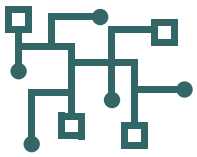


ISE

Industrial and
Systems Engineering

A Theoretical and Empirical Comparison of Gradient Approximations in Derivative-Free Optimization

A. S. Berahas[1], L. Cao[1], K. Choromanski[2], and K. Scheinberg[1]

[1]Lehigh University

[2]Google Brain

ISE Technical Report 19T-004

LEHIGH
UNIVERSITY

A Theoretical and Empirical Comparison of Gradient Approximations in Derivative-Free Optimization

A. S. Berahas* L. Cao* K. Choromanski† K. Scheinberg*‡

May 19, 2019

Abstract

In this paper, we analyze several methods for approximating the gradient of a function using only function values. These methods include finite differences, linear interpolation, Gaussian smoothing and smoothing on a unit sphere. The methods differ in the number of functions sampled, the choice of the sample points and the way in which the gradient approximations are derived. For each method, we derive bounds on the number of samples and the sampling radius which guarantee favorable convergence properties for a line search or fixed step size descent method. To this end, we derive one common condition on the accuracy of gradient approximations which guarantees these convergence properties and then show how each method can satisfy this condition. We analyze the convergence properties even when this condition is only satisfied with some sufficiently large probability at each iteration, as happens to be the case with Gaussian smoothing and smoothing on a unit sphere. Finally, we present numerical results evaluating the quality of the gradient approximations as well as their performance in conjunction with a line search derivative-free optimization algorithm.

1 Introduction

We consider an unconstrained optimization problem of the form

$$\min_{x\in\mathbb{R}^n} f(x),$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the output of some black-box procedure, e.g., the output of a simulation. In this setting, for any given $x \in \mathbb{R}^n$, one is able to obtain (at some cost) a possibly noisy evaluation of the objective function $f(x)$, or at least an unbiased estimate, but one cannot obtain explicit estimates of $\nabla f(x)$. We call this the Derivative-Free Optimization (DFO) setting [11, 16].

DFO problems arise in a plethora of science and engineering applications, e.g., engineering design, weather forecasting and molecular geometry, to mention a few. Recently there has been increased interest in applying and analyzing DFO methods for policy optimization in reinforcement learning (RL) [13, 26] as a particular case of simulation optimization. The key step in the majority of these methods is the computation of an estimate of the gradient of f. These estimates are inherently inexact for two main reasons: firstly, because they are computed using only function values, and secondly, because these function values can be noisy or stochastic. These two sources of inexactness, in principle, can be analyzed and bounded separately; in this paper we are interested in the first source of inexactness. In particular, we derive simple conditions on the accuracy of the gradient estimates under which standard optimization schemes, such as gradient descent, have fast and reliable convergence rates and we then analyze several popular methods for gradient estimation

*Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA; E-mails: albertberahas@gmail.com, liyuan@lehigh.edu, katyas@lehigh.edu

†Google Brain, New York, NY, USA; Email: kchoro@google.com

‡Corresponding author.

in terms of the cost of guaranteeing these conditions. This cost includes the number of function evaluations per iteration and potentially other costs that depend on the dimension n.

The cost of evaluating the function f varies for different DFO problems. Specifically, there are problems for which it takes hours, or even days, to evaluate f, and other problems in which it may take seconds, or less. In some settings the function can be evaluated in parallel, and the algorithms can exploit this. Methods such as finite differences, for example, require exactly $n+1$ (or $2n$, in the case of central finite difference) evaluation of f at precisely chosen sample points. Other methods, such as Gaussian smoothing [17, 21, 26] or smoothing on a unit sphere [13, 14], require as few as one or two function evaluations per iteration to obtain a (biased) stochastic gradient estimate (even for smooth deterministic functions). Of course, the variance of such estimates affects the behavior of the optimization algorithm and has to be considered in conjunction with the algorithm itself. In particular, in the case of large variance a diminishing sequence of step sizes needs to be employed, while in the case when the variance is controlled and reduced a fixed step size scheme can be employed and can result in fast convergence rates. So far most of papers that employ smoothing techniques for gradient approximation (e.g., [13, 14, 21, 26]) do so within a simple gradient descent scheme with a fixed step size parameter. It is well established that a line search method where step size is adaptive is often much more efficient than a fixed step method. However, for line search to be effective the search direction should be a descent direction at least with high enough probability, which means that the gradient estimates have to be sufficiently accurate (have low variance). Moreover, it has been observed that quasi-Newton methods, while extremely useful in accelerating optimization algorithms, are also only effective with sufficiently accurate gradient estimates. To the best of our best knowledge, there has been no systematic analysis of the accuracy of the stochastic gradient estimates used in the DFO literature (such as Gaussian smoothing smoothing on a unit sphere) specifically in conjunction with requirements of convergence of a line search algorithm.

