq \nabla f\left(x\right) + c_1 \eta^2 \mathbf{1}_{d\times 1} + \frac{c_3}{\eta\sqrt{m}} \mathbf{1}_{d\times 1}$,

(b) $\mathbb{E}_\xi \left[\left\| g\left(x, \xi, m\right) - \mathbb{E}_\xi \left[g\left(x, \xi, m\right) \right] \right\|^2 \right] \leq \frac{c_2}{\eta^2}$,

for some constants $c_1, c_2, c_3 > 0$.

The oracle outlined above is appealing in several practical applications where f has to be estimated from i.i.d. samples coming from r.v. X. E.g., let $\hat{f}_{m_k}$ be an estimate of f from m_k i.i.d. samples. Then, one usually has a Hoeffding type bond $\mathbb{P}(|\hat{f}_{m_k} - f(X)| \geq \epsilon) \leq 2\exp(-cm_k\epsilon^2)$, and this leads to $\mathbb{E}|\hat{f}_{m_k} - f(X)| \leq \frac{c_3}{\sqrt{m_k}}$, where c_3 is an absolute constant. Such a oracle can also be used in policy-gradient type RL algorithms, where one usually simulates several episodes of the underlying Markov decision process, and then estimates the value function. The latter quantity will have an estimation bias, of the order $\mathcal{O}\left(\frac{1}{\sqrt{m_k}}\right)$, where m_k is the number of episodes.

Next, we define a biased gradient/Hessian oracle below.

(**O3**) **Biased gradient/Hessian oracle**

Input: $x \in \mathbb{R}^d$ and perturbation parameter $\eta > 0$.

Output: a gradient estimate $g(x, \xi) \in \mathbb{R}^d$, and a Hessian inverse estimate $H(x, \xi) \in \mathbb{R}^{d\times d}$. These quantities satisfy

(a) Same as **(O1)**-(a),

(b) Same as **(O1)**-(b),

(c) $\mathbb{E}_\xi \left[H(x, \xi) \right] \leq H(x) + c_1^{'} \eta^2 \mathbf{1}_{d\times d}$,

for some constant $c_1^{'} > 0$.

SP-based methods can be used to obtain estimates of the Hessian, in addition to gradient estimates, when the underlying function f is either four-times continuously differentiable or convex and smooth. The reader is referred to Lemma 6 in (Prashanth *et al.*, 2020) or Lemma 7.11 in (Bhatnagar *et al.*, 2013) for an SP-based Hessian estimate that satisfies the condition (c) above.

Next, we define a variant of (**O3**) that is along of the lines of (**O2**).

(**O4**) **Biased gradient/Hessian oracle with estimation error**

Input: $x \in \mathbb{R}^d$, perturbation parameter $\eta > 0$ and mini-batch size $m > 0$.

Output: a gradient estimate $g(x, \xi) \in \mathbb{R}^d$, and a Hessian inverse estimate $H(x, \xi, m) \in \mathbb{R}^{d \times d}$. These quantities satisfy

(a) Same as (**O2**)-(a),

(b) Same as (**O2**)-(b),

(c) $\mathbb{E}_\xi \left[H(x, \xi, m) \right] \leq H(x) + c_1^{'} \eta^2 \mathbf{1}_{d \times d} + \frac{c_3}{\eta \sqrt{m}} \mathbf{1}_{d \times d}$,

for some constants $c_1^{'}, c_3 > 0$.

We also consider an alternative gradient estimation scheme based on the idea of Gaussian smoothing (GS) (Nesterov and Spokoiny, 2017) method. Variant of oracles **(O1)** and (**O2**), motivated by the GS method are presented in Section 4.5.

4.2.1 Value of constants for the SP-based oracles

The constants for the various SP-based gradient estimates depend on the type of random perturbation used, and also, the nature of the objective, i.e., whether it is convex or not. We summarize these constants below, while hiding the dependence on the moments of the random perturbation inside constant factors.

1. If the function f is three-times continuously differentiable, then the constants c_1 and c_2 are as follows (see (Spall, 1992; Prashanth *et al.*, 2017, 2020)):

$$c_1 = \alpha_0 d^3 \quad \text{and} \quad c_2 = \alpha_1 d,$$

where the constant α_0 depends on the second moment of the random perturbation employed in the gradient estimate, and a bound on the third derivative of the objective f. The constant α_1 depends on the variance of the measurement noise.

2. If the function f is convex and smooth, then the constants c_1 and c_2 are as follows (see (Hu *et al.*, 2016)):

$$c_1 = \alpha_0 L d^2 \quad \text{and} \quad c_2 = \alpha_1 d,$$

where L is the Lipschitz constant defined in (**A1**), α_0 is a constant that depends on the second moment of the random perturbation employed in the gradient estimate,

and α_1 is a constant that depends on the variance of the measurement noise.

The constant c_1', which features in the bias of the Hessian estimate in oracle (**O3**), is of the same order as c_1, in terms of the dependence on the dimension d.

4.3 Stochastic Non-convex Optimization

In this section, we consider the problem in (4.1), where the objective f is not assumed to be convex. We analyze gradient-based algorithms for solving (4.1), under the following smoothness assumption:

(**A1**) Function f has Lipschitz continuous gradient with constant $L > 0$, i.e.,

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^d.$$

We study the performance of the randomized stochastic gradient and quasi-Newton algorithms, proposed in (Ghadimi and Lan, 2013; Wang *et al.*, 2017). The gradient method is analyzed in the section below, while the quasi-Newton method is handled in the subsequent section.

We make the following assumption for the analysis of the gradient-based methods in a zeroth-order setting (a similar assumption is used in (Balasubramanian and Ghadimi, 2018)):

(**A2**) The feasible set $\mathcal{W}$ is bounded and there exists a constant $B > 0$ such that $\|\nabla f(x)\|_1 \leq B, \forall x \in \mathcal{W}$.

4.3.1 Zeroth-order randomized stochastic gradient (ZRSG)

The pseudocode for the ZRSG algorithm is given below. The ZRSG algorithm performs an incremental update as defined in (4.2), and outputs a random iterate, after N iterations.

Bounding the optimization error, i.e., $f(x_N) - f(x^*)$ is difficult, when the objective is non-convex. However, a popular alternative is to show that the RSG algorithm converges to a point, where the gradient of the objective is small (quantified by a bound

Algorithm 3 Zeroth-order Randomized Stochastic Gradient (ZRSG)

Input: Initial point $x_1 \in \mathcal{W}$, iteration limit N, stepsizes γ_k, perturbation parameter η_k, mini-batch size m_k (for the oracle (**O2**) with estimation error), projection operator $\Pi_{\mathcal{W}}$, and probability mass function $P_R(\cdot)$ supported on $\{1, \ldots, N\}$ (Let R denote the corresponding random variable).

for $k = 1, \ldots, R$ **do**

Call the oracle **(O1)** with x_k and η_k, or call the oracle (**O2**) with x_k, η_k and m_k, to obtain the gradient estimate g_k.

Perform the following stochastic gradient update:

$$x_{k+1} = \Pi_{\mathcal{W}}(x_k - \gamma_k g_k), \tag{4.2}$$

where $\Pi_{\mathcal{W}}$ is a operator that projects on to the closed convex set $\mathcal{W} \subset \mathbb{R}^d$ and

$$g_k = \begin{cases} g\left(x_k, \xi_k\right) & \text{with (\textbf{O1}),} \\ g\left(x_k, \xi_k, m_k\right) & \text{with (\textbf{O2}).} \end{cases} \tag{4.3}$$

end for

Return x_R.

on the squared norm of the gradient) (cf. (Ghadimi and Lan, 2013; Bottou *et al.*, 2018; Wang *et al.*, 2017)), and the following definition makes the optimization objective apparent.

Definition 1. ***(ϵ-stationary point)*** *Let x_R be the output of an algorithm. Then, x_R is called an ϵ-stationary point of problem* (4.1)*, if $\mathbb{E}\left\|\nabla f\left(x_R\right)\right\|^2 \leq \epsilon$.*

We provide below a non-asymptotic bound for ZRSG with the oracle **(O1)**[1]. The oracle variant with estimation error is handled in the subsequent theorem.

Theorem 9. ***(ZRSG with the oracle (O1))***

*Assume (**A1**) and (**A2**). With the oracle **(O1)**, suppose that the ZRSG algorithm is run with the stepsize γ_k and perturbation constant η_k set as follows:*

$$\gamma_k = min\left\{\frac{1}{L}, \frac{1}{(d^2N)^{2/3}}\right\}, \qquad \textit{and} \qquad \eta_k = \frac{1}{(d^5N)^{1/6}}, \qquad \forall k \geq 1. \tag{4.4}$$

Then, for any $N \geq 1$, we have

$$\mathbb{E}\left\|\nabla f\left(x_R\right)\right\|^2 \leq \mathcal{B}_{\mathcal{SP}} := \frac{2LD_f}{N} + \frac{\mathcal{Z}_1}{N^{1/3}},$$

[1]The bounds in Section 4.3 are for a random iterate x_R, where R is uniformly distributed over $\{1, \ldots, N\}$, and the expectation is taken with respect to R and noise ξ.

where $\mathcal{Z}_1 = 2D_f d^{4/3} + \frac{4Bc_1}{d^{5/3}} + \frac{Lc_1^2}{d^{11/3}N} + Lc_2 d^{1/3}$*, constants* c_1, c_2 *are defined in* ***(O1)****,* B *is as defined in* ***(A2)****,*

$$D_f = f(x_1) - f(x^*), \tag{4.5}$$

and x^* *is an optimal solution to* (4.1)*.*

Proof. See Section 4.6.1. □

The overall rate, from the bound above, is $\mathcal{O}\left(N^{-1/3}\right)$, and this is not surprising because the bias of the gradient cannot be made arbitrarily small by setting η to a low value, as the variance of the gradient estimates scales inversely with η. The (asymptotic) convergence rate results for SPSA in (Spall, 1992), and RDSA in (Prashanth *et al.*, 2017), also exhibit the same order.

Using the bound in Theorem 9, it is easy to see that the total number of iterations required for finding an ϵ-stationary point is at most $\mathcal{O}(\frac{d^4}{\epsilon^3})$. The stepsize γ_k and perturbation constant η_k are chosen as in (4.4), so that the overall rate is $\mathcal{O}(\frac{d^4}{\epsilon^3})$ for finding an ϵ-stationary point. In arriving at this choice, we have considered dimension dependence in the constants c_1 and c_2 (see Section 4.2.1).

Remark 3. *In (Ghadimi and Lan, 2013), the authors derive a non-asymptotic bound for a zeroth-order variant of their RSG algorithm under an oracle that is a variant to* ***(O1)*** *(see Section 4.5 below). Our result in Theorem 9 matches their bound. Moreover, unlike (Ghadimi and Lan, 2013), we derive a non-asymptotic bound for the oracle (****O2****), which involves an estimation error component (see Theorem 10 below).*

An advantage with our analysis is that it allows a simpler distribution for picking the final iterate (see Proposition 1 in the Section 4.6.1). In particular, our bounds hold for an iterate x_R *that is picked uniformly at random from* $\{x_1, \ldots, x_N\}$*. The net effect is that of iterate averaging, except that the averaging happens in expectation.*

Theorem 10. ***(ZRSG with the oracle (O2))***
*Assume (****A1****) and (****A2****). With the oracle (****O2****), suppose that the ZRSG algorithm is run with the stepsize* γ_k *and perturbation constant* η_k *set as defined in* (4.4)*.*

(i) *If the mini-batch size $m_k = N, \forall k \geq 1$, then, for any $N \geq 1$, we have*

$$\mathbb{E}\left\|\nabla f\left(x_R\right)\right\|^2 \leq \mathcal{B}_{\mathcal{SP}} + \frac{\mathcal{Z}_2}{N^{1/3}}, \tag{4.6}$$

*where $\mathcal{Z}_2 = 4Bc_3d^{5/6} + \frac{L}{N}\left(c_3^2 d^{1/2} + \frac{2c_1c_3}{d^{7/6}}\right)$, constants c_1, c_2 and c_3 are as defined in (**O2**), B is as defined in (**A2**), and $\mathcal{B}_{\mathcal{SP}}$ is as defined in Theorem 9.*

(ii) *If the mini-batch size $m_k = k^\beta, \forall k \geq 1$, for some constant $\beta > 0$, then, for any $N \geq 1$, we have*

$$\mathbb{E}\left\|\nabla f\left(x_R\right)\right\|^2 \leq \mathcal{B}_{\mathcal{SP}} + \underbrace{\frac{4Bc_3d^{5/6}}{N^{\frac{3\beta-1}{6}}\left(-\frac{\beta}{2}+1\right)}}_{(I)} + \underbrace{\frac{2Lc_1c_3}{d^{7/6}N^{\frac{3\beta+5}{6}}\left(-\frac{\beta}{2}+1\right)}}_{(II)} + \underbrace{\frac{Lc_3^2d^{4/3}}{N^{\frac{3\beta+1}{3}}\left(-\beta+1\right)}}_{(III)},$$

where constants are the same as in part (i).

Proof. See Section 4.6.1. □

It is interesting to note that, even with estimation error, the mini-batch size m_k can be controlled to recover a rate that matches the order in the oracle **(O1)** up to constant factors (see Theorem 9). As before, the total number of iterations required for finding an ϵ-stationary point is at most $\mathcal{O}(\frac{d^4}{\epsilon^3})$.

From Theorem 10, it is apparent that increasing the mini-batches at a rate k^β, with $\beta > 2$, leads to a better bound as compared to the case when the batch sizes increase linearly with N. More precisely, while the overall order of the bound remains $\mathcal{O}\left(N^{-1/3}\right)$, the terms marked $(I), (II)$ and (III) are significantly smaller in the case when $\beta > 2$.

Remark 4. *By a completely parallel argument to that in the proof of Theorem 10, one can infer that a constant batch size, i.e., $m_k \equiv m_0$, would result in an order $\mathcal{O}\left(N^{-1/6}\right)$ bound. The latter bound is clearly inferior to those with increasing batch sizes.*

4.3.2 Zeroth-order randomized stochastic quasi-Newton (ZRSQN) method

The zeroth-order variant of RSQN (Wang *et al.*, 2017) is presented below. As with ZRSG, the algoritm below picks a random iteration, after N update iterations using

(4.7).

Algorithm 4 Zeroth-order Randomized Stochastic quasi-Newton (ZRSQN)

Input: Initial point $x_1 \in \mathcal{W}$, iteration limit N, stepsizes γ_k, perturbation parameter η_k, mini-batch size m_k (for the oracle (**O4**) with estimation error), projection operator $\Pi_{\mathcal{W}}$, and probability mass function $P_R(\cdot)$ supported on $\{1, \ldots, N\}$ (Let R denote the corresponding random variable).
for $k = 1, \ldots, R$ **do**
Call the oracle (**O3**) with x_k and η_k, or call the oracle (**O4**) with x_k, η_k and m_k, to obtain the gradient estimate g_k, and a Hessian inverse estimate H_k.
Perform the following stochastic quasi-Newton update:

$$x_{k+1} = \Pi_{\mathcal{W}}(x_k - \gamma_k H_k g_k), \tag{4.7}$$

where $\Pi_{\mathcal{W}}$ is a operator that projects on to the closed convex set $\mathcal{W} \subset \mathbb{R}^d$, g_k is as defined in (4.3) and

$$H_k = \begin{cases} H\left(x_k, \xi_k\right) & \text{with (\textbf{O3}),} \\ H\left(x_k, \xi_k, m_k\right) & \text{with (\textbf{O4}).} \end{cases}$$

end for
Return x_R.

For the sake of analysis, we make the following assumption:

(**A3**) For any $k \geq 1$,
(a) There exist a positive constant $\Lambda < \infty$ such that, $-\Lambda I \preceq \nabla^2 f(x_k) \preceq \Lambda I$, and

(b) there exist positive constants $C_l, C_u < \infty$ such that, $C_l I \preceq H(x_k, \xi_k) \preceq C_u I$, and $C_l I \preceq H(x_k, \xi_k, m_k) \preceq C_u I$,

where the notation $A \succeq B$ with $A, B \in \mathbb{R}^{d \times d}$ means that $A - B$ is positive semidefinite.

The assumption above can be ensured by having $H(x_k, \xi_k) = [\Upsilon(B(x_k, \xi_k))]^{-1}, \forall k \geq 1$, where $B(x_k, \xi_k)$ is an approximation of the Hessian $\nabla^2 f(x_k)$, and the projection operator $\Upsilon(B(x_k, \xi_k))$ is defined as performing an eigen-decomposition of matrix $B(x_k, \xi_k)$ followed by projecting the eigenvalues on to the range $[C_l, C_u]$, as discussed in (Prashanth *et al.*, 2020).

We provide below a non-asymptotic bound for ZRSQN with the oracle (**O3**). The subsequent theorem handles the oracle variant that involves an estimation error component.

Theorem 11. ***(ZRSQN with the oracle (O3))***

*Assume (**A2**) and (**A3**). With the oracle (**O3**), suppose that the ZRSQN algorithm is run with the stepsize γ_k and perturbation constant η_k set as follows:*

$$\gamma_k = min\left\{\frac{2C_l - 1}{\Lambda C_u^2}, \frac{1}{(d^2N)^{2/3}}\right\}, \qquad \textit{and} \qquad \eta_k = \frac{1}{(d^5N)^{1/6}}, \qquad \forall k \geq 1, \quad (4.8)$$

*where Λ, C_l, and C_u are as in (**A3**). Then, for any $N \geq 1$, we have*

$$\mathbb{E}\left\|\nabla f\left(x_R\right)\right\|^2 \leq \frac{2\Lambda C_u^2 D_f}{2NC_l - N} + \frac{\mathcal{Z}_3}{N^{1/3}},$$

*where $\mathcal{Z}_3 = 2D_f d^{4/3} + \Lambda C_u^2(\frac{c_1^2}{d^{11/3}N} + c_2 d^{1/3}) + \frac{2B}{d^{5/3}}(3c_1C_l + \frac{c_1 c_1^{'}}{d^{2/3}N^{1/3}} + c_1^{'}B)$, constants c_1, c_2 are as defined in **(O1)**, B is as defined in (**A2**), and D_f is as defined in* (4.5).

Proof. See Section 4.6.2. □

A second-order method such as RSQN would provide a rate similar to that in RSG, since the net effect of RSG algorithm is that of iterate averaging in expectation. Such a finding is not surprising, and the reader is referred to an analysis of iterate averaging and second-order methods in Section 5 of (Dippon and Renz, 1997), albeit from an asymptotic convergence rate viewpoint, to see the parallel.

Comparing the bound obtained above with that in Theorem 9, we observe that, the initial error (the first term in either bound) that relates to the starting point of the algorithm is forgotten a little faster in the quasi-Newton case, while the other term matches up to constant factors.

Theorem 12. ***(ZRSQN with the oracle (O4))***

*Assume (**A2**) and (**A3**). With the oracle (**O4**), suppose that the ZRSQN algorithm is run with the stepsize γ_k and perturbation constant η_k set as in Theorem 11, and mini-batch size $m_k = N, \forall k \geq 1$. Then, $\forall N \geq 1$, we have*

$$\mathbb{E}\left\|\nabla f\left(x_R\right)\right\|^2 \leq \frac{2\Lambda C_u^2 D_f}{2NC_l - N} + \frac{\mathcal{Z}_4}{N^{1/3}},$$

*where $\mathcal{Z}_4 = 2D_f d^{4/3} + 2B\left(B\mathcal{K}_2 + \mathcal{K}_1\left(3C_l + \frac{d\mathcal{K}_2}{N^{1/3}}\right)\right) + \Lambda C_u^2\left(\frac{\mathcal{K}_1^2}{d^{1/3}N} + c_2 d^{1/3}\right)$, $\mathcal{K}_1 = c_1 d^{-5/3} + c_3 d^{5/6}$, $\mathcal{K}_2 = c_1^{'} d^{-5/3} + c_3 d^{5/6}$, constants c_1, $c_1^{'}$, c_2 and c_3 are as defined in (**O4**), B is as defined in (**A2**), and D_f is as defined in* (4.5).

Proof. See Section 4.6.2. □

From the bounds in Theorems 11 and 12, we observe that the number of iterations required for finding an ϵ-stationary point is at most $\mathcal{O}(\frac{d^4}{\epsilon^3})$.

As in the case of ZRSG, the stepsize γ_k and perturbation constant η_k are chosen as in (4.8), so that the overall rate is $\mathcal{O}(\frac{d^4}{\epsilon^3})$ for finding an ϵ-stationary point. In arriving at this choice, we have considered dimension dependence in the constants $c_1, c_1^{'}$ and c_2 in oracles (**O3**) and (**O4**).

4.4 Stochastic Convex Optimization

In this section, we consider the problem in (4.1), under the assumption that f is a convex function, and $\mathcal{W}$ is a bounded convex set. These assumptions are made precise below.

(**A4**) The function f satisfies $\|\nabla f(x)\| \leq G$, for every $x \in \mathcal{W}$.

(**A5**) The set $\mathcal{W}$ is convex and compact. Further, $\|x - y\| \leq D, \forall x, y \in \mathcal{W}$, for some $D > 0$.

Note that the function f is not assumed to be strongly convex. Let $x^* \in \mathcal{W}$ be a minimizer of $f(\cdot)$. We first analyze the ZRSG algorithm in a convex setting, and subsequently present the ZSGD algorithm, which is a zeroth-order variant of the algorithm in (Jain *et al.*, 2019).

4.4.1 Zeroth-order randomized stochastic gradient (ZRSG)

We provide below a non-asymptotic bound for ZRSG with the oracle **(O1)**. The subsequent theorem handles the oracle variant that involves an estimation error component.