1.1 Assumptions

It is fair to point out up front that the smoothing methods considered in this paper are primarily used for noisy or nonsmooth functions. However, in this case what is optimized is not the original function, but its smoothed version. Hence, for the sake of the clarity of the analysis, we assume Lipschitz smoothness of f. Moreover, we assume that the function evaluations of f are computed exactly. Extensions of this analysis to nonsmooth functions and functions with noise is a subject of future study and will be build upon theory derived here.

Throughout the paper we will assume that f is Lipschitz smooth.

Assumption 1.1. ***(Lipschitz continuity of the gradients of** f**)*** *The function f is continuously differentiable, and the gradient of f is L-Lipschitz continuous for all $x \in \mathbb{R}^n$.*

In some cases, to establish better approximations of the gradient, we will assume that f has Lipschitz continuous Hessians.

Assumption 1.2. ***(Lipschitz continuity of the Hessian of** f**)*** *The function f is twice continuously differentiable, and the Hessian of f is M-Lipschitz continuous for all $x \in \mathbb{R}^n$.*

We will also assume that the objective function is bounded from below.

Assumption 1.3. ***(Lower bound on** f**)*** *The function f is bounded below by a scalar $\hat{f}$.*

1.2 Summary of Results

In this paper, we first establish complexity bounds on a general line search algorithm applied to the minimization of convex, strongly convex and nonconvex functions, under the condition that the gradient estimate $g(x)$ satisfies $\|g(x) - \nabla f(x)\| \leq \theta \|\nabla f(x)\|$ for some $\theta \in [0, 1)$[1]. When this condition is not (or cannot be) satisfied

[1]The norms used in this paper are Euclidean norms.

deterministically, but can only be guaranteed with probability $1-\delta$ (for some $\delta < \frac{1}{2}$) we establish expected complexity bounds using the results in [5], where a similar line search is analyzed under a more complicated bound on $\|g(x) - \nabla f(x)\|$. For the methods in this paper, the condition $\|g(x) - \nabla f(x)\| \leq \theta \|\nabla f(x)\|$ turns out not only to be achievable, but also preferable.

We then consider four methods for approximating gradients of black-box functions and establish conditions under which $\|g(x) - \nabla f(x)\| \leq \theta \|\nabla f(x)\|$ holds either deterministically or with sufficiently high probability. Given a point x, all of these methods compute $g(x)$ using samples $f(x + \sigma u_i)$ for $i = 1, \ldots, N$, where u_i's are the *sampling directions* and σ is the *sampling radius*. The methods vary in their selection of the *number of the samples* N, the *set of directions* $\{u_i : i = 1, \ldots, N\}$ and the *sampling radius* σ.

The most straightforward way to approximate $\nabla f(x)$ is to use forward finite differences, where $N = n$ and u_i are the columns of the identity matrix I_n. Alternatively, one can approximate $\nabla f(x)$ using central finite differences where $N = 2n$ and the set of directions u_i include the columns of the matrix I_n as well as the columns of the matrix $-I_n$. In both cases, the sampling radius σ is chosen to be as small as possible, given the computational noise in the function [19]. We discuss this in more detail in the section on finite difference approximations.

As a generalization of the finite difference approach, $g(x)$ can be computed via linear interpolation (or regression). In this case $N \geq n$ and the set of directions, $\{u_i : i = 1, \ldots, N\}$, can be chosen arbitrarily, as long as they contain a set of n linearly independent directions. This method is very useful when coupled with an optimization algorithm that reuses some (or all) of the sample function values computed on prior iterations, thus avoiding the need to compute N new function values at each iteration. The accuracy of the resulting gradient approximation depends on the sampled directions u_i, specifically, on the conditioning of the matrix Q, which is the matrix whose columns are the sampling directions u_i. An extensive study of optimization methods based on interpolation gradients can be found in [11]. We discuss the key results for this method in the corresponding section.

Two methods that have become popular for estimating gradients using only function values are based on Gaussian smoothing and smoothing on a unit sphere. In principle, for these methods N can be chosen arbitrarily, the sampling directions u_i are chosen from either a standard Gaussian distribution or a uniform distribution on a unit sphere, and σ is chosen to be sufficiently small. Centralized version of these methods also can be used. We provide a detailed analysis of these methods, the resulting accuracy of their gradient estimates and appropriate choices of the number of samples N and the sampling radius σ in the corresponding section.

Here, upfront, we present a simplified summary of the lower bounds on N and the upper bounds on σ for each method that we consider in this paper, to guarantee $\|g(x) - \nabla f(x)\| \leq r$ for some $r > 0$; Table 1. Note that for the smoothing methods the bound $\|g(x) - \nabla f(x)\| \leq r$ holds with probability $1 - \delta$ and the number of samples depends on δ. We point out that the bounds on N for smoothing methods are a simplification of the more detailed bounds derived in the paper and apply when $n > 12$, while for smaller n some of the constants are larger.

Table 1: Bounds on N and σ which ensure $\|g(x) - \nabla f(x)\| \leq r$ (possibly with probability $1 - \delta$), for $n > 12$

Gradient Approximation	**# of Samples (N)**	σ
Forward Finite Differences	n	$\frac{2r}{\sqrt{n}L}$
Central Finite Differences	$2n$	$\sqrt{\frac{6r}{\sqrt{n}M}}$
Linear Interpolation	n	$\frac{2r}{\sqrt{n}L\|Q^{-1}\|_2}$
Gaussian Smoothed	$\frac{6n\|\nabla f(x)\|^2}{\delta r^2} + \frac{(2n+13)}{4\delta}$	$\frac{r}{nL}$
Centered Gaussian Smoothed	$\frac{6n\|\nabla f(x)\|^2}{\delta r^2} + \frac{(2n+26)}{36\delta}$	$\sqrt{\frac{r}{n^{3/2}M}}$
Sphere Smoothed	$\left(\frac{4n\|\nabla f(x)\|^2}{r^2} + n + \frac{4\sqrt{2}n\|\nabla f(x)\|}{3r} + \frac{2\sqrt{2}\sqrt{n}}{3}\right)\log\frac{n+1}{\delta}$	$\frac{r}{\sqrt{n}L}$
Centered Sphere Smoothed	$\left(\frac{4n\|\nabla f(x)\|^2}{r^2} + \frac{n}{9} + \frac{4\sqrt{2}n\|\nabla f(x)\|}{3r} + \frac{2\sqrt{2}\sqrt{n}}{9}\right)\log\frac{n+1}{\delta}$	$\sqrt{\frac{r}{\sqrt{n}M}}$

In our analysis we will be particularly interested in the case when $r = \theta\|\nabla f(x)\|$, in which case the bounds become as follows; see Table 2. From this table we note that the bounds on the number of samples

Table 2: Bounds on N and σ which ensure $\|g(x) - \nabla f(x)\| \leq \theta\|\nabla f(x)\|$ (possibly with probability $1 - \delta$), for $n > 12$