Theorem 13. ***(ZRSG with the oracle (O1))***
*Assume (**A1**) and (**A5**). With the oracle **(O1)**, suppose that the ZRSG algorithm is run with the stepsize γ_k and perturbation constant η_k set as defined in* (4.4)*, then, for any $N \geq 1$, we have*

$$\mathbb{E}\left[f\left(x_R\right)\right] - f(x^*) \leq \frac{LD^2}{N} + \frac{\mathcal{K}_1}{N^{1/3}},$$

where $\mathcal{K}_1 = D^2 d^{4/3} + \frac{4Dc_1}{d^{7/6}} + \frac{c_1^2}{d^{11/3}N} + d^{1/3}c_2$*, constants* c_1 *and* c_2 *are as defined in* ***(O1)****, and* D *as defined in* ***(A5)****.*

Proof. See Section 4.6.3. □

The $\mathcal{O}(N^{-1/3})$ bound of the RHS above matches that in Theorem 9 with the non-convex objective. However, unlike non-convex case, we bound the optimization error $\mathbb{E}[f(x_R)] - f(x^*)$ and as a result few terms are independent of L. A similar observation holds for the Theorem 14 below with the oracle involving an estimation error component.

Theorem 14. ***(ZRSG with the oracle (O2))***

Assume ***(A1)*** *and* ***(A5)****. With the oracle* ***(O2)****, suppose that the ZRSG algorithm is run with the stepsize* γ_k*, perturbation constant* η_k *and mini-batch size* m_k *set as follows:*

$$\gamma_k = min\left\{\frac{1}{L}, \frac{1}{(d^2N)^{2/3}}\right\}, \qquad \eta_k = \frac{1}{(d^5N)^{1/6}}, \qquad \textit{and} \qquad m_k = N, \qquad \forall k \geq 1. \tag{4.9}$$

Then, for any $N \geq 1$*, we have*

$$\mathbb{E}\left[f\left(x_R\right)\right] - f(x^*) \leq \frac{LD^2}{N} + \frac{\mathcal{K}_2}{N^{1/3}},$$

where $\mathcal{K}_2 = D^2 d^{4/3} + 4\sqrt{d}D\left(\frac{c_1}{d^{5/3}} + c_3 d^{5/6}\right) + \frac{c_1^2}{d^{11/3}N} + \frac{2c_1c_3}{d^{7/6}N} + \frac{d^{4/3}c_3^2}{N} + d^{1/3}c_2$*, constants* c_1*,* c_2 *and* c_3 *are as defined in* ***(O2)****, and* D *as defined in* ***(A5)****.*

Proof. See Section 4.6.3. □

In the next section, we study a zeroth-order stochastic gradient descent (ZSGD) method, to derive non-asymptotic bound on the optimization error for the last iterate, i.e., $\mathbb{E}[f(x_N)] - f(x^*)$. This is unlike the bounds in Theorem 13 and 14 for ZRSG, which was for a random iterate. In practice, the last iterate is usually preferred. Moreover, the analysis in ZSGD is superior to that of ZRSG, because it does not require smoothness in the analysis.

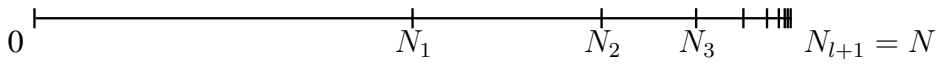


Figure 4.2: Spliting of the horizon into phases

4.4.2 Zeroth-order stochastic gradient descent (ZSGD)

The pseudocode for the ZSGD algorithm, which is designed to minimize f, given biased gradient measurements, through the oracle **(O1)** or (**O2**) is given below.

Algorithm 5 Zeroth-order Stochastic Gradient Descent (ZSGD)

Input: Initial point $x_1 \in \mathcal{W}$, iteration limit N, stepsizes γ_k, perturbation parameter η_k, mini-batch size m_k (for the oracle (**O2**) with estimation error) and projection operator $\Pi_{\mathcal{W}}$.

for $k = 1, \ldots, N$ **do**

Call the oracle **(O1)** with x_k and η_k, or call the oracle (**O2**) with x_k, η_k and m_k, to obtain the gradient estimate g_k.

Perform the following stochastic gradient update:

$$x_{k+1} \leftarrow \Pi_{\mathcal{W}}\left(x_k - \gamma_k g_k\right), \tag{4.10}$$

where $\Pi_{\mathcal{W}}$ is a operator that projects on to the closed convex set $\mathcal{W} \subset \mathbb{R}^d$ and g_k is as defined in (4.3).

end for

Return x_N.

We follow the approach from (Jain *et al.*, 2019), i.e., we assume the knowledge of N, which is the total number of iterations of ZSGD, and split the horizon N into l phases as shown in Figure 4.2. The choice of phase lengths, and the step-size decay in each phase is performed along the lines of (Jain *et al.*, 2019). However, unlike their work that assumed unbiased gradient information, we operate in a setting where biased gradient information is available through oracle **(O1)**, and this induces significant deviations in the proof. Morever, our setting features a perturbation constant parameter, which has to be chosen in a phase-dependent manner as well. We make the choice of phases precise below.

$$\begin{aligned} &\text{Let } l := \inf\{i : N \cdot 2^{-i} \leq 1\}, \\ &N_i := N - \lceil N \cdot 2^{-i} \rceil,\ 0 \leq i \leq l, \text{ and } N_{l+1} := N. \end{aligned} \tag{4.11}$$

From the phase definitions above, it can be seen that N_i is an increasing sequence. Further, $N_1 \approx \frac{N}{2}, N_2 \approx \frac{N}{2} + \frac{N}{4}$, and so on. In the theorem below, we provide a non-asymptotic bound on the optimization error, i.e., $\mathbb{E}[f(x_N)] - f(x^*)$ for the ZSGD with

the oracle **(O1)** and (**O2**).

Theorem 15. ***(ZSGD with the oracle (O1))***
*Assume (**A4**) and (**A5**). With the oracle **(O1)**, suppose that the ZSGD algorithm is run with the stepsize γ_k and perturbation constant η_k set as follows:*

$$\gamma_k = \frac{C \cdot 2^{-i}}{\sqrt{d}N^{2/3}}, \quad \textit{and} \quad \eta_k = \frac{2^{-i/4}}{\sqrt{d}N^{1/6}}, \tag{4.12}$$

when $N_i < k \leq N_{i+1}, 0 \leq i \leq l$, where $C > 0$ and N_i, l is as defined in (4.11)*. Then, for any $N \geq 4$, we have*

$$\mathbb{E}[f(x_N)] - f(x^*) \leq \frac{\mathcal{K}_3}{N^{1/3}},$$

*where $\mathcal{K}_3 = \frac{4\sqrt{d}D^2}{C} + \frac{11CG^2}{\sqrt{d}N^{1/3}} + \frac{39c_1D}{d} + \frac{20Cc_1G}{\sqrt{d}N^{2/3}} + \frac{10Cc_1^2}{d^{3/2}N} + 18\sqrt{d}Cc_2$, and constants c_1, c_2 are as defined in **(O1)**.*

Proof. See Section 4.6.4. □

The overall rate, from the bound above, is $\mathcal{O}\left(N^{-1/3}\right)$, and as discussed in Theorem 9, this is not surprising since the perturbation parameter η relates to bias-variance tradeoff. Moreover, a lower bound in (Hu *et al.*, 2016) shows that, with a biased gradient oracle (such as **(O1)**), the optimization error $(\mathbb{E}[f(x_N)] - f(x^*))$ is $\Omega\left(N^{-1/3}\right)$ in a minimax (or information-theoretic) sense for the case of a convex objective f.

Unlike (Hu *et al.*, 2016), we derive a matching upper bound, albeit with a regular SGD algorithm, with the added advantage that the stepsize we employ does not require knowledge of the underlying smoothness parameter.

The theorem below provides a bound for the case when ZSGD algorithm is run with the oracle (**O2**), which contains an estimation error component.

Theorem 16. ***(ZSGD with the oracle (O2))***
*Assume (**A4**) and (**A5**). With the oracle (**O2**), suppose that the ZSGD algorithm is run with the stepsize γ_k, perturbation constant η_k and mini-batch size m_k set as follows:*

$$\gamma_k = \frac{C \cdot 2^{-i}}{\sqrt{d}N^{2/3}}, \qquad \eta_k = \frac{2^{-i/4}}{\sqrt{d}N^{1/6}}, \quad \textit{and} \quad m_k = 2^i N, \tag{4.13}$$

when $N_i < k \leq N_{i+1}, 0 \leq i \leq l$, where $C > 0$ and N_i, l is as defined in (4.11)*. Then,*

for any $N \geq 4$, we have

$$\mathbb{E}[f(x_N)] - f(x^*) \leq \frac{\mathcal{K}_4}{N^{1/3}},$$

where $\mathcal{K}_4 = \frac{4\sqrt{d}D^2}{C} + \frac{11CG^2}{\sqrt{d}N^{1/3}} + D(39c_1d^{-1} + 67\sqrt{d}c_3) + \frac{20Cc_1G(d^{-1/2}+dc_3)}{N^{2/3}} + \frac{10C(c_1d^{-1/2}+dc_3)^2}{\sqrt{d}N} + 18\sqrt{d}Cc_2$*, and constants* c_1, c_2, c_3 *are as defined in (**O2**).*

Proof. See Section 4.6.4. □

Interestingly, the bound above matches the one obtained for ZSGD with **(O1)**, and this is because of an increasing mini-batch size m_k, which is also phase-dependent.

The analysis used in arriving at the bounds in Theorems 15 and 16 cannot be extended to the non-convex case. This is because the analysis takes a dual viewpoint and approaches the minima of the objective from below, and in this process, convexity is strictly necessary. Intuitively, it may be challenging to provide bounds for the last iterate sans averaging in a non-convex optimization setting, while it is possible to provide bounds for the averaged iterates (or the random iterate of ZRSG, which is an average in expectation) in the non-convex case.

4.5 Gaussian Smoothing

In this section, we define variants of the oracles **(O1)** and (**O2**), and derive non-asymptotic bounds that are parallel to those in Theorems 10, 14 and 15.

4.5.1 Zeroth-order optimization oracles

The biased gradient oracle variant is defined below.

(**O1'**) **Biased gradient oracle - variant**

Input: $x \in \mathbb{R}^d$ and smoothing parameter $\eta > 0$.

Output: a gradient estimate $g(x, \xi) \in \mathbb{R}^d$ that satisfies

(a) $\mathbb{E}_\xi [g(x, \xi)] \leq \nabla f(x) + c_1\eta \mathbf{1}_{d\times 1}$,

(b) $\mathbb{E}_\xi \left[\|g(x, \xi) - \mathbb{E}_\xi [g(x, \xi)]\|^2\right] \leq c_2\eta^2 + \widetilde{c_2}$,

for some constants $c_1, c_2, \widetilde{c_2} > 0$.

The oracle defined above can be constructed using the Gaussian smoothing approach, proposed in (Katkovnik and Kulchitsky, 1972), and studied later in a convex optimization setting in (Nesterov and Spokoiny, 2017). In particular, the reader is referred to Lemma 3 in (Nesterov and Spokoiny, 2017) and Lemma B.1 in (Balasubramanian and Ghadimi, 2018) for constructing a gradient estimate that satisfies conditions (a) and (b), respectively.

Next, we define a variant of (**O2**), motivated by the GS approach.

(**O2'**) **Biased gradient oracle with estimation error - variant**

Input: $x \in \mathbb{R}^d$, smoothing parameter $\eta > 0$ and mini-batch size $m > 0$.

Output: a gradient estimate $g(x, \xi, m) \in \mathbb{R}^d$, such that the following hold:

(a) $\mathbb{E}_\xi \left[g\left(x, \xi, m\right) \right] \leq \nabla f\left(x\right) + c_1 \eta \mathbf{1}_{d\times 1} + \frac{c_3}{\eta\sqrt{m}} \mathbf{1}_{d\times 1}$,

(b) $\mathbb{E}_\xi \left[\left\| g\left(x, \xi, m\right) - \mathbb{E}_\xi \left[g\left(x, \xi, m\right) \right] \right\|^2 \right] \leq c_2 \eta^2 + \tilde{c_2}$,

for some positive constants $c_1, c_2, \widetilde{c_2}$ and c_3.

For the two oracles defined above, using the GS approach leads to the following constants (Balasubramanian and Ghadimi, 2018): $c_1 = \frac{L(d+3)^{\frac{3}{2}}}{2}, c_2 = \frac{L^2(d+3)^3}{2}, \widetilde{c_2} = 2(d+5)(B^2+\sigma^2)$, where σ^2 is the bound on variance of estimator $F(x, \xi)$ of $f(x)$.

4.5.2 Non-asymptotic bounds

We provide below a non-asymptotic bound for the ZRSG algorithm with the oracle (**O2'**) and non-convex objective.

Theorem 17. ***(ZRSG with the oracle (O2'))***

*Assume (**A1**) and (**A2**). With the oracle (**O2'**), suppose that the ZRSG algorithm is run with the stepsize γ_k, smoothing parameter η_k and mini-batch size m_k set as follows:*

$$\gamma_k = min\left\{ \frac{1}{L}, \frac{1}{\sqrt{dN}} \right\}, \quad \eta_k = \frac{1}{d\sqrt{N}}, \quad \textit{and} \quad m_k = dN^2, \quad \forall k \geq 1. \tag{4.14}$$

Then, for any $N \geq 1$, we have

$$\mathbb{E} \left\| \nabla f\left(x_R\right) \right\|^2 \leq \frac{2LD_f}{N} + \frac{\mathcal{Z}_5}{\sqrt{N}},$$

where $\mathcal{Z}_5 = 2\sqrt{d}D_f + 4B\mathcal{K}_3 + L\left(\frac{\sqrt{d}\mathcal{K}_3^2}{N} + \frac{c_2}{Nd^{5/2}} + \frac{\tilde{c_2}}{\sqrt{d}} \right)$, $\mathcal{K}_3 = c_1 d^{-1} + \sqrt{d}c_3$, constants

*c_1, c_2, $\tilde{c_2}$ and c_3 are as defined in (**O2'**), B is as defined in (**A2**), and D_f is as defined in* (4.5).

Proof. See Section 4.6.5. □

From the bound in Theorem 17, it is easy to see that the total number of iterations required for finding an ϵ-stationary point is at most $\mathcal{O}(\frac{d}{\epsilon^2})$. In comparison to the bound for the SP method, the $\mathcal{O}\left(N^{-1/2}\right)$ is better, and we believe that this improvement is because the variance of the gradient estimate in this oracle does not increase at the cost of bias.

Remark 5. *In comparison to the bound obtained for biased gradient without estimation error oracle (**O1'**) in Corollary 3.3 of (Ghadimi and Lan, 2013), we remark that our above bound matches their order, except that there are additional factors owing to estimation error.*

We now provide a non-asymptotic bound for the ZRSG and ZSGD algorithm for the convex objective.

Theorem 18. ***(ZRSG with the oracle (O2'))***
*Assume (**A1**) and (**A5**). With the oracle (**O2'**), suppose that the ZRSG algorithm is run with the stepsize γ_k, smoothing parameter η_k and mini-batch size m_k set as follows:*

$$\gamma_k = min\left\{\frac{1}{L}, \frac{1}{\sqrt{dN}}\right\}, \quad \eta_k = \frac{1}{d\sqrt{N}}, \quad \textit{and} \quad m_k = d^2N^2, \quad \forall k \geq 1. \tag{4.15}$$

Then, for any $N \geq 1$, we have

$$\mathbb{E}\left[f\left(x_R\right)\right] - f(x^*) \leq \frac{LD^2}{N} + \frac{\mathcal{K}_5}{\sqrt{N}}$$

*where $\mathcal{K}_5 = \sqrt{d}D^2 + 4\sqrt{d}D\left(\frac{c_1}{d} + c_3\right) + \frac{c_1^2}{d^{3/2}N} + \frac{2c_1c_3}{\sqrt{d}N} + \frac{\sqrt{d}c_3^2}{N} + \frac{c_2}{d^{5/2}N} + \frac{\tilde{c_2}}{\sqrt{d}}$, constants c_1, c_2, $\tilde{c_2}$ and c_3 are as defined in (**O2'**), and D as defined in (**A5**).*

Proof. See Section 4.6.5. □

From the bound in Theorem 18, it is easy to see that the total number of iterations required for finding an ϵ-optimal solution is at most $\mathcal{O}(\frac{d}{\epsilon^2})$. Similar to the previous case

with non-convex objective, we get a better bound of $\mathcal{O}(N^{-1/2})$ for the GS method with convex objective.

The non-asymptotic bounds similar to those in Theorems 9 and 13 for the Gaussian smoothing case with oracle (**O1'**) are derived in (Ghadimi and Lan, 2013).

Theorem 19. ***(ZSGD with the oracle (O1'))***
*Assume (**A4**) and (**A5**). With the oracle (**O1'**), suppose that the ZSGD algorithm is run with the stepsize γ_k and perturbation constant η_k set as follows:*

$$\gamma_k = \frac{C \cdot 2^{-i}}{\sqrt{dN}} \quad \textit{and} \quad \eta_k = \frac{2^{-i}}{\sqrt{d}N}, \tag{4.16}$$

when $N_i < k \leq N_{i+1}, 0 \leq i \leq l$, where $C > 0$ and N_i, l is as defined in (4.11)*. Then, for any $N \geq 4$, we have*

$$\mathbb{E}[f(x_N)] - f(x^*) \leq \frac{\mathcal{K}_6}{\sqrt{N}},$$

*where $\mathcal{K}_6 = \frac{4D^2\sqrt{d}}{C} + \frac{11CG^2}{\sqrt{d}} + \frac{24c_1D}{\sqrt{N}} + \frac{20Cc_1G}{\sqrt{d}N} + \frac{10C(dc_1^2+c_2)}{d^{3/2}N^2} + \frac{10C\tilde{c_2}}{\sqrt{d}}$, and constants c_1, c_2, $\tilde{c_2}$ are as in (**O1'**).*

Proof. See Section 4.6.5. □

From the bound in Theorem 19, it is easy to see that the total number of iterations required for finding an ϵ-optimal solution is at most $\mathcal{O}(\frac{d}{\epsilon^2})$. Further, it is interesting to note that the overall rate of $\mathcal{O}(N^{-1/2})$ obtained for the zeroth order case, with biased gradients estimated using GS method, matches with the case when unbiased gradient information is available (Jain *et al.*, 2019). Unlike (Ghadimi and Lan, 2013) where the authors provide a $\mathcal{O}(N^{-1/2})$ bound for a random iterate using the ZRSG algorithm, we provide bound for the last iterate of ZSGD. Apart from a practical preferance for using the last iterate, an advantage with our approach is that for setting the step size γ_k and smoothing parameter η_k (4.16), we do not require the knowledge of Lipschitz constant L (see (**A1**)) and $D_X := \|x_1 - x^*\|$. The latter quantity is typically unavailable in practice, as it relates to the initial error. A similar observation holds true for the non-convex case as well (see Theorem 17).

Remark 6. *Recent work in (Yousefian* et al.*, 2017) analyzed a regularized quasi-Newton algorithm for stochastic convex optimization. Specializing their non-asymptotic bound to a regularized stochastic gradient algorithm would lead to a bound of the order*

$\mathcal{O}\left(N^{-1/3}\right)$ *on the optimization error* $\mathbb{E}[f(x_N)] - f(x^*)$. *In contrast, we obtain a bound of the order* $\mathcal{O}\left(N^{-1/2}\right)$ *using Theorem 19.*

Non-asymptotic bound similar to those in Theorems 11, 12 and 16 for the Gaussian smoothing case, can be derived by using a parallel argument to the proof of simultaneous perturbation method, and we omit the details.

4.6 Convergence proofs

This section is organized as follows: In Section 4.6.1, we prove the bounds for the ZRSG algorithm with oracles **(O1)** and (**O2**). Recall that ZRSG is a gradient-based method for solving stochastic non-convex optimization problems, while **(O1)** (resp. (**O2**)) is a simultaneous perturbation-based optimization oracle that provides biased gradient information (resp. with estimation error). In Section 4.6.2, we prove the bounds for the ZRSQN algorithm with oracles (**O3**) and (**O4**). Recall that ZRSQN is a gradient/Hessian-based method for solving stochastic non-convex optimization problems, while (**O3**) (resp. (**O4**)) is a simultaneous perturbation-based optimization oracle that provides biased gradient/Hessian information (resp. with estimation error). In Section 4.6.3 (resp. 4.6.4), we prove the bounds for solving stochastic convex optimization problems using the ZRSG (resp. ZSGD) algorithm with oracles **(O1)** and (**O2**). In Section 4.6.5, we prove the bound for the ZRSG and ZSGD algorithm with oracle (**O1'**) and for the ZRSG algorithm with oracle (**O2'**). Recall that (**O1'**) (resp. (**O2'**)) is a Gaussian smoothing-based optimization oracle that provides biased gradient information (resp. with estimation error).

For the proofs, we follow the technique from (Ghadimi and Lan, 2013) for the case of stochastic non-convex optimization and from (Jain *et al.*, 2019) for the case of stochastic convex optimization. However, there are significant deviations in our proofs since we employ a biased gradient model, with/without estimation error. In particular, the analysis includes additional terms owing to the gradient bias and estimation error, in turn leading to a variation in the optimal choice for stepsizes γ_k and perturbation constant η_k, as compared to previous works. Further, the model with estimation errors has an additional batch size m_k parameter that needs to be optimized as well.

4.6.1 Proofs for Stochastic Non-Convex Optimization: ZRSG

We prove Theorem 10 first, and Theorem 9 would follow through a simple modification to the proof of Theorem 10.

Proof of Theorem 10

In the proposition below, we state and prove a general result that holds for any choice of non-increasing stepsize sequence, perturbation constants and batch sizes. Subsequently, we specialize the result for the choice of parameters suggested in Theorem 10, to prove the same.

Proposition 1. *Assume (**A1**) and (**A2**). With the oracle (**O2**), suppose that the ZRSG algorithm is run with a non-increasing stepsize sequence satisfying* $0 < \gamma_k \leq 1/L, \forall k \geq 1$ *and with the probability mass function* $P_R(\cdot)$

$$P_R(k) := Prob\{R = k\} = \frac{\gamma_k}{\sum_{i=1}^{N} \gamma_i}, \quad k = 1, \ldots, N, \tag{4.17}$$

then, for any $N \geq 1$*, we have*

$$\mathbb{E}\left[\|\nabla f(x_R)\|^2\right] \leq \frac{1}{\sum_{k=1}^{N} \gamma_k} \left[\frac{2D_f}{(2 - L\gamma_1)} + 2B \sum_{k=1}^{N} \left(c_1 \eta_k^2 + \frac{c_3}{\eta_k \sqrt{m_k}} \right) \left(\frac{\gamma_k + L\gamma_k^2}{2 - L\gamma_k} \right) + L \sum_{k=1}^{N} \frac{\gamma_k^2}{(2 - L\gamma_k)} \left[d c_1^2 \eta_k^4 + 2 d c_1 c_3 \frac{\eta_k}{\sqrt{m_k}} + \frac{d c_3^2}{\eta_k^2 m_k} + \frac{c_2}{\eta_k^2} \right] \right], \tag{4.18}$$

where c_1*,* c_2 *and* c_3 *are as defined in (**O2**),* B *is as defined in (**A2**), and* D_f *as defined in* (4.5).

Proof. We use the technique from (Ghadimi and Lan, 2013). However, our proof involves significant deviations owing to the fact that the simultaneous perturbation method has a variance in gradient estimates that scales inversely with perturbation constant η_k, and this is unlike the Gaussian smoothing approach, where such an inverse scaling is absent (instead, the variance scales directly with η_k). Further, the model with estimation errors has an additional batch size m_k parameter that needs to be optimized as well.

First, notice that

$$\begin{aligned}\|x_{k+1}-x_k\| &= \|\Pi_{\mathcal{W}}\left(x_k-\gamma_k g(x_k,\xi_k,m_k)\right)-x_k\| \\ &\leq \|x_k-\gamma_k g\left(x_k,\xi_k,m_k\right)-x_k\| \\ &= -\gamma_k\|g\left(x_k,\xi_k,m_k\right)\|. \end{aligned} \tag{4.19}$$

The inequality in (4.19) holds because x_k is already in the convex set $\mathcal{W}$ and $\Pi_{\mathcal{W}}$ is a non-expansive projection operator. Further notice that

$$\begin{aligned}\mathbb{E}_{\xi_{[k]}}\left[g\left(x_k,\xi_k,m_k\right)\right] &= \mathbb{E}_{\xi_k}\left[g\left(x_k,\xi_k,m_k\right)|\xi_{[k-1]}\right] = \mathbb{E}_{\xi_k}\left[g\left(x_k,\xi_k,m_k\right)|x_k\right] \\ &\leq \nabla f\left(x_k\right)+c_1\eta_k^2\mathbf{1}_{d\times 1}+\frac{c_3}{\eta_k\sqrt{m_k}}\mathbf{1}_{d\times 1}, \end{aligned} \tag{4.20}$$

and

$$\mathbb{E}_{\xi_{[k]}}\left[\|g\left(x_k,\xi_k,m_k\right)\|^2\right] \leq \left\|\mathbb{E}_{\xi_{[k]}}\left[g\left(x_k,\xi_k,m_k\right)\right]\right\|^2+c_2/\eta_k^2. \tag{4.21}$$

Now, under assumption (**A1**), we have

$$\begin{aligned} f\left(x_{k+1}\right) &\leq f\left(x_k\right)+\left\langle\nabla f\left(x_k\right),x_{k+1}-x_k\right\rangle+\frac{L}{2}\left\|x_{k+1}-x_k\right\|^2 \\ &= f\left(x_k\right)-\gamma_k\left\langle\nabla f\left(x_k\right),g\left(x_k,\xi_k,m_k\right)\right\rangle+\frac{L}{2}\gamma_k^2\left\|g\left(x_k,\xi_k,m_k\right)\right\|^2. \end{aligned} \tag{4.22}$$

Taking expectations with respect to $\xi_{[k]}$ on both sides of (4.22) and using (4.20) and (4.21), we obtain

$$\begin{aligned} &\mathbb{E}_{\xi_{[k]}}\left[f\left(x_{k+1}\right)\right] \\ &\leq \mathbb{E}_{\xi_{[k]}}\left[f\left(x_k\right)\right]-\gamma_k\left\langle\nabla f\left(x_k\right),\nabla f\left(x_k\right)+c_1\eta_k^2\mathbf{1}_{d\times 1}+\frac{c_3}{\eta_k\sqrt{m_k}}\mathbf{1}_{d\times 1}\right\rangle \\ &\quad+\frac{L}{2}\gamma_k^2\left[\left\|\mathbb{E}_{\xi_{[k]}}\left[g\left(x_k,\xi_k,m_k\right)\right]\right\|^2+\frac{c_2}{\eta_k^2}\right] \\ &\leq f\left(x_k\right)-\gamma_k\left\|\nabla f\left(x_k\right)\right\|^2+\left(c_1\eta_k^2+\frac{c_3}{\eta_k\sqrt{m_k}}\right)\gamma_k\mathbb{E}_{\xi_{[k]}}\|\nabla f\left(x_k\right)\|_1 \\ &\quad+\frac{L}{2}\gamma_k^2\Bigg[\left\|\nabla f\left(x_k\right)\right\|^2+2\left(c_1\eta_k^2+\frac{c_3}{\eta_k\sqrt{m_k}}\right)\mathbb{E}_{\xi_{[k]}}\|\nabla f\left(x_k\right)\|_1 \\ &\quad+\left(\sqrt{d}c_1\eta_k^2+\frac{\sqrt{d}c_3}{\eta_k\sqrt{m_k}}\right)^2+\frac{c_2}{\eta_k^2}\Bigg] \end{aligned} \tag{4.23}$$

$$\leq f\left(x_k\right) - \left(\gamma_k - \frac{L}{2}\gamma_k^2\right)\left\|\nabla f\left(x_k\right)\right\|^2 + \left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)\left(\gamma_k + L\gamma_k^2\right)B$$
$$+ \frac{L}{2}\gamma_k^2\left[\left(\sqrt{d}c_1\eta_k^2 + \frac{\sqrt{d}c_3}{\eta_k\sqrt{m_k}}\right)^2 + \frac{c_2}{\eta_k^2}\right],$$

where we have used the fact that $-\|X\|_1 \leq \sum_{i=1}^{d} x_i$ for any vector X in arriving at the inequality (4.23) and the last inequality follows from the fact that $\|\nabla f\left(x_k\right)\|_1 \leq B$. Re-arranging the terms, we obtain

$$\left(\gamma_k - \frac{L}{2}\gamma_k^2\right)\left\|\nabla f\left(x_k\right)\right\|^2 = f\left(x_k\right) - \mathbb{E}_{\xi_k} f\left(x_{k+1}\right)$$
$$+ \left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)\left(\gamma_k + L\gamma_k^2\right)B$$
$$+ \frac{L}{2}\gamma_k^2\left[\left(\sqrt{d}c_1\eta_k^2 + \frac{\sqrt{d}c_3}{\eta_k\sqrt{m_k}}\right)^2 + \frac{c_2}{\eta_k^2}\right]$$
$$\gamma_k\left\|\nabla f\left(x_k\right)\right\|^2 = \frac{2}{\left(2 - L\gamma_k\right)}\Bigg[f\left(x_k\right) - \mathbb{E}_{\xi_k} f\left(x_{k+1}\right)$$
$$+ \left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)\left(\gamma_k + L\gamma_k^2\right)B\Bigg]$$
$$+ \frac{L\gamma_k^2}{\left(2 - L\gamma_k\right)}\left[dc_1^2\eta_k^4 + 2dc_1c_3\frac{\eta_k}{\sqrt{m_k}} + \frac{dc_3^2}{\eta_k^2 m_k} + \frac{c_2}{\eta_k^2}\right].$$

Now, summing up the inequality above over $k = 1$ to N, and taking expectations, we obtain

$$\sum_{k=1}^{N}\gamma_k\mathbb{E}_{\xi_{[N]}}\left\|\nabla f\left(x_k\right)\right\|^2$$
$$\leq 2\sum_{k=1}^{N}\frac{\left(\mathbb{E}_{\xi_{[N]}} f\left(x_k\right) - \mathbb{E}_{\xi_{[N]}} f\left(x_{k+1}\right)\right)}{\left(2 - L\gamma_k\right)}$$
$$+ 2\sum_{k=1}^{N}\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)\left(\frac{\gamma_k + L\gamma_k^2}{2 - L\gamma_k}\right)B$$
$$+ L\sum_{k=1}^{N}\frac{\gamma_k^2}{\left(2 - L\gamma_k\right)}\left[dc_1^2\eta_k^4 + 2dc_1c_3\frac{\eta_k}{\sqrt{m_k}} + \frac{dc_3^2}{\eta_k^2 m_k} + \frac{c_2}{\eta_k^2}\right]$$
$$= 2\left[\frac{f\left(x_1\right)}{\left(2 - L\gamma_1\right)} - \sum_{k=2}^{N}\left(\frac{1}{\left(2 - L\gamma_{k-1}\right)} - \frac{1}{\left(2 - L\gamma_k\right)}\right)\mathbb{E}_{\xi_{[N]}} f\left(x_k\right) - \frac{\mathbb{E}_{\xi_{[N]}} f\left(x_{N+1}\right)}{\left(2 - L\gamma_N\right)}\right]$$
$$+ 2\sum_{k=1}^{N}\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)\left(\frac{\gamma_k + L\gamma_k^2}{2 - L\gamma_k}\right)B$$

$$+ L \sum_{k=1}^{N} \frac{\gamma_k^2}{(2 - L\gamma_k)} \left[dc_1^2 \eta_k^4 + 2dc_1 c_3 \frac{\eta_k}{\sqrt{m_k}} + \frac{dc_3^2}{\eta_k^2 m_k} + \frac{c_2}{\eta_k^2} \right].$$

Noting that $\mathbb{E}_{\xi_{[N]}} [f(x_k)] \geq f(x^*)$, we obtain

$$\begin{aligned}
&\sum_{k=1}^{N} \gamma_k \mathbb{E}_{\xi_{[N]}} \|\nabla f(x_k)\|^2 \\
&\leq 2 \left[\frac{f(x_1)}{(2 - L\gamma_1)} - f(x^*) \sum_{k=2}^{N} \left(\frac{1}{(2 - L\gamma_{k-1})} - \frac{1}{(2 - L\gamma_k)} \right) - \frac{f(x^*)}{(2 - L\gamma_N)} \right] \\
&\quad + 2 \sum_{k=1}^{N} \left(c_1 \eta_k^2 + \frac{c_3}{\eta_k \sqrt{m_k}} \right) \left(\frac{\gamma_k + L\gamma_k^2}{2 - L\gamma_k} \right) B \\
&\quad + L \sum_{k=1}^{N} \frac{\gamma_k^2}{(2 - L\gamma_k)} \left[dc_1^2 \eta_k^4 + 2dc_1 c_3 \frac{\eta_k}{\sqrt{m_k}} + \frac{dc_3^2}{\eta_k^2 m_k} + \frac{c_2}{\eta_k^2} \right] \\
&\leq \frac{2(f(x_1) - f(x^*))}{(2 - L\gamma_1)} + 2 \sum_{k=1}^{N} \left(c_1 \eta_k^2 + \frac{c_3}{\eta_k \sqrt{m_k}} \right) \left(\frac{\gamma_k + L\gamma_k^2}{2 - L\gamma_k} \right) B \\
&\quad + L \sum_{k=1}^{N} \frac{\gamma_k^2}{(2 - L\gamma_k)} \left[dc_1^2 \eta_k^4 + 2dc_1 c_3 \frac{\eta_k}{\sqrt{m_k}} + \frac{dc_3^2}{\eta_k^2 m_k} + \frac{c_2}{\eta_k^2} \right].
\end{aligned}$$

The last inequality follows from the fact that $\left(\frac{1}{(2 - L\gamma_{k-1})} - \frac{1}{(2 - L\gamma_k)} \right) \geq 0$. The bound in (4.18) follows by using the distribution of R (specified in (4.17)) in the RHS above.

□

We now specialize the result obtained in the proposition above, to derive a non-asymptotic bound for ZRSG with gradients estimated by the SP method with function estimation error.

Proof. ***(Theorem 10 (i))***

Recall that the stepsize γ_k, perturbation constant η_k and mini-batch size m_k are defined as follows:

$$\gamma_k = min\left\{ \frac{1}{L}, \frac{1}{(d^2 N)^{2/3}} \right\}, \qquad \eta_k = \frac{1}{(d^5 N)^{1/6}}, \qquad \text{and} \qquad m_k = N, \qquad \forall k \geq 1. \tag{4.24}$$

Combining (4.17) with (4.18), we obtain

$$\mathbb{E}\left[\|\nabla f(x_R)\|^2 \right]$$

$$
\begin{aligned}
&\leq \frac{1}{\sum_{k=1}^{N}\gamma_k}\left[\frac{2\left(f(x_1)-f(x^*)\right)}{(2-L\gamma_1)}+2B\sum_{k=1}^{N}\left(c_1\eta_k^2+\frac{c_3}{\eta_k\sqrt{m_k}}\right)\left(\frac{\gamma_k+L\gamma_k^2}{2-L\gamma_k}\right)\right.\\
&\qquad\left.+L\sum_{k=1}^{N}\frac{\gamma_k^2}{(2-L\gamma_k)}\left[dc_1^2\eta_k^4+2dc_1c_3\frac{\eta_k}{\sqrt{m_k}}+\frac{dc_3^2}{\eta_k^2m_k}+\frac{c_2}{\eta_k^2}\right]\right]\\
&=\frac{1}{N\gamma}\left[\frac{2\left(f(x_1)-f(x^*)\right)}{(2-L\gamma)}+2BN\left(c_1\eta^2+\frac{c_3}{\eta\sqrt{m}}\right)\left(\frac{\gamma+L\gamma^2}{2-L\gamma}\right)\right.\\
&\qquad\left.+\frac{LN\gamma^2}{(2-L\gamma)}\left[dc_1^2\eta^4+2dc_1c_3\frac{\eta}{\sqrt{m}}+\frac{dc_3^2}{\eta^2m}+\frac{c_2}{\eta^2}\right]\right]\\
&\leq\frac{1}{N\gamma}\left[2\left(f(x_1)-f(x^*)\right)+4N\gamma B\left(c_1\eta^2+\frac{c_3}{\eta\sqrt{m}}\right)\right.\\
&\quad\left.+LN\gamma^2\left[dc_1^2\eta^4+2dc_1c_3\frac{\eta}{\sqrt{m}}+\frac{dc_3^2}{\eta^2m}+\frac{c_2}{\eta^2}\right]\right] \qquad (4.25)\\
&=\frac{2\left(f(x_1)-f(x^*)\right)}{N\gamma}+4B\left(c_1\eta^2+\frac{c_3}{\eta\sqrt{m}}\right)\\
&\quad+L\gamma\left[dc_1^2\eta^4+2dc_1c_3\frac{\eta}{\sqrt{m}}+\frac{dc_3^2}{\eta^2m}+\frac{c_2}{\eta^2}\right]\\
&\leq\frac{2\left(f(x_1)-f(x^*)\right)}{N}max\left\{L,(d^2N)^{2/3}\right\}+4B\left(\frac{c_1}{(d^5N)^{1/3}}+\frac{c_3d^{5/6}}{N^{1/3}}\right)\\
&\quad+L\left[\frac{dc_1^2}{(d^5N)^{2/3}}+2dc_1c_3\frac{1}{d^{5/6}N^{2/3}}+\frac{dd^{5/6}c_3^2}{N^{2/3}}+\frac{d^{5/3}c_2}{N^{-1/3}}\right]\frac{1}{(d^2N)^{2/3}} \qquad (4.26)\\
&=\frac{2L\left(f(x_1)-f(x^*)\right)}{N}+\frac{2d^{4/3}\left(f(x_1)-f(x^*)\right)}{N^{1/3}}+4B\left(\frac{c_1}{(d^5N)^{1/3}}+\frac{c_3d^{5/6}}{N^{1/3}}\right)\\
&\quad+L\left[\frac{c_1^2}{d^{7/3}N^{2/3}}+2c_1c_3\frac{d^{1/6}}{N^{2/3}}+\frac{d^{11/6}c_3^2}{N^{2/3}}+\frac{d^{5/3}c_2}{N^{-1/3}}\right]\frac{1}{(d^2N)^{2/3}}\\
&=\frac{2L\left(f(x_1)-f(x^*)\right)}{N}+\frac{1}{N^{1/3}}\left[2d^{4/3}\left(f(x_1)-f(x^*)\right)+4B\left(\frac{c_1}{d^{5/3}}+c_3d^{5/6}\right)\right.\\
&\quad\left.+\frac{Lc_1^2}{d^{11/3}N}+2Lc_1c_3\frac{1}{d^{7/6}N}+\frac{Lc_3^2d^{1/2}}{N}+Lc_2d^{1/3}\right].
\end{aligned}
$$

In the above, inequality (4.25) follows by using the fact that $\gamma \leq 1/L$, and the inequality (4.26) follows by using the definition of γ, η and m. □

Now, we specialize the result in (4.18) for increasing batch size i.e., $m_k = k^\beta$ for some constant $\beta > 0$.

Proof. ***(Theorem 10 (ii))***

Recall the stepsize γ_k and perturbation constant η_k from equation (4.24). Combining (4.17) with (4.18), we obtain

$$\mathbb{E}\left[\|\nabla f\left(x_R\right)\|^2\right]$$

$$
\begin{aligned}
&\leq \frac{1}{\sum_{k=1}^{N}\gamma_k}\left[\frac{2\left(f(x_1)-f(x^*)\right)}{(2-L\gamma_1)}+2B\sum_{k=1}^{N}\left(c_1\eta_k^2+\frac{c_3}{\eta_k\sqrt{m_k}}\right)\left(\frac{\gamma_k+L\gamma_k^2}{2-L\gamma_k}\right)\right.\\
&\qquad\left.+L\sum_{k=1}^{N}\frac{\gamma_k^2}{(2-L\gamma_k)}\left[dc_1^2\eta_k^4+2dc_1c_3\frac{\eta_k}{\sqrt{m_k}}+\frac{dc_3^2}{\eta_k^2m_k}+\frac{c_2}{\eta_k^2}\right]\right]\\
&=\frac{1}{N\gamma}\left[\frac{2\left(f(x_1)-f(x^*)\right)}{(2-L\gamma)}+2B\sum_{k=1}^{N}\left(c_1\eta^2+\frac{c_3}{\eta\sqrt{k^\beta}}\right)\left(\frac{\gamma+L\gamma^2}{2-L\gamma}\right)\right.\\
&\quad\left.+L\sum_{k=1}^{N}\frac{\gamma^2}{(2-L\gamma)}\left[dc_1^2\eta^4+2dc_1c_3\frac{\eta}{\sqrt{k^\beta}}+\frac{dc_3^2}{\eta^2k^\beta}+\frac{c_2}{\eta^2}\right]\right]\\
&\leq\frac{1}{N\gamma}\left[2\left(f(x_1)-f(x^*)\right)+4N\gamma Bc_1\eta^2+\frac{4\gamma Bc_3}{\eta}\sum_{k=1}^{N}k^{-\frac{\beta}{2}}+LN\gamma^2\left[dc_1^2\eta^4+\frac{c_2}{\eta^2}\right]\right.\\
&\quad\left.+2Ld\gamma^2c_1c_3\eta\sum_{k=1}^{N}k^{-\frac{\beta}{2}}+\frac{Ld\gamma^2c_3^2}{\eta^2}\sum_{k=1}^{N}k^{-\beta}\right] \qquad (4.27)\\
&\leq\frac{2\left(f(x_1)-f(x^*)\right)}{N\gamma}+4Bc_1\eta^2+\frac{4Bc_3}{N\eta}\int_0^N x^{-\frac{\beta}{2}}dx+L\gamma\left[dc_1^2\eta^4+\frac{c_2}{\eta^2}\right]\\
&\quad+\frac{2Ld\gamma c_1c_3\eta}{N}\int_0^N x^{-\frac{\beta}{2}}dx+\frac{Ld\gamma c_3^2}{N\eta^2}\int_0^N x^{-\beta}dx\\
&=\frac{2\left(f(x_1)-f(x^*)\right)}{N\gamma}+4Bc_1\eta^2+\frac{4Bc_3}{N\eta}\left(\frac{N^{-\frac{\beta}{2}+1}}{-\frac{\beta}{2}+1}\right)+L\gamma\left[dc_1^2\eta^4+\frac{c_2}{\eta^2}\right]\\
&\quad+\frac{2Ld\gamma c_1c_3\eta}{N}\left(\frac{N^{-\frac{\beta}{2}+1}}{-\frac{\beta}{2}+1}\right)+\frac{Ld\gamma c_3^2}{N\eta^2}\left(\frac{N^{-\beta+1}}{-\beta+1}\right)\\
&=\frac{2\left(f(x_1)-f(x^*)\right)}{N\gamma}+4Bc_1\eta^2+\frac{4Bc_3N^{-\frac{\beta}{2}}}{\eta\left(-\frac{\beta}{2}+1\right)}+L\gamma\left[dc_1^2\eta^4+\frac{c_2}{\eta^2}\right]\\
&\quad+\frac{2Ld\gamma c_1c_3\eta N^{-\frac{\beta}{2}}}{\left(-\frac{\beta}{2}+1\right)}+\frac{Ld\gamma c_3^2N^{-\beta}}{\eta^2\left(-\beta+1\right)}\\
&\leq\frac{2\left(f(x_1)-f(x^*)\right)}{N}max\left\{L,(d^2N)^{2/3}\right\}+\frac{4Bc_1}{(d^5N)^{1/3}}+\frac{4Bc_3N^{-\frac{\beta}{2}}(d^5N)^{1/6}}{\left(-\frac{\beta}{2}+1\right)}\\
&\quad+\frac{L}{(d^2N)^{2/3}}\left[\frac{dc_1^2}{(d^5N)^{2/3}}+c_2(d^5N)^{1/3}+\frac{2dc_1c_3}{(d^5N)^{1/6}N^{\frac{\beta}{2}}\left(-\frac{\beta}{2}+1\right)}+\frac{dc_3^2(d^5N)^{1/3}}{N^\beta\left(-\beta+1\right)}\right]\\
&\qquad\qquad (4.28)\\
&=\frac{2L\left(f(x_1)-f(x^*)\right)}{N}+\frac{2d^{4/3}\left(f(x_1)-f(x^*)\right)}{N^{1/3}}+\frac{4Bc_1}{(d^5N)^{1/3}}+\frac{4Bc_3d^{5/6}}{N^{\frac{3\beta-1}{6}}\left(-\frac{\beta}{2}+1\right)}\\
&\quad+\frac{L}{(d^2N)^{2/3}}\left[\frac{c_1^2}{N^{2/3}d^{7/3}}+c_2(d^5N)^{1/3}+\frac{2c_1c_3d^{1/6}}{N^{\frac{3\beta+1}{6}}\left(-\frac{\beta}{2}+1\right)}+\frac{c_3^2d^{8/3}}{N^{\frac{3\beta-1}{3}}\left(-\beta+1\right)}\right]\\
&=\frac{2L\left(f(x_1)-f(x^*)\right)}{N}+\frac{1}{N^{1/3}}\left[2d^{4/3}\left(f(x_1)-f(x^*)\right)+\frac{4Bc_1}{d^{5/3}}+\frac{4Bc_3d^{5/6}}{N^{\frac{3\beta-1}{6}}\left(-\frac{\beta}{2}+1\right)}\right.\\
&\quad\left.+\frac{Lc_1^2}{Nd^{11/3}}+Lc_2d^{1/3}+\frac{2Lc_1c_3}{d^{7/6}N^{\frac{3\beta+5}{6}}\left(-\frac{\beta}{2}+1\right)}+\frac{Lc_3^2d^{4/3}}{N^{\frac{3\beta+1}{3}}\left(-\beta+1\right)}\right].
\end{aligned}
$$

In the above, inequality (4.27) follows by using the fact that $\gamma \leq 1/L$, and the inequality (4.28) follows by using the definition of γ, η and m. □

Proof of Theorem 9

Proof. ***(Theorem 9)***
Proof follows in a similar manner as that of Theorem 10(i) in Section 4.6.1 after setting $m_k = \infty, \forall k \geq 1$ or $c_3 = 0$. □

4.6.2 Proofs for Stochastic Non-Convex Optimization: ZRSQN

We prove Theorem 12 first, and Theorem 11 would follow through a simple modification to the proof of Theorem 12.

Proof of Theorem 12

In the proposition below, we state and prove a general result that holds for any choice of non-increasing stepsize sequence, perturbation constants and batch sizes. Subsequently, we specialize the result for the choice of parameters suggested in Theorem 12, to prove the same.

(Wang *et al.*, 2017) has a result in Theorem 2.4 for an unbiased gradient/Hessian oracle, however, our proof involves significant deviations owing to the fact that we employ biased gradient/Hessian oracle. Further, the simultaneous perturbation method has a variance in gradient estimates that scales inversely with perturbation constant η_k.

Proposition 2. *Assume* ***(A2)*** *and* ***(A3)****. With the oracle* ***(O4)****, suppose that the ZRSQN algorithm is run with a non-increasing stepsize sequence satisfying* $0 < \gamma_k \leq \frac{2C_l - 1}{\Lambda C_u^2}, \forall k \geq 1$ *and with the probability mass function* $P_R(\cdot)$ *as defined in* (4.17)*, then, for any* $N \geq 1$*, we have*

$$\mathbb{E}\left[\|\nabla f(x_R)\|^2\right]$$
$$\leq \frac{1}{\sum_{k=1}^{N} \gamma_k} \left[\frac{2D_f}{(2C_l - \Lambda \gamma_1 C_u^2)} \right.$$

$$+2B\sum_{k=1}^{N}\frac{\gamma_k\left[Bc_1'\eta_k^2+\frac{Bc_3}{\eta_k\sqrt{m_k}}+\left(c_1\eta_k^2+\frac{c_3}{\eta_k\sqrt{m_k}}\right)\left(C_l+dc_1'\eta_k^2+\frac{dc_3}{\eta_k\sqrt{m_k}}+\Lambda C_u^2\gamma_k\right)\right]}{(2C_l-\Lambda\gamma_k C_u^2)}$$
$$+\sum_{k=1}^{N}\frac{\Lambda C_u^2\gamma_k^2}{(2C_l-\Lambda\gamma_k C_u^2)}\left[d\left(c_1\eta_k^2+\frac{c_3}{\eta_k\sqrt{m_k}}\right)^2+\frac{c_2}{\eta_k^2}\right], \tag{4.29}$$

*where constants c_1, c_1', c_2 and c_3 are as defined in (**O4**), Λ, C_l, C_u is as defined in (**A3**), B is as defined in (**A2**), and D_f as defined in* (4.5)*.*

Proof. First, notice that

$$\begin{aligned}\|x_{k+1}-x_k\| &= \|\Pi_{\mathcal{W}}\left(x_k-\gamma_k H\left(x_k,\xi_k,m_k\right)g(x_k,\xi_k,m_k)\right)-x_k\|\\ &\leq \|x_k-\gamma_k H\left(x_k,\xi_k,m_k\right)g\left(x_k,\xi_k,m_k\right)-x_k\| \qquad (4.30)\\ &= -\gamma_k\|H\left(x_k,\xi_k,m_k\right)g\left(x_k,\xi_k,m_k\right)\|.\end{aligned}$$

The inequality in (4.30) holds because x_k is already in the convex set $\mathcal{W}$ and $\Pi_{\mathcal{W}}$ is a non-expansive projection operator. Now, under assumption (**A3**), we have

$$\begin{aligned}&f\left(x_{k+1}\right)\\ &\leq f\left(x_k\right)+\left\langle\nabla f\left(x_k\right),x_{k+1}-x_k\right\rangle+\frac{1}{2}\langle(x_{k+1}-x_k),\nabla^2 f(x_k)(x_{k+1}-x_k)\rangle\\ &\leq f\left(x_k\right)+\left\langle\nabla f\left(x_k\right),x_{k+1}-x_k\right\rangle+\frac{\Lambda}{2}\left\|x_{k+1}-x_k\right\|^2\\ &= f\left(x_k\right)-\gamma_k\left\langle\nabla f\left(x_k\right),H(x_k,\xi_k,m_k)g(x_k,\xi_k,m_k)\right\rangle\\ &\quad+\frac{\Lambda}{2}\gamma_k^2\left\|H(x_k,\xi_k,m_k)g(x_k,\xi_k,m_k)\right\|^2\\ &\leq f\left(x_k\right)-\gamma_k\left\langle\nabla f\left(x_k\right),H(x_k,\xi_k,m_k)g(x_k,\xi_k,m_k)\right\rangle+\frac{\Lambda}{2}\gamma_k^2C_u^2\left\|g(x_k,\xi_k,m_k)\right\|^2.\end{aligned} \tag{4.31}$$

Taking expectations with respect to $\xi_{[k]}$ on both sides of (4.31) and using (4.20) and (4.21), we obtain

$$\begin{aligned}&\mathbb{E}_{\xi_{[k]}}\left[f\left(x_{k+1}\right)\right]\\ &\leq \mathbb{E}_{\xi_{[k]}}\left[f\left(x_k\right)\right]-\gamma_k\left\langle\mathbb{E}_{\xi_{[k]}}[\nabla f\left(x_k\right)],\mathbb{E}_{\xi_{[k]}}\left[H(x_k,\xi_k,m_k)g(x_k,\xi_k,m_k)\right]\right\rangle\\ &\quad+\frac{\Lambda}{2}C_u^2\gamma_k^2\left[\left\|\mathbb{E}_{\xi_{[k]}}\left[g\left(x_k,\xi_k,m_k\right)\right]\right\|^2+\frac{c_2}{\eta_k^2}\right]\\ &\leq \mathbb{E}_{\xi_{[k]}}\left[f\left(x_k\right)\right]-\gamma_k\left\langle\mathbb{E}_{\xi_{[k]}}[\nabla f\left(x_k\right)],\mathbb{E}_{\xi_{[k]}}\left[H(x_k,\xi_k,m_k)g(x_k,\xi_k,m_k)\right]\right\rangle\end{aligned}$$

$$+\frac{\Lambda}{2}C_u^2\gamma_k^2\Bigg[\left\|\nabla f\left(x_k\right)\right\|^2+2\left(c_1\eta_k^2+\frac{c_3}{\eta_k\sqrt{m_k}}\right)\mathbb{E}_{\xi_{[k]}}\|\nabla f\left(x_k\right)\|_1$$
$$+d\left(c_1\eta_k^2+\frac{c_3}{\eta_k\sqrt{m_k}}\right)^2+\frac{c_2}{\eta_k^2}\Bigg].$$

Noting that we make calls to the oracle (**O4**), to obtain $H(x_k,\xi_k,m_k)$ and $g(x_k,\xi_k,m_k)$, and assuming independence between them, we have $E_{\xi_k}[H(x_k,\xi_k,m_k)g(x_k,\xi_k,m_k)|\xi_{[k-1]}] = E_{\xi_k}[H(x_k,\xi_k,m_k)|\xi_{[k-1]}]E_{\xi_k}[g(x_k,\xi_k,m_k)|\xi_{[k-1]}] \leq (H(x_k) + c_1^{'}\eta_k^2\mathbf{1}_{d\times d} + \frac{c_3}{\eta_k\sqrt{m_k}}\mathbf{1}_{d\times d})(\nabla f(x_k) + c_1\eta_k^2\mathbf{1}_{d\times 1} + \frac{c_3}{\eta_k\sqrt{m_k}}\mathbf{1}_{d\times 1}) = (H(x_k) + c_1^{'}\eta_k^2\mathbf{1}_{d\times d} + \frac{c_3}{\eta_k\sqrt{m_k}}\mathbf{1}_{d\times d})\nabla f(x_k) + (H(x_k) + c_1^{'}\eta_k^2\mathbf{1}_{d\times d} + \frac{c_3}{\eta_k\sqrt{m_k}}\mathbf{1}_{d\times d})(c_1\eta_k^2\mathbf{1}_{d\times 1} + \frac{c_3}{\eta_k\sqrt{m_k}}\mathbf{1}_{d\times 1})$. Plugging this equality in the equation above and noting (**A3**), we obtain

$$\begin{aligned}
&\mathbb{E}_{\xi_{[k]}}\left[f\left(x_{k+1}\right)\right]\\
&\leq \mathbb{E}_{\xi_{[k]}}\left[f\left(x_k\right)\right]-\gamma_k C_l\mathbb{E}_{\xi_{[k]}}\left\|\nabla f\left(x_k\right)\right\|^2+\left(c_1^{'}\eta_k^2+\frac{c_3}{\eta_k\sqrt{m_k}}\right)\gamma_k\mathbb{E}_{\xi_{[k]}}\left[\|\nabla f\left(x_k\right)\|_1^2\right]\\
&\quad+\left(c_1\eta_k^2+\frac{c_3}{\eta_k\sqrt{m_k}}\right)\left(C_l+dc_1^{'}\eta_k^2+\frac{dc_3}{\eta_k\sqrt{m_k}}\right)\gamma_k\mathbb{E}_{\xi_{[k]}}\left[\|\nabla f\left(x_k\right)\|_1\right]\\
&\quad+\frac{\Lambda}{2}C_u^2\gamma_k^2\Bigg[\left\|\nabla f\left(x_k\right)\right\|^2+2\left(c_1\eta_k^2+\frac{c_3}{\eta_k\sqrt{m_k}}\right)\mathbb{E}_{\xi_{[k]}}\|\nabla f\left(x_k\right)\|_1\\
&\quad+d\left(c_1\eta_k^2+\frac{c_3}{\eta_k\sqrt{m_k}}\right)^2+\frac{c_2}{\eta_k^2}\Bigg],
\end{aligned}$$

where we have used the fact that $-\|X\|_1 \leq \sum_{i=1}^d x_i$ for any vector X. Let $\Delta H_k = c_1^{'}\eta_k^2+\frac{c_3}{\eta_k\sqrt{m_k}}$ and $\Delta g_k = c_1\eta_k^2+\frac{c_3}{\eta_k\sqrt{m_k}}$, then using the fact that $\|\nabla f\left(x_k\right)\|_1 \leq B$, we have

$$\begin{aligned}
&\mathbb{E}_{\xi_{[k]}}\left[f\left(x_{k+1}\right)\right]\\
&\leq \mathbb{E}_{\xi_{[k]}}\left[f\left(x_k\right)\right]-\gamma_k C_l\mathbb{E}_{\xi_{[k]}}\left\|\nabla f\left(x_{[k]}\right)\right\|^2+\Delta H_k\gamma_k B^2+\Delta g_k\left(C_l+d\Delta H_k\right)\gamma_k B\\
&\quad+\frac{\Lambda}{2}C_u^2\gamma_k^2\left[\left\|\nabla f\left(x_k\right)\right\|^2+2\Delta g_k B+d\Delta g_k^2+\frac{c_2}{\eta_k^2}\right]\\
&=f\left(x_k\right)-\left(\gamma_k C_l-\frac{\Lambda C_u^2\gamma_k^2}{2}\right)\left\|\nabla f\left(x_k\right)\right\|^2\\
&\quad+\gamma_k B\left(\Delta H_k B+\Delta g_k(C_l+d\Delta H_k+\Lambda C_u^2\gamma_k)\right)+\frac{\Lambda}{2}C_u^2\gamma_k^2\left[d\Delta g_k^2+\frac{c_2}{\eta_k^2}\right].
\end{aligned}$$

Re-arranging the terms, we obtain

$$\left(\gamma_k C_l-\frac{\Lambda C_u^2}{2}\gamma_k^2\right)\left\|\nabla f\left(x_k\right)\right\|^2$$

$$
\begin{aligned}
&\leq f\left(x_k\right) - \mathbb{E}_{\xi_k}\left[f\left(x_{k+1}\right)\right] + \gamma_k B\left(\Delta H_k B + \Delta g_k(C_l + d\Delta H_k + \Lambda C_u^2 \gamma_k)\right) \\
&+ \frac{\Lambda}{2} C_u^2 \gamma_k^2 \left[d\Delta g_k^2 + \frac{c_2}{\eta_k^2}\right] \\
&\gamma_k \left\|\nabla f\left(x_k\right)\right\|^2 \\
&\leq \frac{2}{\left(2C_l - \Lambda\gamma_k C_u^2\right)}\left[f\left(x_k\right) - \mathbb{E}_{\xi_k}\left[f\left(x_{k+1}\right)\right]\right] \\
&+ \frac{2\gamma_k B\left(\Delta H_k B + \Delta g_k(C_l + d\Delta H_k + \Lambda C_u^2\gamma_k)\right)}{\left(2C_l - \Lambda\gamma_k C_u^2\right)} + \frac{\Lambda C_u^2 \gamma_k^2}{\left(2C_l - \Lambda\gamma_k C_u^2\right)}\left[d\Delta g_k^2 + \frac{c_2}{\eta_k^2}\right].
\end{aligned}
$$

Now, summing up the inequality above over $k = 1$ to N, and taking expectations, we obtain

$$
\begin{aligned}
&\sum_{k=1}^{N} \gamma_k \mathbb{E}_{\xi_{[N]}} \left\|\nabla f\left(x_k\right)\right\|^2 \\
&\leq 2\sum_{k=1}^{N} \frac{\left(\mathbb{E}_{\xi_{[N]}}\left[f\left(x_k\right)\right] - \mathbb{E}_{\xi_{[N]}}\left[f\left(x_{k+1}\right)\right]\right)}{\left(2C_l - \Lambda\gamma_k C_u^2\right)} \\
&\quad + \sum_{k=1}^{N} \frac{2\gamma_k B\left(\Delta H_k B + \Delta g_k(C_l + d\Delta H_k + \Lambda C_u^2\gamma_k)\right)}{\left(2C_l - \Lambda\gamma_k C_u^2\right)} \\
&\quad + \sum_{k=1}^{N} \frac{\Lambda C_u^2 \gamma_k^2}{\left(2C_l - \Lambda\gamma_k C_u^2\right)}\left[d\Delta g_k^2 + \frac{c_2}{\eta_k^2}\right] \\
&= 2\Bigg[\frac{f\left(x_1\right)}{\left(2C_l - \Lambda\gamma_1 C_u^2\right)} - \sum_{k=2}^{N}\left(\frac{\mathbb{E}_{\xi_{[N]}} f\left(x_k\right)}{\left(2C_l - \Lambda\gamma_{k-1} C_u^2\right)} - \frac{\mathbb{E}_{\xi_{[N]}} f\left(x_k\right)}{\left(2C_l - \Lambda\gamma_k C_u^2\right)}\right) \\
&\quad - \frac{\mathbb{E}_{\xi_{[N]}}\left[f\left(x_{N+1}\right)\right]}{\left(2C_l - \Lambda\gamma_N C_u^2\right)}\Bigg] + \sum_{k=1}^{N} \frac{2\gamma_k B\left(\Delta H_k B + \Delta g_k(C_l + d\Delta H_k + \Lambda C_u^2\gamma_k)\right)}{\left(2C_l - \Lambda\gamma_k C_u^2\right)} \\
&\quad + \sum_{k=1}^{N} \frac{\Lambda C_u^2 \gamma_k^2}{\left(2C_l - \Lambda\gamma_k C_u^2\right)}\left[d\Delta g_k^2 + \frac{c_2}{\eta_k^2}\right].
\end{aligned}
$$

Note that , and $\mathbb{E}_{\xi_{[N]}}\left[f\left(x_k\right)\right] \geq f(x^*)$. Using these facts, we obtain

$$
\begin{aligned}
&\sum_{k=1}^{N} \gamma_k \mathbb{E}_{\xi_{[N]}} \left\|\nabla f\left(x_k\right)\right\|^2 \\
&\leq 2\Bigg[\frac{f\left(x_1\right)}{\left(2C_l - \Lambda\gamma_1 C_u^2\right)} - f(x^*)\sum_{k=2}^{N}\left(\frac{1}{\left(2C_l - \Lambda\gamma_{k-1} C_u^2\right)} - \frac{1}{\left(2C_l - \Lambda\gamma_k C_u^2\right)}\right) \\
&\quad - \frac{f(x^*)}{\left(2C_l - \Lambda\gamma_N C_u^2\right)}\Bigg] + \sum_{k=1}^{N} \frac{2\gamma_k B\left(\Delta H_k B + \Delta g_k(C_l + d\Delta H_k + \Lambda C_u^2\gamma_k)\right)}{\left(2C_l - \Lambda\gamma_k C_u^2\right)} \\
&\quad + \sum_{k=1}^{N} \frac{\Lambda C_u^2 \gamma_k^2}{\left(2C_l - \Lambda\gamma_k C_u^2\right)}\left[d\Delta g_k^2 + \frac{c_2}{\eta_k^2}\right]
\end{aligned}
$$

$$= \frac{2\left(f(x_1) - f(x^*)\right)}{(2C_l - \Lambda\gamma_1 C_u^2)} + \sum_{k=1}^{N} \frac{2\gamma_k B\left(\Delta H_k B + \Delta g_k (C_l + d\Delta H_k + \Lambda C_u^2 \gamma_k)\right)}{(2C_l - \Lambda\gamma_k C_u^2)}$$
$$+ \sum_{k=1}^{N} \frac{\Lambda C_u^2 \gamma_k^2}{(2C_l - \Lambda\gamma_k C_u^2)} \left[d\Delta g_k^2 + \frac{c_2}{\eta_k^2} \right].$$

The last inequality follows from the fact that $\left(\frac{1}{(2C_l - \Lambda\gamma_{k-1}C_u^2)} - \frac{1}{(2C_l - \Lambda\gamma_k C_u^2)}\right) \geq 0$. The bound in (4.29) follows by using the distribution of R (specified in (4.17)), and plugging ΔH_k and Δg_k in the RHS above. □

We now specialize the result obtained in the proposition above, to derive a non-asymptotic bound for ZRSQN with gradients and Hessian estimates provided by (**O4**).

Proof. ***(Theorem 12)***
Recall that the stepsize γ_k, perturbation constant η_k and mini-batch size m_k are defined as follows:

$$\gamma_k = min\left\{ \frac{2C_l - 1}{\Lambda C_u^2}, \frac{1}{(d^2 N)^{2/3}} \right\}, \quad \eta_k = \frac{1}{(d^5 N)^{1/6}}, \quad \text{and} \qquad m_k = N, \quad \forall k \geq 1. \tag{4.32}$$

Combining (4.17) with (4.29), we obtain

$$\begin{aligned}
&\mathbb{E}\left[\|\nabla f(x_R)\|^2\right] \\
&\leq \frac{1}{\sum_{k=1}^{N}\gamma_k}\left[\frac{2D_f}{(2C_l - \Lambda\gamma_1 C_u^2)}\right. \\
&\quad +2B\sum_{k=1}^{N}\frac{\gamma_k\left[Bc_1'\eta_k^2 + \frac{Bc_3}{\eta_k\sqrt{m_k}} + \left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)\left(C_l + dc_1'\eta_k^2 + \frac{dc_3}{\eta_k\sqrt{m_k}} + \Lambda C_u^2\gamma_k\right)\right]}{(2C_l - \Lambda\gamma_k C_u^2)} \\
&\quad \left.+\sum_{k=1}^{N}\frac{\Lambda C_u^2\gamma_k^2}{(2C_l - \Lambda\gamma_k C_u^2)}\left[d\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)^2 + \frac{c_2}{\eta_k^2}\right]\right] \\
&= \frac{1}{N\gamma}\left[\frac{2D_f}{(2C_l - \Lambda\gamma C_u^2)}\right. \\
&\quad +2NB\gamma\frac{\left[Bc_1'\eta^2 + \frac{Bc_3}{\eta\sqrt{m}} + \left(c_1\eta_k^2 + \frac{c_3}{\eta\sqrt{m}}\right)\left(C_l + dc_1'\eta^2 + \frac{dc_3}{\eta\sqrt{m}} + \Lambda C_u^2\gamma\right)\right]}{(2C_l - \Lambda\gamma C_u^2)} \\
&\quad \left.+\frac{\Lambda N C_u^2\gamma^2}{(2C_l - \Lambda\gamma C_u^2)}\left[d\left(c_1\eta^2 + \frac{c_3}{\eta\sqrt{m}}\right)^2 + \frac{c_2}{\eta^2}\right]\right] \\
&\leq \frac{1}{N\gamma}\left[2D_f + \Lambda N C_u^2\gamma^2\left[d\left(c_1\eta^2 + \frac{c_3}{\eta\sqrt{m}}\right)^2 + \frac{c_2}{\eta^2}\right]\right.
\end{aligned}$$

$$+ 2NB\gamma\left[Bc_1^{'}\eta^2 + \frac{Bc_3}{\eta\sqrt{m}} + \left(c_1\eta_k^2 + \frac{c_3}{\eta\sqrt{m}}\right)\left(C_l + dc_1^{'}\eta^2 + \frac{dc_3}{\eta\sqrt{m}} + \Lambda C_u^2\gamma\right)\right]\Bigg] \tag{4.33}$$

$$\begin{aligned}
&= \frac{2D_f}{N\gamma} + \Lambda C_u^2\gamma\left[dc_1^2\eta^4 + 2c_1c_3\frac{\eta}{\sqrt{m}} + \frac{dc_3^2}{\eta^2 m} + \frac{c_2}{\eta^2}\right] \\
&\quad + 2B\Bigg(Bc_1^{'}\eta^2 + \frac{Bc_3}{\eta\sqrt{m}} + 3C_lc_1\eta^2 + dc_1c_1^{'}\eta^4 + \frac{dc_1c_3\eta}{\sqrt{m}} \\
&\quad + \frac{3C_1c_3}{\eta\sqrt{m}} + \frac{dc_1^{'}c_3\eta}{\sqrt{m}} + \frac{dc_3^2}{\eta^2 m}\Bigg) \\
&\leq \frac{2D_f}{N}max\left\{\frac{\Lambda C_u^2}{2C_l - 1}, (d^2N)^{2/3}\right\} \\
&\quad + \Lambda C_u^2\left[\frac{dc_1^2}{(d^5N)^{2/3}} + \frac{2dc_1c_3}{d^{5/6}N^{2/3}} + \frac{dd^{5/3}c_3^2}{N^{2/3}} + c_2d^{5/3}N^{1/3}\right]\frac{1}{(d^2N)^{2/3}} \\
&\quad + 2B\left(\frac{Bc_1^{'}}{(d^5N)^{1/3}} + \frac{Bc_3d^{5/6}}{N^{1/3}} + \frac{3C_lc_1}{(d^5N)^{1/3}} + \frac{dc_1c_1^{'}}{(d^5N)^{2/3}} + \frac{dc_1c_3}{d^{5/6}N^{2/3}}\right. \\
&\quad \left. + \frac{3C_1c_3d^{5/6}}{N^{1/3}} + \frac{dc_1^{'}c_3}{d^{5/6}N^{2/3}} + \frac{dd^{5/3}c_3^2}{N^{2/3}}\right) \\
&\leq \frac{2\Lambda C_u^2D_f}{2NC_l - N} + \frac{2D_fd^{4/3}}{N^{1/3}} \\
&\quad + \Lambda C_u^2\left[\frac{c_1^2}{d^{11/3}N^{4/3}} + \frac{2c_1c_3}{d^{7/6}N^{4/3}} + \frac{d^{4/3}c_3^2}{N^{4/3}} + \frac{c_2d^{1/3}}{N^{1/3}}\right] \\
&\quad\quad + 2B\left(\frac{Bc_1^{'}}{(d^5N)^{1/3}} + \frac{Bc_3d^{5/6}}{N^{1/3}} + \frac{3C_lc_1}{(d^5N)^{1/3}} + \frac{c_1c_1^{'}}{d^{7/3}N^{2/3}} + \frac{d^{1/6}c_1c_3}{N^{2/3}}\right. \\
&\quad \left. + \frac{3C_1c_3d^{5/6}}{N^{1/3}} + \frac{d^{1/6}c_1^{'}c_3}{N^{2/3}} + \frac{d^{8/3}c_3^2}{N^{2/3}}\right) \\
&= \frac{2\Lambda C_u^2D_f}{2NC_l - N} + \frac{1}{N^{1/3}}\left[2D_fd^{4/3} + \Lambda C_u^2\left(\frac{c_1^2}{d^{11/3}N} + \frac{2c_1c_3}{d^{7/6}N} + \frac{d^{4/3}c_3^2}{N} + c_2d^{1/3}\right)\right. \\
&\quad + 2B\left(\frac{Bc_1^{'}}{d^{5/3}} + Bc_3d^{5/6} + \frac{3C_lc_1}{d^{5/3}} + \frac{c_1c_1^{'}}{d^{7/3}N^{1/3}} + \frac{d^{1/6}c_1c_3}{N^{1/3}}\right. \\
&\quad \left.\left. + 3C_1c_3d^{5/6} + \frac{d^{1/6}c_1^{'}c_3}{N^{1/3}} + \frac{d^{8/3}c_3^2}{N^{1/3}}\right)\right].
\end{aligned} \tag{4.34}$$

In the above, inequality (4.33) follows by using the fact that $\gamma \leq 1/L$, and the inequality (4.34) follows by using the definition of γ, η and m. □

Proof of Theorem 11

Proof. ***(Theorem 11)***

Proof follows in a similar manner as that of Theorem 12 in Section 4.6.2 after setting $m_k = \infty, \forall k \geq 1$ or $c_3 = 0$. □

4.6.3 Proofs for Stochastic Convex Optimization: ZRSG

We prove Theorem 14 first, and Theorem 13 would follow through a simple modification to the proof of Theorem 14.

Proof of Theorem 14

In the proposition below, we state and prove a general result that holds for any choice of non-increasing stepsize sequence, perturbation constants and batch sizes. Subsequently, we specialize the result for the choice of parameters suggested in Theorem 14 and 18, to prove the same.

Proposition 3. *Assume (**A1**) and (**A5**). With the oracle (**O2**), suppose that the ZRSG algorithm is run with a non-increasing stepsize sequence satisfying* $0 < \gamma_k \leq 1/L, \forall k \geq 1$ *and with the probability mass function* $P_R(\cdot)$ *as defined in* (4.17)*, then, for any* $N \geq 1$*, we have*

$$\begin{aligned}
&\mathbb{E}\left[f\left(x_R\right)\right] - f(x^*) \\
&\leq \frac{1}{\sum_{k=1}^{N}\gamma_k}\left[\frac{D^2}{(2-L\gamma_1)} + 2\sqrt{d}D\sum_{k=1}^{N}\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)\frac{(\gamma_k + L\gamma_k^2)}{(2-L\gamma_k)}\right. \\
&\qquad \left. + \sum_{k=1}^{N}\frac{\gamma_k^2}{(2-L\gamma_k)}\left(d\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)^2 + \frac{c_2}{\eta_k^2}\right)\right],
\end{aligned} \tag{4.35}$$

where constants c_1*,* c_2 *and* c_3 *are as defined in (**O2**), and* D *as defined in (**A5**).*

Proof. Let $\omega_k = \|x_k - x^*\|$ for any $x_k \in \mathcal{W}$. Then for any $k = 1, \ldots, N$, we have,

$$\begin{aligned}
\omega_{k+1}^2 &= \|x_{k+1} - x^*\|^2 \\
&= \|\Pi_{\mathcal{W}}\left(x_k - \gamma_k g(x_k, \xi_k, m_k)\right) - x^*\|^2 \\
&\leq \|x_k - \gamma_k g\left(x_k, \xi_k, m_k\right) - x^*\|^2 \qquad (4.36) \\
&= \omega_k^2 - 2\gamma_k\left\langle g\left(x_k, \xi_k, m_k\right), x_k - x^*\right\rangle + \gamma_k^2\left\|g\left(x_k, \xi_k, m_k\right)\right\|^2. \qquad (4.37)
\end{aligned}$$

The inequality in (4.36) holds because x^* is already in the convex set $\mathcal{W}$ and $\Pi_{\mathcal{W}}$ is a non-expansive projection operator. Taking expectations with respect to $\xi_{[k]}$ on both sides of (4.37), and using (i) $\mathbb{E}_{\xi_{[k]}}\left[g\left(x_k, \xi_k, m_k\right)\right] =$

$\mathbb{E}_{\xi_k}\left[g\left(x_k,\xi_k,m_k\right)|\xi_{[k-1]}\right] = \mathbb{E}_{\xi_k}\left[g\left(x_k,\xi_k,m_k\right)|x_k\right] \leq \nabla f\left(x_k\right) + c_1\eta_k^2\mathbf{1}_{d\times 1} + \frac{c_3}{\eta_k\sqrt{m_k}}\mathbf{1}_{d\times 1}$, and (ii) $\mathbb{E}_{\xi_{[k]}}\left[\left\|g\left(x_k,\xi_k,m_k\right)\right\|^2\right] \leq \left\|\mathbb{E}_{\xi_{[k]}}\left[g\left(x_k,\xi_k,m_k\right)\right]\right\|^2 + c_2/\eta_k^2$, we obtain

$$\begin{aligned}
\mathbb{E}[\omega_{k+1}^2] &\leq \mathbb{E}[\omega_k^2] - 2\gamma_k\left\langle \nabla f\left(x_k\right) + c_1\eta_k^2\mathbf{1}_{d\times 1} + \frac{c_3}{\eta_k\sqrt{m_k}}\mathbf{1}_{d\times 1}, x_k - x^*\right\rangle \\
&\quad + \gamma_k^2\left[\left\|\mathbb{E}_{\xi_{[k]}}\left[g\left(x_k,\xi_k,m_k\right)\right]\right\|^2 + \frac{c_2}{\eta_k^2}\right] \\
&\leq \mathbb{E}[\omega_k^2] - 2\gamma_k\left\langle \nabla f\left(x_k\right) + c_1\eta_k^2\mathbf{1}_{d\times 1} + \frac{c_3}{\eta_k\sqrt{m_k}}\mathbf{1}_{d\times 1}, x_k - x^*\right\rangle \\
&\quad + \gamma_k^2\Bigg[\left\|\nabla f\left(x_k\right)\right\|^2 + 2\sqrt{d}\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)\left\|\nabla f\left(x_k\right)\right\| \\
&\quad + d\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)^2 + \frac{c_2}{\eta_k^2}\Bigg].
\end{aligned}$$

Using the fact that $f(\cdot)$ is convex, we have $\left\|\nabla f\left(x_k\right)\right\|^2 \leq L\left\langle \nabla f\left(x_k\right), x_k - x^*\right\rangle$, further from (**A1**) and (**A5**), we have $\|\nabla f(x_k)\| \leq L\|x_k - x^*\| \leq LD$. Plugging it in equation above, we obtain,

$$\begin{aligned}
&\mathbb{E}[\omega_{k+1}^2] \\
&\quad \leq \mathbb{E}[\omega_k^2] - 2\gamma_k\left\langle \nabla f\left(x_k\right) + c_1\eta_k^2\mathbf{1}_{d\times 1} + \frac{c_3}{\eta_k\sqrt{m_k}}\mathbf{1}_{d\times 1}, x_k - x^*\right\rangle \\
&\qquad + \gamma_k^2\Bigg[L\left\langle \nabla f\left(x_k\right), x_k - x^*\right\rangle + 2\sqrt{d}LD\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right) \\
&\quad + d\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)^2 + \frac{c_2}{\eta_k^2}\Bigg] \\
&\leq \mathbb{E}[\omega_k^2] - \left(2\gamma_k - L\gamma_k^2\right)\left\langle \nabla f\left(x_k\right), x_k - x^*\right\rangle + 2\gamma_k\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)\|x_k - x_{k_0}\|_1 \\
&\qquad + \gamma_k^2\Bigg[2\sqrt{d}LD\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right) + d\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)^2 + \frac{c_2}{\eta_k^2}\Bigg] \\
&\leq \mathbb{E}[\omega_k^2] - \left(2\gamma_k - L\gamma_k^2\right)\left[f\left(x_k\right) - f(x^*)\right] + 2\sqrt{d}D(\gamma_k + L\gamma_k^2)\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right) \\
&\qquad + \gamma_k^2\Bigg[d\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)^2 + \frac{c_2}{\eta_k^2}\Bigg],
\end{aligned}$$

where the second inequality follows from the fact that $-\sum_{i=1}^d x_i \leq \|X\|_1$ for any vector X, and the last inequality follows from the fact that $f(\cdot)$ is convex along with $\|X\|_1 \leq \sqrt{d}\|X\|$ for any vector X. Re-arranging the terms, we obtain

$$\gamma_k\left[f\left(x_k\right) - f(x^*)\right] \leq \frac{1}{\left(2 - L\gamma_k\right)}\Bigg[\omega_k^2 - \mathbb{E}[\omega_{k+1}^2] + 2\sqrt{d}D(\gamma_k + L\gamma_k^2)\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)$$

$$+ \gamma_k^2 \left(d \left(c_1 \eta_k^2 + \frac{c_3}{\eta_k \sqrt{m_k}} \right)^2 + \frac{c_2}{\eta_k^2} \right) \Bigg].$$

Now summing up the inequality above from $k = 1$ to N and taking expectation on both sides of above equation, we obtain

$$\begin{aligned}
&\sum_{k=1}^{N} \gamma_k \mathbb{E}_{\xi_{[N]}} \left[f\left(x_k\right) - f(x^*) \right] \\
&\leq \sum_{k=1}^{N} \frac{\mathbb{E}_{\xi_{[N]}}[\omega_k^2] - \mathbb{E}_{\xi_{[N]}}[\omega_{k+1}^2]}{(2 - L\gamma_k)} + 2\sqrt{d} D \sum_{k=1}^{N} \left(c_1 \eta_k^2 + \frac{c_3}{\eta_k \sqrt{m_k}} \right) \frac{(\gamma_k + L\gamma_k^2)}{(2 - L\gamma_k)} \\
&\quad + \sum_{k=1}^{N} \frac{\gamma_k^2}{(2 - L\gamma_k)} \left(d \left(c_1 \eta_k^2 + \frac{c_3}{\eta_k \sqrt{m_k}} \right)^2 + \frac{c_2}{\eta_k^2} \right).
\end{aligned}$$

Using the fact that $\mathbb{E}_{\xi_{[N]}}[\omega_k] \geq 0$ and $L\gamma_k \leq 1$ for all $k \geq 1$, we obtain

$$\begin{aligned}
&\sum_{k=1}^{N} \gamma_k \mathbb{E}_{\xi_{[N]}} \left[f\left(x_k\right) - f(x^*) \right] \\
&= \left[\frac{\omega_1^2}{(2 - L\gamma_1)} - \sum_{k=2}^{N} \left(\frac{1}{(2 - L\gamma_{k-1})} - \frac{1}{(2 - L\gamma_k)} \right) \mathbb{E}_{\xi_{[N]}} \left[\omega_k^2 \right] - \frac{\mathbb{E}_{\xi_{[N]}} \left[\omega_{N+1}^2 \right]}{(2 - L\gamma_N)} \right] \\
&\quad + 2\sqrt{d} D \sum_{k=1}^{N} \left(c_1 \eta_k^2 + \frac{c_3}{\eta_k \sqrt{m_k}} \right) \frac{(\gamma_k + L\gamma_k^2)}{(2 - L\gamma_k)} \\
&\quad + \sum_{k=1}^{N} \frac{\gamma_k^2}{(2 - L\gamma_k)} \left(d \left(c_1 \eta_k^2 + \frac{c_3}{\eta_k \sqrt{m_k}} \right)^2 + \frac{c_2}{\eta_k^2} \right) \\
&\leq \frac{\omega_1^2}{(2 - L\gamma_1)} + 2\sqrt{d} D \sum_{k=1}^{N} \left(c_1 \eta_k^2 + \frac{c_3}{\eta_k \sqrt{m_k}} \right) \frac{(\gamma_k + L\gamma_k^2)}{(2 - L\gamma_k)} \\
&\quad + \sum_{k=1}^{N} \frac{\gamma_k^2}{(2 - L\gamma_k)} \left(d \left(c_1 \eta_k^2 + \frac{c_3}{\eta_k \sqrt{m_k}} \right)^2 + \frac{c_2}{\eta_k^2} \right)
\end{aligned}$$

We conclude by combining the above result with (4.17). □

Proof. **(Theorem 14)**

Recall that the stepsize γ_k, perturbation constant η_k and mini-batch size m_k are defined as follows:

$$\gamma_k = min \left\{ \frac{1}{L}, \frac{1}{(d^2 N)^{2/3}} \right\}, \qquad \eta_k = \frac{1}{(d^5 N)^{1/6}}, \qquad \text{and} \qquad m_k = N, \qquad \forall k \geq 1. \tag{4.38}$$

Combining (4.17) with (4.35), we obtain

$$
\begin{aligned}
&\mathbb{E}\left[f\left(x_R\right)\right] - f(x^*) \\
&\leq \frac{1}{\sum_{k=1}^{N}\gamma_k}\left[\frac{D^2}{(2-L\gamma_1)} + 2\sqrt{d}D\sum_{k=1}^{N}\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)\frac{(\gamma_k + L\gamma_k^2)}{(2-L\gamma_k)}\right. \\
&\qquad \left. + \sum_{k=1}^{N}\frac{\gamma_k^2}{(2-L\gamma_k)}\left(d\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)^2 + \frac{c_2}{\eta_k^2}\right)\right] \\
&\leq \frac{1}{N\gamma}\left[D^2 + 4\sqrt{d}DN\gamma\left(c_1\eta^2 + \frac{c_3}{\eta\sqrt{m}}\right) + N\gamma^2\left(d\left(c_1\eta^2 + \frac{c_3}{\eta\sqrt{m}}\right)^2 + \frac{c_2}{\eta^2}\right)\right] \qquad (4.39) \\
&= \frac{D^2}{N\gamma} + 4\sqrt{d}D\left(c_1\eta^2 + \frac{c_3}{\eta\sqrt{m}}\right) + \gamma\left[dc_1^2\eta^4 + 2dc_1c_3\frac{\eta}{\sqrt{m}} + \frac{dc_3^2}{\eta^2 m} + \frac{c_2}{\eta^2}\right] \\
&\leq \frac{D^2}{N}max\left\{L, (d^2N)^{2/3}\right\} + 4\sqrt{d}D\left(\frac{c_1}{(d^5N)^{1/3}} + \frac{c_3d^{5/6}}{N^{1/3}}\right) \\
&\quad + \frac{1}{(d^2N)^{2/3}}\left[\frac{dc_1^2}{(d^5N)^{2/3}} + \frac{2dc_1c_3}{d^{5/6}N^{2/3}} + \frac{dd^{5/3}c_3^2}{N^{2/3}} + \frac{c_2d^{5/3}}{N^{-1/3}}\right] \qquad (4.40) \\
&= \frac{LD^2}{N} + \frac{1}{N^{1/3}}\left[D^2d^{4/3} + 4\sqrt{d}D\left(\frac{c_1}{d^{5/3}} + c_3d^{5/6}\right) + \frac{c_1^2}{d^{11/3}N}\right. \\
&\quad \left. + \frac{2c_1c_3}{d^{7/6}N} + \frac{d^{4/3}c_3^2}{N} + d^{1/3}c_2\right].
\end{aligned}
$$

In the above, inequality (4.39) follows by using the fact that $\gamma \leq 1/L$, and the inequality (4.40) follows by using the definition of γ, η and m. □

Proof of Theorem 13

Proof. Proof follows in a similar manner as that of Theorem 14 in Section 4.6.3 after setting $m_k = \infty, \forall k \geq 1$ or $c_3 = 0$. □

4.6.4 Proofs for Stochastic Convex Optimization: ZSGD

We prove Theorem 16 first, and Theorem 15 would follow through a simple modification to the proof of Theorem 16.

Proof of Theorem 16

The proof proceeds through a sequence of lemmas. We follow the technique from (Jain *et al.*, 2019) and prove that the last iterate x_N of the ZSGD algorithm has an optimization error rate of $\mathcal{O}(N^{-1/3})$ with oracle (**O2**). As mentioned before, the proof involves significant deviations owing to the fact that unbiased gradient information is not available, leading to additional terms involving perturbation constants (arising out of gradient bias), and mini-batch sizes (arising due to estimation errors).

Recall that N_i, l is defined as follows:

$$\begin{aligned} &\text{Let } l := \inf\{i : N \cdot 2^{-i} \leq 1\}, \\ &N_i := N - \lceil N \cdot 2^{-i} \rceil,\ 0 \leq i \leq l, \text{ and } N_{l+1} := N. \end{aligned} \tag{4.41}$$

Further, when $N_i < k \leq N_{i+1}, 0 \leq i \leq l$, stepsize γ_k, perturbation constant η_k, and mini-batch size m_k is defined as follows:

$$\gamma_k = \frac{C \cdot 2^{-i}}{\sqrt{d} N^{2/3}}, \qquad \eta_k = \frac{2^{-i/4}}{\sqrt{d} N^{1/6}} \quad \text{and} \quad m_k = 2^i N, \tag{4.42}$$

where $C > 0$. Note that, unlike (Jain *et al.*, 2019), parameters η_k and m_k are local to our setting, and due to the inverse scaling of variance in gradient estimates with η_k, the stepsizes γ_k chosen is of $\mathcal{O}(\frac{1}{N^{2/3}})$ and not $\mathcal{O}(\frac{1}{\sqrt{N}})$.

We divide the proof into phases N_i, let $x_1, \ldots, x_N$ be the output of the ZSGD algorithm. We start with a variant of Lemma 1 from (Jain *et al.*, 2019). In comparison to their result, our claim below features additional factors involving perturbation constant η_k and mini-batch size m_k owing to the zeroth-order setting we consider.

Lemma 20. *Assume (**A4**) and (**A5**). With the oracle (**O2**), suppose that the ZSGD algorithm is run with stepsize sequence $\{\gamma_k\}_{k=1}^N$. Then, given any $1 < k_0 < k_1 \leq N$, we have*

$$\sum_{k=k_0}^{k_1} 2\gamma_k \mathbb{E}\left[f\left(x_k\right) - f(x_{k_0})\right] \leq \sum_{k=k_0}^{k_1} \left(2\sqrt{d}\gamma_k D\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right) + \gamma_k^2 \mathcal{G}_k^2\right),$$

where $\mathcal{G}_k^2 := \left[G^2 + 2\sqrt{d}G\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right) + d\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)^2 + \frac{c_2}{\eta_k^2}\right]$, *constants* c_1, c_2 *is as defined in (**O2**) and* D *is as defined in (**A5**).*

Proof. Let $\omega_k = \|x_k - x_{k_0}\|$ for any $x_k \in \mathbb{R}^d$. Then for any $k = 1, \ldots, N$, we have,

$$\begin{aligned}
\omega_{k+1}^2 &= \|x_{k+1} - x_{k_0}\|^2 \\
&= \|\Pi_{\mathcal{W}}\left(x_k - \gamma_k g(x_k, \xi_k, m_k)\right) - x_{k_0}\|^2 \\
&\leq \|x_k - \gamma_k g\left(x_k, \xi_k, m_k\right) - x_{k_0}\|^2 \qquad (4.43) \\
&= \omega_k^2 - 2\gamma_k \left\langle g\left(x_k, \xi_k, m_k\right), x_k - x_{k_0}\right\rangle + \gamma_k^2 \left\|g\left(x_k, \xi_k, m_k\right)\right\|^2. \qquad (4.44)
\end{aligned}$$

The inequality in (4.43) holds because x_{k_0} is already in the convex set $\mathcal{W}$, and $\Pi_{\mathcal{W}}$ is a non-expansive projection operator. Taking expectations with respect to $\xi_{[k]}$ on both sides of (4.44), and using (i) $\mathbb{E}_{\xi_{[k]}}\left[g\left(x_k, \xi_k, m_k\right)\right] = \mathbb{E}_{\xi_k}\left[g\left(x_k, \xi_k, m_k\right)|\xi_{[k-1]}\right] = \mathbb{E}_{\xi_k}\left[g\left(x_k, \xi_k, m_k\right)|x_k\right] \leq \nabla f\left(x_k\right) + c_1\eta_k^2 \mathbf{1}_{d\times 1} + \frac{c_3}{\eta_k\sqrt{m_k}}\mathbf{1}_{d\times 1}$, and (ii) $\mathbb{E}_{\xi_{[k]}}\left[\left\|g\left(x_k, \xi_k, m_k\right)\right\|^2\right] \leq \left\|\mathbb{E}_{\xi_{[k]}}\left[g\left(x_k, \xi_k, m_k\right)\right]\right\|^2 + c_2/\eta_k^2$, we obtain

$$\begin{aligned}
\mathbb{E}[\omega_{k+1}^2] &\leq \mathbb{E}[\omega_k^2] - 2\gamma_k \left\langle \nabla f\left(x_k\right) + c_1\eta_k^2 \mathbf{1}_{d\times 1} + \frac{c_3}{\eta_k\sqrt{m_k}}\mathbf{1}_{d\times 1}, x_k - x_{k_0}\right\rangle \\
&\quad + \gamma_k^2 \left[\left\|\mathbb{E}_{\xi_{[k]}}\left[g\left(x_k, \xi_k, m_k\right)\right]\right\|^2 + \frac{c_2}{\eta_k^2}\right] \\
&\leq \mathbb{E}[\omega_k^2] - 2\gamma_k \left\langle \nabla f\left(x_k\right) + c_1\eta_k^2 \mathbf{1}_{d\times 1} + \frac{c_3}{\eta_k\sqrt{m_k}}\mathbf{1}_{d\times 1}, x_k - x_{k_0}\right\rangle \\
&\quad + \gamma_k^2 \left[\|\nabla f\left(x_k\right)\|^2 + 2\sqrt{d}\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)\|\nabla f\left(x_k\right)\| \right. \\
&\quad \left. + d\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)^2 + \frac{c_2}{\eta_k^2}\right].
\end{aligned}$$

Using $\|\nabla f(x)\| \leq G$ from (**A4**), we obtain

$$\begin{aligned}
\mathbb{E}[\omega_{k+1}^2] &\leq \mathbb{E}[\omega_k^2] - 2\gamma_k \left\langle \nabla f\left(x_k\right) + c_1\eta_k^2 \mathbf{1}_{d\times 1} + \frac{c_3}{\eta_k\sqrt{m_k}}\mathbf{1}_{d\times 1}, x_k - x_{k_0}\right\rangle \\
&\quad + \gamma_k^2 \left[G^2 + 2\sqrt{d}G\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right) + d\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)^2 + \frac{c_2}{\eta_k^2}\right] \\
&\leq \mathbb{E}[\omega_k^2] - 2\gamma_k \left\langle \nabla f\left(x_k\right), x_k - x_{k_0}\right\rangle + 2\gamma_k \left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)\|x_k - x_{k_0}\|_1 \\
&\quad + \gamma_k^2 \left[G^2 + 2\sqrt{d}G\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right) + d\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)^2 + \frac{c_2}{\eta_k^2}\right] \\
&\leq \mathbb{E}[\omega_k^2] - 2\gamma_k \left[f\left(x_k\right) - f(x_{k_0})\right] + 2\sqrt{d}\gamma_k\omega_k \left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right) \\
&\quad + \gamma_k^2 \left[G^2 + 2\sqrt{d}G\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right) + d\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)^2 + \frac{c_2}{\eta_k^2}\right],
\end{aligned}$$

where the second inequality follows from the fact that $-\sum_{i=1}^{d} x_i \leq \|X\|_1$ for any vector X, and the last inequality follows from the fact that $f(\cdot)$ is convex along with $\|X\|_1 \leq \sqrt{d}\|X\|$ for any vector X. Re-arranging the terms, we obtain

$$\begin{aligned}2\gamma_k \left[f\left(x_k\right) - f(x_{k_0})\right] \leq \mathbb{E}[\omega_k^2] - \mathbb{E}[\omega_{k+1}^2] + 2\sqrt{d}\gamma_k\omega_k \left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right) \\ + \gamma_k^2\Bigg[G^2 + 2\sqrt{d}G\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right) \\ + d\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)^2 + \frac{c_2}{\eta_k^2}\Bigg].\end{aligned}$$

Summing the above over $k = k_0$ to k_1, taking expectations, and using (**A5**), i.e., $\|x_1 - x^*\| \leq D$, we conclude

$$\begin{aligned}\sum_{k=k_0}^{k_1} 2\gamma_k \mathbb{E}\left[f\left(x_k\right) - f(x_{k_0})\right] \leq \sum_{k=k_0}^{k_1}\Bigg(2\sqrt{d}\gamma_k D\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right) \\ + \gamma_k^2\Bigg[G^2 + 2\sqrt{d}G\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right) \\ + d\left(c_1\eta_k^2 + \frac{c_3}{\eta_k\sqrt{m_k}}\right)^2 + \frac{c_2}{\eta_k^2}\Bigg]\Bigg).\end{aligned}$$

□

Lemma 21. *Under conditions of Lemma 20, with $\gamma_k = \gamma, \eta_k = \eta, \forall k \geq 1$, for any $N \geq 1$, we have*

$$\sum_{k=1}^{N} \mathbb{E}\left[f\left(x_k\right) - f\left(x^*\right)\right] \leq \frac{D^2}{2\gamma} + 2ND\sqrt{d}\left(c_1\eta^2 + \frac{c_3}{\eta\sqrt{m}}\right),$$

*where c_1 is as defined in (**O2**), G is as defined in (**A4**) and D is as defined in (**A5**).*

Proof. Let $\Delta g_k := g(x_k, \xi_k, m_k) - \nabla f(x_k)$ and $y_{k+1} = x_k - \gamma_k\left(\nabla f(x_k) + \Delta g_k\right)$, then we have $x_{k+1} = \Pi_{\mathcal{X}}\left(y_{k+1}\right)$. Using the definition of convexity, we obtain

$$\begin{aligned}&f\left(x_k\right) - f\left(x^*\right) \\ &\leq \nabla f(x_k)^\top\left(x_k - x^*\right) \\ &= \left(\frac{x_k - y_{k+1}}{\gamma_k} - \Delta g_k\right)^\top\left(x_k - x^*\right) \\ &= \frac{1}{\gamma_k}\left(x_k - y_{k+1} - \gamma_k\Delta g_k\right)^\top\left(x_k - x^*\right)\end{aligned}$$

$$= \frac{1}{2\gamma_k}\left(\|x_k - x^*\|^2 + \|x_k - y_{k+1} - \gamma_k \Delta g_k\|^2 - \|y_{k+1} - x^* + \gamma_k \Delta g_k\|^2\right) \quad (4.45)$$
$$= \frac{1}{2\gamma_k}\left(\|x_k - x^*\|^2 - \|y_{k+1} - x^* + \gamma_k \Delta g_k\|^2\right) + \frac{\gamma_k}{2}\|\nabla f(x_k)\|^2,$$

where we have used the identity $2a^\top b = \|a\|^2 + \|b\|^2 - \|a-b\|^2$ in arriving at the equality in (4.45). Using $\|\nabla f(x_k)\|^2 \leq G^2$, we have

$$\begin{aligned}
&f(x_k) - f(x^*) \\
&\leq \frac{1}{2\gamma_k}\left(\|x_k - x^*\|^2 - \|y_{k+1} - x^* + \gamma_k \Delta g_k\|^2\right) + \frac{\gamma_k G^2}{2} \\
&= \frac{1}{2\gamma_k}\left(\|x_k - x^*\|^2 - \|y_{k+1} - x^*\|^2 - \gamma_k^2\|\Delta g_k\|^2 - 2\gamma_k (y_{k+1} - x^*)^T \Delta g_k\right) + \frac{\gamma_k G^2}{2} \\
&\leq \frac{1}{2\gamma_k}\left(\|x_k - x^*\|^2 - \|y_{k+1} - x^*\|^2 - 2\gamma_k (y_{k+1} - x^*)^T \Delta g_k\right) + \frac{\gamma_k G^2}{2}.
\end{aligned}$$

Taking expectations and using $\|y_{k+1} - x^*\| \geq \|x_{k+1} - x^*\|$ (see Lemma 3.1 in (Bubeck, 2015)), we obtain

$$\begin{aligned}
&\mathbb{E}[f(x_k) - f(x^*)] \\
&\leq \frac{1}{2\gamma_k}\Bigg(\mathbb{E}[\|x_k - x^*\|^2] - \mathbb{E}[\|x_{k+1} - x^*\|^2] \\
&\quad - 2\gamma_k \mathbb{E}\left[(y_{k+1} - x^*)^T \left(c_1 \eta_k^2 \mathbf{1}_{d\times 1} + \frac{c_3}{\eta_k \sqrt{m_k}} \mathbf{1}_{d\times 1}\right)\right]\Bigg) + \frac{\gamma_k G^2}{2} \\
&\leq \frac{1}{2\gamma_k}\Bigg(\mathbb{E}[\|x_k - x^*\|^2] - \mathbb{E}[\|x_{k+1} - x^*\|^2] \\
&\quad + 2\left(c_1 \eta_k^2 + \frac{c_3}{\eta_k \sqrt{m_k}}\right)\gamma_k \mathbb{E}[\|y_{k+1} - x^*\|_1]\Bigg) + \frac{\gamma_k G^2}{2} \\
&\leq \frac{1}{2\gamma_k}\Bigg(\mathbb{E}[\|x_k - x^*\|^2] - \mathbb{E}[\|x_{k+1} - x^*\|^2] \\
&\quad + 2\left(c_1 \eta_k^2 + \frac{c_3}{\eta_k \sqrt{m_k}}\right)\gamma_k \sqrt{d}\,\mathbb{E}[\|x_{k+1} - x^*\|]\Bigg) + \frac{\gamma_k G^2}{2}. \qquad (4.46)
\end{aligned}$$

In the above, the second inequality follows from the fact that $-\sum_{i=1}^d x_i \leq \|X\|_1$ for any vector X, and the last inequality follows from the fact that $\|X\|_1 \leq \sqrt{d}\|X\|$ for any vector X. Summing (4.46) over k, with $\gamma_k = \gamma, \eta_k = \eta, \forall k \geq 1$, and using $\|x_1 - x^*\| \leq D$, we conclude

$$\sum_{k=1}^N \mathbb{E}\left[f(x_k) - f(x^*)\right] \leq \frac{D^2}{2\gamma} + 2ND\sqrt{d}\left(c_1 \eta^2 + \frac{c_3}{\eta\sqrt{m}}\right) + \frac{\gamma N G^2}{2}.$$

□

Proof. **(Theorem 16)**

Recall the definition of N_i, l from equation (4.41) and let n_i, $0 \leq i \leq l+1$ be defined as follows:

$$n_i = \arg \inf_{N_i < k \leq N_{i+1}} \mathbb{E}[f(x_k)],\ i \in [l+1], \quad \text{and } n_0 = \arg \inf_{\lceil \frac{N}{4} \rceil \leq k \leq N_1} \mathbb{E}[f(x_k)]. \tag{4.47}$$

We split the horizon N into l phases, then to show that the function value for the final iterate x_N in the last phase ($N_{l+1} = N$) is close to optima $f(x^*)$. Using the fact that $n_{l+1} = N$, we have

$$\mathbb{E}[f(x_N)] = \mathbb{E}[f(x_{n_{l+1}})] = \mathbb{E}[f(x_{n_0})] + \sum_{i=0}^{l} \mathbb{E}[f(x_{n_{i+1}}) - f(x_{n_i})]. \tag{4.48}$$

Now to bound $\mathbb{E}[f(x_{n_{i+1}}) - f(x_{n_i})]$, we first consider the case when $i \geq 1$. Using Lemma 20 with $k_0 = n_i$ and $k_1 = N_{i+2}$, we obtain

$$\begin{aligned}
&\frac{\sum_{k=n_i}^{N_{i+2}} 2\gamma_k \mathbb{E}\left[f(x_k) - f(x_{n_i})\right]}{N_{i+2} - n_i + 1} \\
&\leq \frac{\sum_{k=n_i}^{N_{i+2}} \left(2\sqrt{d}\gamma_k D \left(c_1 \eta_k^2 + \frac{c_3}{\eta_k \sqrt{m_k}} \right) + \gamma_k^2 \mathcal{G}_k^2 \right)}{N_{i+2} - n_i + 1} \\
&\leq 2\sqrt{d}\gamma_{N_{i+1}} D \left(c_1 \eta_{N_{i+1}}^2 + \frac{c_3}{\eta_{N_{i+1}} \sqrt{m_{N_{i+1}}}} \right) + \mathcal{G}_{N_{i+1}}^2 \gamma_{N_{i+1}}^2 \qquad (4.49) \\
&= 2\sqrt{d}\gamma_{N_{i+1}} D \left(c_1 \eta_{N_{i+1}}^2 + \frac{c_3}{\eta_{N_{i+1}} \sqrt{m_{N_{i+1}}}} \right) \\
&\quad + \gamma_{N_{i+1}}^2 \left[G^2 + 2\sqrt{d} G \left(c_1 \eta_{N_{i+1}}^2 + \frac{c_3}{\eta_{N_{i+1}} \sqrt{m_{N_{i+1}}}} \right) \right. \\
&\quad \left. + d \left(c_1 \eta_{N_{i+1}}^2 + \frac{c_3}{\eta_{N_{i+1}} \sqrt{m_{N_{i+1}}}} \right)^2 + \frac{c_2}{\eta_{N_{i+1}}^2} \right] \\
&= \frac{2c_1 DC 2^{-3i/2}}{dN} + \frac{2\sqrt{d} DC c_3 2^{-5i/4}}{N} + \frac{2^{-2i} C^2}{dN^{4/3}} \left[G^2 + \frac{2c_1 G 2^{-i/2}}{\sqrt{d} N^{1/3}} \right. \\
&\quad \left. + \frac{2dc_1 G c_3 2^{-i/4}}{N^{1/3}} + \frac{c_1^2 2^{-i}}{dN^{2/3}} + \frac{2\sqrt{d} c_1 c_3 2^{-3i/4}}{N^{2/3}} + \frac{d^2 c_3^2 2^{-i/2}}{N^{2/3}} + \frac{dc_2 N^{1/3}}{2^{-i/2}} \right]. \qquad (4.50)
\end{aligned}$$

The inequality in (4.49) follows from the fact that γ_k and η_k are decaying in a phase-dependent manner (see (4.42)). Note that from the definition of n_i, $\mathbb{E}[f(x_k) - f(x_{n_i})] \geq$

0 whenever $N_i < k \leq N_{i+1}$. Thus, we have

$$\begin{aligned}
\frac{\sum_{k=n_i}^{N_{i+2}} 2\gamma_k \mathbb{E}\left[f(x_k) - f(x_{n_i})\right]}{N_{i+2} - n_i + 1} &\geq \frac{\sum_{k=N_{i+1}+1}^{N_{i+2}} 2\gamma_k \mathbb{E}\left[f(x_k) - f(x_{n_i})\right]}{N_{i+2} - n_i + 1} \\
&\geq 2\gamma_{N_{i+2}} \frac{N_{i+2} - N_{i+1}}{N_{i+2} - N_i} \mathbb{E}\left[f(x_{n_{i+1}}) - f(x_{n_i})\right] \\
&\geq \frac{2\gamma_{N_{i+2}}}{5} \mathbb{E}\left[f(x_{n_{i+1}}) - f(x_{n_i})\right] \\
&= \frac{2^{-i}C}{5\sqrt{d}N^{2/3}} \mathbb{E}\left[f(x_{n_{i+1}}) - f(x_{n_i})\right], \qquad (4.51)
\end{aligned}$$

where the second inequality follows from the assumption that $\mathbb{E}[f(x_{n_{i+1}})] \geq \mathbb{E}[f(x_{n_i})]$, and the fact that $N_{i+2} - N_i \geq N_{i+2} - n_i + 1$. The last inequality follows from the Lemma 4 of (Jain *et al.*, 2019). Combining (4.50) and (4.51), we obtain

$$\begin{aligned}
&\mathbb{E}[f(x_{n_{i+1}}) - f(x_{n_i})] \\
&\leq \frac{5\sqrt{d}N^{2/3}}{2^{-i}C}\left(\frac{2c_1 DC 2^{-3i/2}}{dN} + \frac{2\sqrt{d}DCc_3 2^{-5i/4}}{N} + \frac{2^{-2i}C^2}{dN^{4/3}}\left[G^2 + \frac{2c_1 G 2^{-i/2}}{\sqrt{d}N^{1/3}}\right.\right. \\
&\quad \left.\left. + \frac{2dc_1 Gc_3 2^{-i/4}}{N^{1/3}} + \frac{c_1^2 2^{-i}}{dN^{2/3}} + \frac{2\sqrt{d}c_1 c_3 2^{-3i/4}}{N^{2/3}} + \frac{d^2 c_3^2 2^{-i/2}}{N^{2/3}} + \frac{dc_2 N^{1/3}}{2^{-i/2}}\right]\right) \\
&= \frac{10c_1 D 2^{-i/2}}{\sqrt{d}N^{1/3}} + \frac{10dc_3 D 2^{-i/4}}{N^{1/3}} + \frac{5C2^{-i}}{\sqrt{d}N^{2/3}}\left[G^2 + \frac{2c_1 G 2^{-i/2}}{\sqrt{d}N^{1/3}}\right. \\
&\quad \left. + \frac{2dc_1 Gc_3 2^{-i/4}}{N^{1/3}} + \frac{c_1^2 2^{-i}}{dN^{2/3}} + \frac{2\sqrt{d}c_1 c_3 2^{-3i/4}}{N^{2/3}} + \frac{d^2 c_3^2 2^{-i/2}}{N^{2/3}} + \frac{dc_2 N^{1/3}}{2^{-i/2}}\right]. \qquad (4.52)
\end{aligned}$$

This completes the proof for the case when $i \geq 1$. The proof for the case when $i = 0$ follows in a similar manner. Plugging (4.52) in (4.48), we obtain

$$\begin{aligned}
&\mathbb{E}[f(x_N)] \\
&= \mathbb{E}[f(x_{n_{l+1}})] = \mathbb{E}[f(x_{n_0})] + \sum_{i=0}^{l} \mathbb{E}[f(x_{n_{i+1}}) - f(x_{n_i})] \\
&\leq \mathbb{E}[f(x_{n_0})] + \frac{10c_1 D}{\sqrt{d}N^{1/3}} + \frac{10dc_3 D}{N^{1/3}} + \frac{5C}{\sqrt{d}N^{2/3}}\left[G^2 + \frac{2c_1 G}{\sqrt{d}N^{1/3}}\right. \\
&\quad \left. + \frac{2dc_1 Gc_3}{N^{1/3}} + \frac{c_1^2}{dN^{2/3}} + \frac{2\sqrt{d}c_1 c_3}{N^{2/3}} + \frac{d^2 c_3^2}{N^{2/3}} + dc_2 N^{1/3}\right] \\
&\quad + \sum_{i=1}^{l}\left(\frac{10c_1 D 2^{-i/2}}{\sqrt{d}N^{1/3}} + \frac{10dc_3 D 2^{-i/4}}{N^{1/3}} + \frac{5C2^{-i}}{\sqrt{d}N^{2/3}}\left[G^2 + \frac{2c_1 G 2^{-i/2}}{\sqrt{d}N^{1/3}}\right.\right. \\
&\quad \left.\left. + \frac{2dc_1 Gc_3 2^{-i/4}}{N^{1/3}} + \frac{c_1^2 2^{-i}}{dN^{2/3}} + \frac{2\sqrt{d}c_1 c_3 2^{-3i/4}}{N^{2/3}} + \frac{d^2 c_3^2 2^{-i/2}}{N^{2/3}} + \frac{dc_2 N^{1/3}}{2^{-i/2}}\right]\right) \\
&\leq \mathbb{E}[f(x_{n_0})] + \frac{10c_1 D}{\sqrt{d}N^{1/3}} + \frac{10dc_3 D}{N^{1/3}} + \frac{5CG^2}{\sqrt{d}N^{2/3}} + \frac{10Cc_1 G}{dN} + \frac{10\sqrt{d}c_1 CGc_3}{N}
\end{aligned}$$

$$
\begin{aligned}
&+\frac{5C(\frac{c_1^2}{d}+2\sqrt{d}c_1c_3+d^2c_3^2)}{\sqrt{d}N^{4/3}}+\frac{5\sqrt{d}Cc_2}{N^{1/3}}+\Bigg(\frac{25c_1D}{\sqrt{d}N^{1/3}}+\frac{53dc_3D}{N^{1/3}}+\frac{5CG^2}{\sqrt{d}N^{2/3}}\\
&+\frac{10Cc_1G}{dN}+\frac{10\sqrt{d}c_1CGc_3}{N}+\frac{5C(\frac{c_1^2}{d}+2\sqrt{d}c_1c_3+d^2c_3^2)}{\sqrt{d}N^{4/3}}+\frac{12.5\sqrt{d}Cc_2}{N^{1/3}}\Bigg)\\
&=\inf_{\lceil\frac{N}{4}\rceil\le k\le N_1}\mathbb{E}[f(x_k)]+\frac{35c_1D}{\sqrt{d}N^{1/3}}+\frac{63dc_3D}{N^{1/3}}+\frac{10CG^2}{\sqrt{d}N^{2/3}}+\frac{20Cc_1G}{dN}\\
&+\frac{20\sqrt{d}c_1CGc_3}{N}+\frac{10C(\frac{c_1}{\sqrt{d}}+dc_3)^2}{\sqrt{d}N^{4/3}}+\frac{17.5\sqrt{d}Cc_2}{N^{1/3}}\\
&=\inf_{\lceil\frac{N}{4}\rceil\le k\le N_1}\mathbb{E}[f(x_k)]+\frac{D(35\frac{c_1}{\sqrt{d}}+63dc_3)}{N^{1/3}}+\frac{10CG^2}{\sqrt{d}N^{2/3}}\\
&+\frac{20Cc_1G(d^{-1}+\sqrt{d}c_3)}{N}+\frac{10C(\frac{c_1}{\sqrt{d}}+dc_3)^2}{\sqrt{d}N^{4/3}}+\frac{17.5\sqrt{d}Cc_2}{N^{1/3}}. \qquad (4.53)
\end{aligned}
$$

Note that for all $k \le N_1$, we have step size $\gamma_k = \frac{C}{\sqrt{d}N^{2/3}}$ and perturbation parameter $\eta_k = \frac{1}{\sqrt{d}N^{1/6}}$. Let x_k be the output of ZSGD algorithm, then using the fact that infimum is smaller than any weighted average, we have

$$
\begin{aligned}
\inf_{\lceil\frac{N}{4}\rceil\le k\le N_1}\mathbb{E}[f(x_k)-f(x^*)] &\le \frac{1}{N_1-\lceil\frac{N}{4}\rceil+1}\sum_{k=\lceil\frac{N}{4}\rceil}^{N_1}\mathbb{E}[f(x_k)-f(x^*)]\\
&\le \frac{2}{N_1}\sum_{k=1}^{N_1}\mathbb{E}[f(x_k)-f(x^*)] \qquad (4.54)\\
&\le \frac{2}{N_1}\left[\frac{\sqrt{d}D^2N^{2/3}}{2C}+\frac{CG^2N_1}{2\sqrt{d}N^{2/3}}+\frac{2N_1Dc_1}{\sqrt{d}N^{1/3}}+\frac{2N_1dDc_3}{N^{1/3}}\right] \qquad (4.55)\\
&= \frac{\sqrt{d}D^2N^{2/3}}{CN_1}+\frac{CG^2}{\sqrt{d}N^{2/3}}+\frac{4Dc_1}{\sqrt{d}N^{1/3}}+\frac{4dDc_3}{N^{1/3}}\\
&\le \frac{4\sqrt{d}D^2N^{2/3}}{CN}+\frac{CG^2}{\sqrt{d}N^{2/3}}++\frac{4Dc_1}{\sqrt{d}N^{1/3}}+\frac{4dDc_3}{N^{1/3}}\\
&= \frac{1}{N^{1/3}}\left[\frac{4\sqrt{d}D^2}{C}+\frac{CG^2}{\sqrt{d}N^{1/3}}+4D\left(\frac{c_1}{\sqrt{d}}+dc_3\right)\right], \qquad (4.56)
\end{aligned}
$$

where the inequality in (4.54) follows from the fact that $N_1 \le 2(N_1 - \lceil\frac{N}{4}\rceil + 1)$, the inequality in (4.55) follows from the Lemma 21 and the final inequality follows from the fact that $\frac{N}{4} \le N_1 \le \frac{N}{2}$. We conclude by plugging (4.56) in (4.53) to obtain

$$\mathbb{E}[f(x_N)] - f(x^*)$$

$$
\begin{aligned}
&\leq \frac{1}{N^{1/3}}\left[\frac{4\sqrt{d}D^2}{C}+\frac{CG^2}{\sqrt{d}N^{1/3}}+4D\left(\frac{c_1}{\sqrt{d}}+dc_3\right)\right]+\frac{D(35\frac{c_1}{\sqrt{d}}+63dc_3)}{N^{1/3}}\\
&\quad+\frac{10CG^2}{\sqrt{d}N^{2/3}}+\frac{20Cc_1G(d^{-1}+\sqrt{d}c_3)}{N}+\frac{10C(\frac{c_1}{\sqrt{d}}+dc_3)^2}{\sqrt{d}N^{4/3}}+\frac{17.5\sqrt{d}Cc_2}{N^{1/3}}\\
&=\frac{1}{N^{1/3}}\Bigg[\frac{4\sqrt{d}D^2}{C}+\frac{11CG^2}{\sqrt{d}N^{1/3}}+D(39c_1d^{-1}+67\sqrt{d}c_3)+\frac{20Cc_1G(d^{-1/2}+dc_3)}{N^{2/3}}\\
&\quad+\frac{10C(c_1d^{-1/2}+dc_3)^2}{\sqrt{d}N}+17.5\sqrt{d}Cc_2\Bigg].
\end{aligned}
$$

□

Proof of Theorem 15

Proof. **(Theorem 15)**

Proof follows in a similar manner as that of Theorem 16 in Section 4.6.4 after setting $m_k = \infty, \forall k \geq 1$ or $c_3 = 0$. □

4.6.5 Proofs for Gaussian Smoothing method

Proof of Theorem 17

Proof. Following the proof in a similar manner as that of Proposition 1, we obtain

$$
\begin{aligned}
\mathbb{E}\left[\|\nabla f\left(x_R\right)\|^2\right] \leq \frac{1}{\sum_{k=1}^N \gamma_k}\Bigg[&\frac{2\left(f(x_1)-f(x^*)\right)}{(2-L\gamma_1)}\\
&+2\sum_{k=1}^N\left(c_1\eta_k+\frac{c_3}{\eta_k\sqrt{m_k}}\right)\left(\frac{\gamma_k+L\gamma_k^2}{2-L\gamma_k}\right)\mathbb{E}_{\xi_{[N]}}\|\nabla f\left(x_k\right)\|_1\\
&+L\sum_{k=1}^N\frac{\gamma_k^2}{(2-L\gamma_k)}\left[dc_1^2\eta_k^2+\frac{2dc_1dc_3}{\sqrt{m_k}}+\frac{c_3^2}{\eta_k^2m_k}+c_2\eta_k^2+\tilde{c_2}\right]\Bigg].
\end{aligned}
$$

Then, following the proof in a similar manner as that of Theorem 10, we obtain

$$
\begin{aligned}
\mathbb{E}\left[\|\nabla f\left(x_R\right)\|^2\right] \leq &\frac{2\left(f(x_1)-f(x^*)\right)}{N\gamma}+4B\left(c_1\eta+\frac{c_3}{\eta\sqrt{m}}\right)\\
&+L\gamma\left[dc_1^2\eta^2+\frac{2dc_1c_3}{\sqrt{m}}+\frac{dc_3^2}{\eta^2m}+c_2\eta^2+\tilde{c_2}\right].
\end{aligned}
$$

We conclude by plugging values of γ, η, and m as defined in Theorem 17 in the above equation. □

Proof of Theorem 18

Proof. Following the proof in a similar manner as that of Proposition 3, we obtain

$$\mathbb{E}\left[f\left(x_R\right)\right] - f(x^*) \leq \frac{1}{\sum_{k=1}^{N}\gamma_k}\left[\frac{D^2}{(2-L\gamma_1)}\right.$$
$$+2\sqrt{d}D\sum_{k=1}^{N}\left(c_1\eta_k + \frac{c_3}{\eta_k\sqrt{m_k}}\right)\left(\frac{\gamma_k + L\gamma_k^2}{2-L\gamma_k}\right)$$
$$\left.+\sum_{k=1}^{N}\frac{\gamma_k^2}{(2-L\gamma_k)}\left[dc_1^2\eta_k^2 + \frac{2dc_1c_3}{\sqrt{m_k}} + \frac{dc_3^2}{\eta_k^2 m_k} + c_2\eta_k^2 + \tilde{c_2}\right]\right].$$

Then, following the proof in a similar manner as that of Theorem 14, we obtain

$$\begin{aligned}
&\mathbb{E}\left[f\left(x_R\right)\right] - f(x^*) \\
&\leq \frac{D^2}{N\gamma} + 4\sqrt{d}D\left(c_1\eta + \frac{c_3}{\eta\sqrt{m}}\right) + \gamma\left[dc_1^2\eta^2 + \frac{2dc_1c_3}{\sqrt{m}} + \frac{dc_3^2}{\eta^2 m} + c_2\eta^2 + \tilde{c_2}\right] \\
&\leq \frac{LD^2}{N} + \frac{1}{\sqrt{N}}\left[\sqrt{d}D^2 + 4\sqrt{d}D\left(\frac{c_1}{d} + c_3\right) + \frac{c_1^2}{d^{3/2}N} + \frac{2c_1c_3}{\sqrt{d}N}\right. \\
&\quad \left. + \frac{\sqrt{d}c_3^2}{N} + \frac{c_2}{d^{5/2}N} + \frac{\tilde{c_2}}{\sqrt{d}}\right].
\end{aligned}$$

We conclude by plugging values of γ, η and m as defined in (4.15) in the above equation.

□

Proof of Theorem 19

The proof proceeds through a sequence of lemmas, similar to the proof of Theorem 16 in Section 4.6.4 for the simultaneous perturbation method.

Lemma 22. *Assume **(A4)** and **(A5)**. With the oracle **(O1')**, suppose that the ZSGD algorithm is run with stepsize sequence $\{\gamma_k\}_{k=1}^N$. Then, given any $1 < k_0 < k_1 \leq N$, we have*

$$\sum_{k=k_0}^{k_1} 2\gamma_k\mathbb{E}\left[f\left(x_k\right) - f(x_{k_0})\right] \leq \sum_{k=k_0}^{k_1}\left(2\sqrt{d}\gamma_k c_1\eta_k D + \gamma_k^2\mathcal{G}^2\right),$$

*where $\mathcal{G}^2 := \left[G^2 + 2\sqrt{d}c_1\eta_k G + dc_1^2\eta_k^2 + c_2\eta_k^2 + \tilde{c_2}\right]$, c_1, c_2 is as defined in **(O1')** and D is as defined in **(A5)**.*

Proof. Follows by a completely parallel argument to the proof of Lemma 20, after observing that $\mathbb{E}_{\xi_{[k]}}\left[g\left(x_k,\xi_k\right)\right] \leq \nabla f\left(x_k\right) + c_1\eta_k \mathbf{1}_{d\times 1}$, and $\mathbb{E}_{\xi_{[k]}}\left[\left\|g\left(x_k,\xi_k\right)\right\|^2\right] \leq \left\|\mathbb{E}_{\xi_{[k]}}\left[g\left(x_k,\xi_k\right)\right]\right\|^2 + c_2\eta_k^2 + \tilde{c_2}$.

□

Lemma 23. *Assume (**A4**) and (**A5**). With the oracle (**O1'**), suppose that the ZSGD algorithm is run with a constant stepsize and constant perturbation parameter, i.e., $\gamma_k = \gamma, \eta_k = \eta, \forall k \geq 1$. Then, for any $k \geq 1$, we have*

$$\sum_{k=1}^{N} \mathbb{E}\left[f\left(x_k\right) - f\left(x^*\right)\right] \leq \frac{D^2}{2\gamma} + \frac{\gamma N G^2}{2} + 2Nc_1\eta D\sqrt{d},$$

*where c_1 is as defined in **(O1)**, G is as defined in **(A4)** and D is as defined in **(A5)**.*

Proof. Proof follows in a similar manner as that of Lemma 21, with the following modification: $\mathbb{E}[\Delta g_k] = c_1\eta_k \mathbf{1}_{d\times 1}$. □

Proof. **(Theorem 19)** Using a parallel argument to the initial passage in the proof of Theorem 16 leading upto equation (4.52), we obtain

$$\begin{aligned}
&\mathbb{E}[f(x_{n_{i+1}}) - f(x_{n_i})] \\
&\leq \frac{5\sqrt{dN}}{2^{-i}C}\left(\frac{2c_1 DC2^{-2i}}{\sqrt{d}N^{3/2}} + \frac{2^{-2i}C^2}{dN}\left[G^2 + \frac{2c_1 G2^{-i}}{N} + \frac{(dc_1^2 + c_2)2^{-2i}}{dN^2} + \tilde{c_2}\right]\right) \\
&= \frac{10c_1 D2^{-i}}{N} + \frac{5C2^{-i}}{\sqrt{dN}}\left[G^2 + \frac{2c_1 G2^{-i}}{N} + \frac{(dc_1^2 + c_2)2^{-2i}}{dN^2} + \tilde{c_2}\right]. \qquad (4.57)
\end{aligned}$$

Plugging (4.57) in (4.48), we get

$$\begin{aligned}
\mathbb{E}[f(x_N)] = \mathbb{E}[f(x_{n_{l+1}})] &= \mathbb{E}[f(x_{n_0})] + \sum_{i=0}^{l}\mathbb{E}[f(x_{n_{i+1}}) - f(x_{n_i})] \\
&\leq \mathbb{E}[f(x_{n_0})] + \frac{10c_1 D}{N} + \frac{5C}{\sqrt{dN}}\left[G^2 + \frac{2c_1 G}{N} + \frac{(dc_1^2 + c_2)}{dN^2} + \tilde{c_2}\right] \\
&\quad + \sum_{i=1}^{l}\left(\frac{10c_1 D2^{-i}}{N} + \frac{5C2^{-i}}{\sqrt{dN}}\left[G^2 + \frac{2c_1 G2^{-i}}{N} + \frac{(dc_1^2 + c_2)2^{-2i}}{dN^2} + \tilde{c_2}\right]\right) \\
&\leq \mathbb{E}[f(x_{n_0})] + \frac{10c_1 D}{N} + \frac{5CG^2}{\sqrt{dN}} + \frac{10Cc_1 G}{\sqrt{d}N^{3/2}} + \frac{5C(dc_1^2 + c_2)}{d^{3/2}N^{5/2}} + \frac{5C\tilde{c_2}}{\sqrt{dN}} \\
&\quad + \left(\frac{10c_1 D}{N} + \frac{5CG^2}{\sqrt{dN}} + \frac{10Cc_1 G}{\sqrt{d}N^{3/2}} + \frac{5C(dc_1^2 + c_2)}{d^{3/2}N^{5/2}} + \frac{5C\tilde{c_2}}{\sqrt{dN}}\right)
\end{aligned}$$

$$\leq \inf_{\lceil \frac{N}{4} \rceil \leq k \leq N_1} \mathbb{E}[f(x_k)] + \frac{20c_1 D}{N} + \frac{10CG^2}{\sqrt{dN}} + \frac{20Cc_1 G}{\sqrt{d}N^{3/2}} + \frac{10C(dc_1^2 + c_2)}{d^{3/2}N^{5/2}} + \frac{10C\tilde{c_2}}{\sqrt{dN}}. \tag{4.58}$$

As in the proof of Theorem 16, we obtain

$$\begin{aligned}
\inf_{\lceil \frac{N}{4} \rceil \leq k \leq N_1} \mathbb{E}[f(x_k) - f(x^*)] &\leq \frac{1}{N_1 - \lceil \frac{N}{4} \rceil + 1} \sum_{k=\lceil \frac{N}{4} \rceil}^{N_1} \mathbb{E}[f(x_k) - f(x^*)] \\
&\leq \frac{2}{N_1} \sum_{k=1}^{N_1} \mathbb{E}[f(x_k) - f(x^*)] \\
&\leq \frac{2}{N_1} \left[\frac{D^2\sqrt{dN}}{2C} + \frac{CG^2 N_1}{2\sqrt{dN}} + \frac{2N_1 c_1 D}{N} \right] \\
&= \frac{D^2\sqrt{dN}}{CN_1} + \frac{CG^2}{\sqrt{dN}} + \frac{4c_1 D}{N} \\
&\leq \frac{4D^2\sqrt{dN}}{CN} + \frac{CG^2}{\sqrt{dN}} + \frac{4c_1 D}{N} \\
&= \frac{1}{\sqrt{N}} \left[\frac{4D^2\sqrt{d}}{C} + \frac{CG^2}{\sqrt{d}} + \frac{4c_1 D}{\sqrt{N}} \right],
\end{aligned} \tag{4.59}$$

where the second inequality follows from the fact that $N_1 \leq 2(N_1 - \lceil \frac{N}{4} \rceil + 1)$, third inequality follows from the Lemma 23 and the final inequality follows from the fact that $\frac{N}{4} \leq N_1 \leq \frac{N}{2}$. We conclude by plugging (4.59) in (4.58) to obtain

$$\begin{aligned}
&\mathbb{E}[f(x_N)] - f(x^*) \\
&\leq \frac{1}{\sqrt{N}} \left[\frac{4D^2\sqrt{d}}{C} + \frac{CG^2}{\sqrt{d}} + \frac{4c_1 D}{\sqrt{N}} \right] \\
&\quad + \frac{20c_1 D}{N} + \frac{10CG^2}{\sqrt{dN}} + \frac{20Cc_1 G}{\sqrt{d}N^{3/2}} + \frac{10C(dc_1^2 + c_2)}{d^{3/2}N^{5/2}} + \frac{10C\tilde{c_2}}{\sqrt{dN}} \\
&= \frac{1}{\sqrt{N}} \left[\frac{4D^2\sqrt{d}}{C} + \frac{11CG^2}{\sqrt{d}} + \frac{24c_1 D}{\sqrt{N}} + \frac{20Cc_1 G}{\sqrt{d}N} + \frac{10C(dc_1^2 + c_2)}{d^{3/2}N^2} + \frac{10C\tilde{c_2}}{\sqrt{d}} \right].
\end{aligned}$$

□

4.7 Simulation Experiments

4.7.1 Implementation[2]

We perform simulation experiments to evaluate the performance of the ZRSG and ZRSQN algorithm in two different settings. In the first setting, unbiased gradient/Hessian information is available to the ZRSG/ZRSQN algorithm, while in the second setting, only biased gradient/Hessian information (albeit with a controllable bias) is available. We test the performance of the ZRSG/ZRSQN algorithm on two different objective functions: (i) a support vector machine (SVM) problem that has been used earlier to test gradient-based schemes under a non-convex objective (cf. (Mason *et al.*, 2000; Ghadimi and Lan, 2013)); and (ii) a multi-modal function (Miller and Shaw, 1996) that is part of the problems library of simulation optimization toolkit[3].

We perform experiments using the GS and SP methods for estimating gradients/Hessian. We consider the following three estimation variants: (i) GS: This corresponds to the Gaussian smoothing method proposed in (Nesterov and Spokoiny, 2017); (ii) 1SPSA and 2SPSA: This corresponds to the first- and second-order SPSA algorithm (Spall, 2000) with Bernoulli perturbations; and (iii) 1RDSA-AsymBer and 2RDSA-AsymBer: This corresponds to the first- and second-order RDSA algorithm with asymmetric Bernoulli perturbations (distribution parameter ϵ is set to 0.0001, see (Prashanth *et al.*, 2017)); and (iv) 1RDSA-Perm-DP and 2RDSA-Perm-DP: This is the recently proposed first- and second-order variant of RDSA, where the perturbations are non-random, and instead use the rows of a permutation matrix (Prashanth *et al.*, 2020).

To estimate the problem parameters, namely, L, Λ, σ^2, and a bound, say α_0, on the derivative of the objective function, we use an initial i.i.d. sample of size $N_0 = 200$. We compute the l_2-norm of the Hessian of the objective function at 200 randomly selected points, by averaging over N_0 samples, and then take the maximum l_2-norm of the Hessian over these points as an estimation of L, Λ. A similar procedure has been employed in (Ghadimi and Lan, 2013). Similarly, 200 i.i.d. samples of the squared norm of stochastic gradient of the objective, and third derivative of the objective, respectively, are used to estimate σ^2 and α_0. For the SVM problem setting, the optima

[2]The implementation is available at `https://github.com/niravnb/Zeroth-Order-Stochastic-Optimization`.

[3] `http://simopt.org/wiki/index.php?title=A_Multimodal_Function`

x^* is unknown. However, using the fact that the objective has non-negative optimal values, i.e., $f(x^*) \geq 0$, we infer that $D_f \leq f(x_1)$. Using these estimates, we implement the ZRSG and ZRSQN algorithm with a stepsize and perturbation parameter chosen as mentioned in Theorem 9 and 11 for different settings.

For performance evaluation, we use the squared norm of the gradient (SNG) at x_R as the performance metric. All results are averages over 50 independent simulations.

4.7.2 (Non-convex) SVM objective function

In our first experiment, we consider the following SVM problem with a non-convex sigmoid loss function:

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \mathbb{E}_{u,v}[1 - \tanh(v\langle x, u\rangle)] + \lambda \|x\|^2 \right\}, \tag{4.60}$$

for some $\lambda > 0$. We consider synthetic data set and two real data sets, namely, heart disease and banknote authentication data set. In this experiment, we set $\lambda = 0.01$ and use 60% of the records as training data and the remaining 40% as testing data for performance evaluation.

Synthetic data set

Here, we assume that each data point (u, v) is drawn from the uniform distribution on $[0, 1]^d \times \{-1, 1\}$, where $u \in \mathbb{R}^d$ is the feature vector and $v \in \{-1, 1\}$ denotes the corresponding label.

We set the initial point to $x_1 = 5 * \bar{x}_1$, where $\bar{x}_1$ was drawn from the uniform distribution over $[0, 1]^d$. We generated data set of length 10000 using the following steps: (i) Generate a sparse vector u with 5% nonzero components following the uniform distribution on $[0, 1]^d$; (ii) Set $v = sign(\langle \bar{x}, u\rangle)$ for some $\bar{x} \in \mathbb{R}^d$ drawn from the uniform distribution on $[-1, 1]^d$. A similar procedure is employed in (Wang *et al.*, 2017; Ghadimi and Lan, 2013).

Figure 4.3 present the SNG at x_R for the ZRSG and ZRSQN algorithm with unbiased and biased gradients/Hessian for the nonconvex SVM problem (4.60) for $d = 50$. The ZRSG/ZRSQN algorithm with unbiased gradient/Hessian outperforms the other

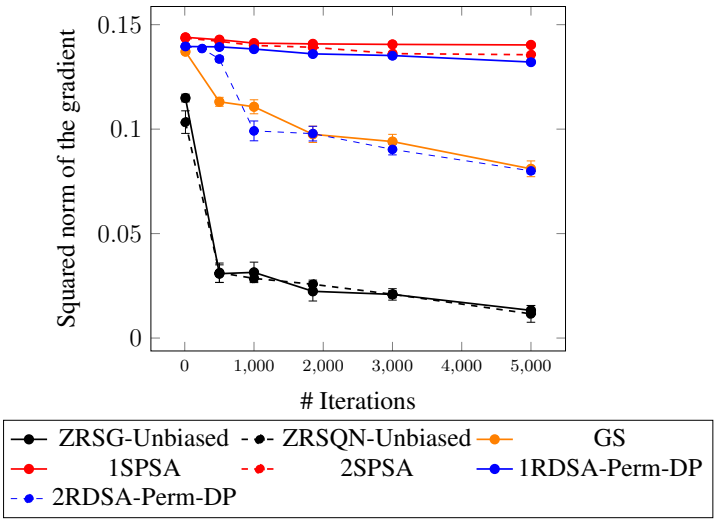


Figure 4.3: Evolution of the SNG as the iteration limit is varied, for the ZRSG and ZRSQN algorithm under the non-convex SVM problem (4.60) on synthetic dataset for $d = 50$.

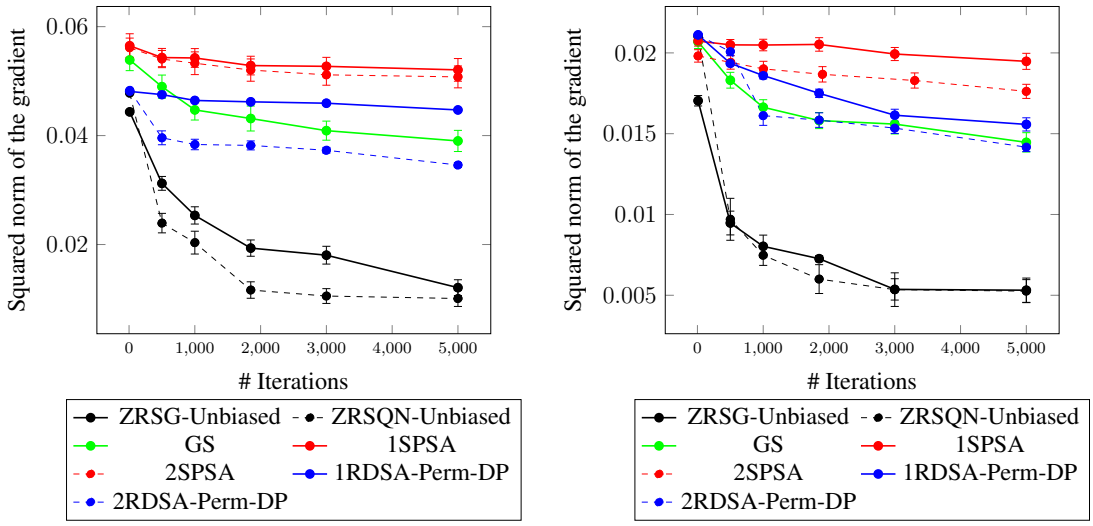
algorithms. Among the biased gradient/Hessian methods, 2RDSA-Perm-DP and GS performed best. Here, also we observe that the second-order methods perform better than their first-order counterpart. 1SPSA (resp. 2SPSA) and 1RDSA-AsymBer (resp. 2RDSA-AsymBer) exhibited similar performance. Hence, for the sake of readability, the SNG of 1RDSA-AsymBer and 2RDSA-AsymBer is not shown in the figure.

Heart Disease and Banknote Authentication Data Sets

Heart disease data set was taken from the StatLog database available in the UCI Repository[4]. It contains 270 records and 13 distinct attributes belonging to two classes: the presence or absence of heart disease. Banknote authentication data set was taken from the UCI Repository[5]. It contains 1,372 observations (banknotes) and four attributes belonging to two classes: genuine or counterfeit banknotes.

Figure 4.4a presents the SNG at x_R for the ZRSG and ZRSQN algorithms with unbiased and biased gradients/Hessian for the nonconvex SVM problem (4.60) on the heart disease data set, while Figure 4.4b compares the same algorithms on the banknote authentication data set. As expected, ZRSG/ZRSQN algorithms with unbiased gradient/Hessian information outperform the other algorithms. Among the algorithms using both (biased) gradient/Hessian information, 2RDSA-Perm-DP performed best, while

[4] `http://archive.ics.uci.edu/ml/datasets/statlog+(heart)`
[5] `https://archive.ics.uci.edu/ml/datasets/banknote+authentication`



(a) Heart disease data set

(b) Banknote authentication data set

Figure 4.4: Evolution of the SNG as the iteration limit is varied, for the ZRSG and ZRSQN algorithm under the non-convex SVM problem.

GS outperformed other algorithm that use gradients, on both datasets. For a given estimation method, for instance, Perm-DP, we observe that the quasi-Newton ZRSQN variant outperforms the gradient RSG variant.

To further evaluate algorithms' performance, we also report average classification accuracies on heart disease and banknote authentication datasets evaluated at obtained classifier x_R after 5000 iterations in Table 4.1. The result is consistent with the ones shown in the above figures, i.e., the ones with the lower SNG give a higher classification accuracy.

Table 4.1: Average classification accuracies for ZRSG and ZRSQN algorithm on heart disease and banknote authentication dataset after 5000 iterations.

Method	Heart Disease	Banknote Authentication
RSG-Unbiased	56.94	58.69
RSQN-Unbiased	57.10	58.70
GS	55.94	56.69
1SPSA	55.87	56.54
2SPSA	55.87	56.61
1RDSA-Perm-DP	55.90	56.69
2RSDA-Perm-DP	56.10	56.69

4.7.3 Multimodal Function

In our second experiment, we consider the following multimodal objective function F_2 studied in (Miller and Shaw, 1996; Xu *et al.*, 2010):

$$F_2(x_i) = \frac{\sin^6(0.05\pi x_i)}{2^{2\left(\frac{x_i-10}{80}\right)^2}}, \quad 0 \leq x_i \leq 100,$$

and define the function $F(x, \xi)$ as

$$F(x, \xi) = -\sum_{i=1}^{d} F_2(x_i) + d + \xi, \tag{4.61}$$

where $F(x, \xi)$ is the sample observation of the objective function corrupted with zero mean noise ξ. In particular, the noise is $[x^T, 1]\xi$, where ξ is a multivariate Gaussian distribution with mean zero and covariance $\sigma^2 \mathcal{I}_{(d+1)}$. A similar noise structure has been used earlier in the study of SP methods (cf.(Prashanth *et al.*, 2020; Spall, 2000)).

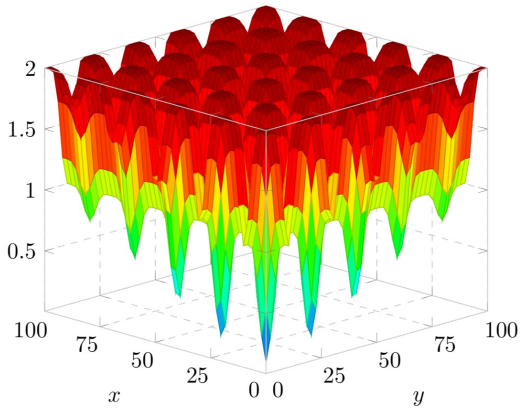
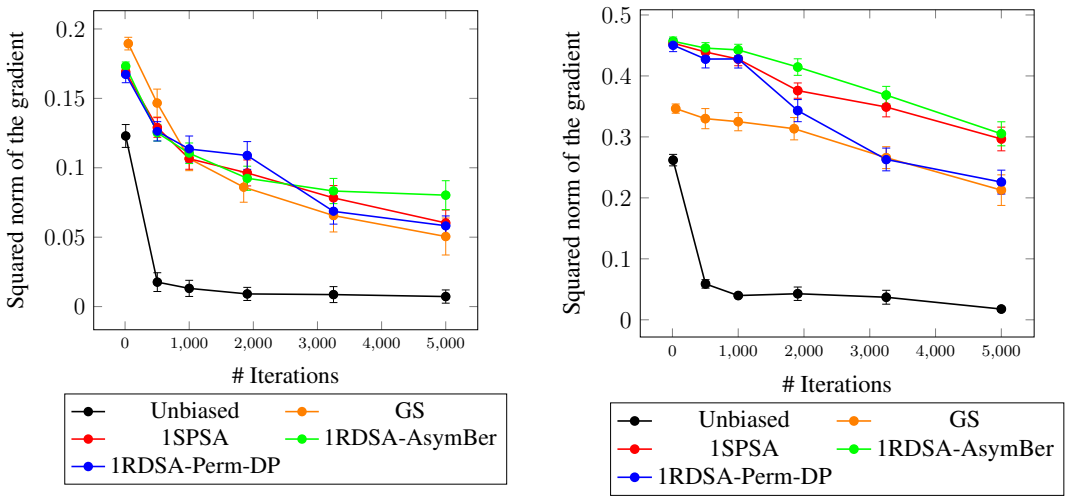


Figure 4.5: A plot of the Multimodal function (4.61), $d = 2$.

We set $\sigma = 0.3$ and use an i.i.d. sample of size $T = 10000$, to estimate the SNG at x_R for this experiment. The initial point x_1 is set to $[7, \ldots, 7]$ and the optimal point x^* is $[10, \ldots, 10]$, with $f(x^*) = \mathbb{E}_\xi[F(x^*, \xi)] = 0$. Figure 4.5 shows a plot of the multimodal function in two dimensions, and it is apparent that this objective has several widely spaced local minima.

Figure 4.6 presents the SNG at x_R for the ZRSG algorithm with unbiased and biased gradients for $d = 5$ and $d = 10$. As in the case of the non-convex SVM objective function, ZRSG algorithm with unbiased gradients outperforms the other algorithms.



(a) $d = 5$

(b) $d = 10$

Figure 4.6: Evolution of the SNG as the iteration limit is varied, for the ZRSG algorithm under the Multimodal function (4.61) with $x_1 = [7, \ldots, 7]^T$.

Among the biased gradient methods, GS performed best, and 1RDSA-Perm-DP performed on par with GS, when $d = 5$ as well as $d = 10$.

4.8 Summary

We studied gradient-based algorithms for solving stochastic convex and non-convex optimization problems when only zeroth-order information is available. In the non-convex case, we derived non-asymptotic bounds for randomized stochastic gradient and quasi-Newton algorithms in a setting where biased gradient information is made available. We also proposed and studied a variant of the biased gradient oracle, where the function measurements include estimation errors. For this oracle, we derived non-asymptotic bounds, which exhibit rates that match the oracle without estimation errors. In the convex case, we derived non-asymptotic bounds that hold in expectation for the last iterate of stochastic gradient descent algorithm, when gradient estimates with a controllable bias are provided. Our rate for the Gaussian smoothing-based oracle matches the rate obtained with unbiased gradient information.

CHAPTER 5

Conclusions and Future Work

In this thesis, we studied two problems in the context of zeroth-order stochastic optimization. In the first problem, we incorporated two novel deterministic perturbation (DP) schemes into the random directions stochastic approximation (RDSA) class of simultaneous perturbation algorithms. We proposed two new DP variants of the first-order and second-order algorithms. We have shown that the gradient and/or Hessian estimates are asymptotically unbiased, thus resulting in provably convergent 1RDSA/2RDSA variants. We also derived convergence rates to establish the superiority of the first-order and second-order algorithms, for the special case of a convex and a quadratic optimization problem, respectively. Finally, we performed numerical experiments to validate the theoretical findings.

In the second problem, we studied gradient-based algorithms for solving stochastic convex and non-convex optimization problems when given access to a stochastic zeroth-order oracle, via two techniques: simultaneous perturbation (SP), and Gaussian smoothing (GS). We also proposed an optimization oracle to capture a setting where the function measurements have an estimation error that can be controlled. We derived non-asymptotic bounds for the randomized stochastic gradient and quasi-Newton algorithms in the non-convex setting and for the last iterate of stochastic gradient descent algorithm in the convex setting, when gradient/Hessian estimates with a controllable bias are provided. In both convex and non-convex optimization setting, our bound matches the state-of-the-art complexity bounds in the literature, further, we provide a guideline for choosing the batch size for estimation, so that the overall bound matches with the one obtained when there is no estimation error. Our rate for the GS-based oracle matches the rate obtained with unbiased gradient information. Finally, we performed simulation experiments on synthetic as well as real-world datasets, and the empirical results validate the theoretical findings.

As future work, it would be interesting to derive non-asymptotic bounds for the randomized stochastic quasi-Newton algorithm with a GS-based biased gradient/Hessian

oracle. An orthogonal direction of future work is to perform an empirical investigation of stochastic gradient/Hessian schemes, with parameters chosen according to the bounds we derived in Chapter 4, on a reinforcement learning benchmark. A related empirical task is to try the deterministic perturbation variants of RDSA in sophisticated real-world applications, e.g., in transportation, networks, and service systems.

REFERENCES

1. **Balasubramanian, K.** and **S. Ghadimi**, Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates. *In Advances in Neural Information Processing Systems*. 2018.

2. **Bhatnagar, S.** (2005). Adaptive multivariate three-timescale stochastic approximation algorithms for simulation based optimization. *ACM TOMACS*, **15**(1), 74–107.

3. **Bhatnagar, S.** (2007). Adaptive Newton-based smoothed functional algorithms for simulation optimization. *ACM Transactions on Modeling and Computer Simulation*, **18**(1), 2:1–2:35.

4. **Bhatnagar, S.**, **M. C. Fu**, **S. I. Marcus**, and **I. Wang** (2003). Two-timescale simultaneous perturbation stochastic approximation using deterministic perturbation sequences. *ACM TOMACS*, **13**(2), 180–209.

5. **Bhatnagar, S.**, **H. L. Prasad**, and **L. A. Prashanth**, *Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods (Lecture Notes in Control and Information Sciences)*, volume 434. Springer, 2013.

6. **Bhatnagar, S.** and **L. A. Prashanth** (2015). Simultaneous perturbation Newton algorithms for simulation optimization. *Journal of Optimization Theory and Applications*, **164**(2), 621–643.

7. **Bottou, L.**, **F. E. Curtis**, and **J. Nocedal** (2018). Optimization methods for large-scale machine learning. *SIAM Review*, **60**(2), 223–311.

8. **Bubeck, S.** (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, **8**(3-4), 231–357.

9. **Chen, H. F.**, **L. Guo**, and **A. J. Gao** (1987). Convergence and robustness of the robbins-monro algorithm truncated at randomly varying bounds. *Stochastic Processes and their Applications*, **27**, 217–231.

10. **Chin, D. C.** (1997). Comparative study of stochastic algorithms for system optimization based on gradient approximations. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **27**(2), 244–249.

11. **Conn, A. R.**, **K. Scheinberg**, and **L. N. Vicente**, *Introduction to derivative-free optimization*, volume 8. Siam, 2009.

12. **Dalal, G.**, **B. Szorenyi**, **G. Thoppe**, and **S. Mannor**, Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning. *In Conference on Learning Theory*. 2018.

13. **Dippon, J.** and **J. Renz** (1997). Weighted means in stochastic approximation of minima. *SIAM Journal on Control and Optimization*, **35**(5), 1811–1827.

14. **Fabian, V.**, Stochastic approximation. *In Optimizing Methods in Statistics (ed. J.J.Rustagi)*. Academic Press, New York, 1971.

15. **Frikha, N.** and **S. Menozzi** (2012). Concentration bounds for stochastic approximations. *Electronic Communications in Probability*, **17**.

16. **Fu, M. C.**, *Handbook of Simulation Optimization*. Springer, 2015.

17. **Ghadimi, S.** and **G. Lan** (2013). Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, **23**(4), 2341–2368.

18. **Ghoshdastidar, D.**, **A. Dukkipati**, and **S. Bhatnagar** (2014*a*). Newton based stochastic optimization using q-gaussian smoothed functional algorithms. *Automatica*, **50**(10), 2606–2614.

19. **Ghoshdastidar, D.**, **A. Dukkipati**, and **S. Bhatnagar** (2014*b*). Smoothed functional algorithms for stochastic optimization using q-gaussian distributions. *ACM Transactions on Modeling and Computer Simulation*, **24**(3), 17:1–17:26.

20. **Goodaire, E. G.**, *Linear Algebra: Pure & Applied*. World Scientific Publishing Company, 2013.

21. **Hu, X.**, **L. A. Prashanth**, **A. György**, and **C. Szepesvári**, (bandit) convex optimization with biased noisy gradient oracles. *In Artificial Intelligence and Statistics*. 2016.

22. **Jain, P.**, **D. Nagaraj**, and **P. Netrapalli** (2019). Making the last iterate of sgd information theoretically optimal. *arXiv preprint arXiv:1904.12443*.

23. **Katkovnik, V. Y.** and **Y. Kulchitsky** (1972). Convergence of a class of random search algorithms. *Automation Remote Control*, **8**, 1321–1326.

24. **Kiefer, J.** and **J. Wolfowitz** (1952). Stochastic estimation of the maximum of a regression function. *Ann. Math. Statist.*, **23**, 462–466.

25. **Kushner, H. J.** and **D. S. Clark**, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer Verlag, New York, 1978.

26. **Mason, L.**, **J. Baxter**, **P. L. Bartlett**, and **M. R. Frean**, Boosting algorithms as gradient descent. *In Advances in neural information processing systems*. 2000.

27. **Miller, B. L.** and **M. J. Shaw**, Genetic algorithms with dynamic niche sharing for multimodal function optimization. *In Proceedings of IEEE international conference on evolutionary computation*. IEEE, 1996.

28. **Nesterov, Y.** and **V. Spokoiny** (2017). Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, **17**(2), 527–566.

29. **Prashanth, L. A.**, **S. Bhatnagar**, **N. Bhavsar**, **M. C. Fu**, and **S. I. Marcus** (2020). Random directions stochastic approximation with deterministic perturbations. *IEEE Transactions on Automatic Control*, **65**(6), 2450–2465.

30. **Prashanth, L. A.**, **S. Bhatnagar**, **M. C. Fu**, and **S. I. Marcus** (2017). Adaptive system optimization using random directions stochastic approximation. *IEEE Transactions on Automatic Control*, **62**(5), 2223–2238.

31. **Reddy, D. S.**, **L. A. Prashanth**, and **S. Bhatnagar**, Improved Hessian estimation for adaptive random directions stochastic approximation. *In IEEE Conference on Decision and Control (CDC)*. 2016.

32. **Robbins, H.** and **S. Monro** (1951). A stochastic approximation method. *Ann. Math. Statist.*, **22**, 400–407.

33. **Rubinstein, R. Y.**, *Simulation and the Monte Carlo Method*. Wiley, New York, 1981.

34. **Rubinstein, R. Y.** and **A. Shapiro**, *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*. Wiley, New York, 1993.

35. **Spall, J. C.** (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, **37**(3), 332–341.

36. **Spall, J. C.** (1997). A one-measurement form of simultaneous perturbation stochastic approximation. *Automatica*, **33**(1), 109–112.

37. **Spall, J. C.** (2000). Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Trans. Autom. Contr.*, **45**, 1839–1853.

38. **Spall, J. C.**, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, volume 65. John Wiley & Sons, 2005.

39. **Spall, J. C.** (2009). Feedback and weighting mechanisms for improving Jacobian estimates in the adaptive simultaneous perturbation algorithm. *IEEE Trans. Autom. Contr.*, **54**(6), 1216–1229.

40. **Styblinski, M. A.** and **T.-S. Tang** (1990). Experiments in nonconvex optimization: stochastic approximation with function smoothing and simulated annealing. *Neural Networks*, **3**, 467–483.

41. **Wang, X.**, **S. Ma**, **D. Goldfarb**, and **W. Liu** (2017). Stochastic quasi-Newton methods for nonconvex stochastic optimization. *SIAM Journal on Optimization*, **27**(2), 927–956.

42. **Xu, J.**, **B. L. Nelson**, and **J. Hong** (2010). Industrial strength compass: A comprehensive algorithm and software for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, **20**(1), 3.

43. **Yousefian, F.**, **A. Nedić**, and **U. Shanbhag** (2017). On stochastic and deterministic quasi-Newton methods for non-strongly convex optimization: Asymptotic convergence and rate analysis. *arXiv preprint arXiv:1710.05509*.

LIST OF PAPERS BASED ON THESIS

1. **Prashanth L. A., S. Bhatnagar, N. Bhavsar, M. C. Fu, and S. I. Marcus** (2020). Random directions stochastic approximation with deterministic perturbations. *IEEE Transactions on Automatic Control*, **65**(6), 2450-2465, doi: 10.1109/TAC.2019.2930821.

2. **N. Bhavsar and Prashanth L. A.** (2020). Non-Asymptotic Bounds for Zeroth-Order Stochastic Optimization. *Under review in a top machine learning conference, and preprint is available on arXiv:2002.11440.*