Gradient Approximation	**# of Samples (N)**	$\boldsymbol{\sigma}$
Forward Finite Differences	n	$\frac{2\theta\|\nabla f(x)\|}{\sqrt{n}L}$
Central Finite Differences	$2n$	$\sqrt{\frac{6\theta\|\nabla f(x)\|}{\sqrt{n}M}}$
Linear Interpolation	n	$\frac{2\theta\|\nabla f(x)\|}{\sqrt{n}L\|Q^{-1}\|}$
Gaussian Smoothed	$\frac{6n}{\delta\theta^2} + \frac{(2n+13)}{4\delta}$	$\frac{\theta\|\nabla f(x)\|}{nL}$
Centered Gaussian Smoothed	$\frac{6n}{\delta\theta^2} + \frac{(2n+26)}{36\delta}$	$\sqrt{\frac{\theta\|\nabla f(x)\|}{n^{3/2}M}}$
Sphere Smoothed	$\left(\frac{4n}{\theta^2} + n + \frac{4\sqrt{2}n}{3\theta} + \frac{2\sqrt{2}\sqrt{n}}{3}\right)\log\frac{n+1}{\delta}$	$\frac{\theta\|\nabla f(x)\|}{\sqrt{n}L}$
Centered Sphere Smoothed	$\left(\frac{4n}{\theta^2} + \frac{n}{9} + \frac{4\sqrt{2}n}{3\theta} + \frac{2\sqrt{2}\sqrt{n}}{9}\right)\log\frac{n+1}{\delta}$	$\sqrt{\frac{\theta\|\nabla f(x)\|}{\sqrt{n}M}}$

N for all methods have the same dependence (order of magnitude) on the dimension n; however, for the smoothing methods the constants in the bound are significantly larger than those for deterministic methods, such as finite differences. This suggests that deterministic methods may be more efficient, at least in the setting considered in this paper, when accurate gradient estimates are desired. The bounds on the sampling radius are comparable for the smoothing and deterministic methods, as we will discuss in detail later in the paper. Our numerical results support our theoretical observations. However, if smoothing methods are used, this paper provides conditions on N and σ that guarantee fast convergence of a line search algorithm.

Organization The paper is organized as follows. In Section 2 we present the analysis of a general gradient descent method with a line search that uses gradient approximations in lieu of the true gradient. We introduce and analyze several methods for approximating the gradient using only function values in Section 3. We present a numerical comparison of the gradient approximations and illustrate the performance of different DFO algorithms that employ these gradient approximations in 4. Finally, in Section 5, we make some concluding remarks and discuss avenues for future research.

2 Line Search Algorithms

Line search algorithms, in the DFO setting, approximate the gradient of the objective function using function values only, and compute a search direction using this gradient estimate and possibly additional information, e.g., a quasi-Newton search direction. The step size parameter is then chosen; this could be constant, selected from a predetermined sequence of step lengths (e.g., diminishing) or adaptive (e.g., via a back-tracking Armijo line search [22, Chapter 3]). A generic framework of the derivative-free line search method is given in Algorithm 1. As is clear from Algorithm 1, the key components of this method are: (i) the selection of the sample points used in the gradient approximation (Step 1); (ii) the construction of the gradient approximation (Step 2); (iii) the choice of the search direction (Step 3); and (iv) the choice of the step size parameter and the iterate update (Step 4). We describe and analyze several methods that could be used for Steps 1 and 2 in detail in Section 3.

Algorithm 1 is a generic DFO line search algorithm. For the remainder of this section, let $d_k = -g(x_k)$, although, as mentioned above, other search directions could be used. In order to prove theoretical convergence guarantees, we need to fully specify the manner in which the step size parameter is selected at every iteration

Algorithm 1: Line Search DFO Algorithm

Inputs: Starting point x_0, sampling radius σ, sample size N, initial step size parameter α_0.

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

1 **Construct a sample set $\mathcal{X}_k$:**
Choose a sample set $\mathcal{X}_k = \{x_k + \sigma u_i^k\}_{i=1}^N$, where $\{u_i^k\}_{i=1}^N \subset \mathbb{R}^n$ is the stencil, and evaluate $f(y)$ for all $y \in \mathcal{X}_k$.

2 **Gradient approximation $g(x_k)$:**
Compute an approximation $g(x_k)$ of $\nabla f(x_k)$ using the sample set $\mathcal{X}_k$.

3 **Construct a search direction d_k:**
Construct a search direction d_k, e.g., $d_k = -g(x_k)$ or $d_k = -H_k g(x_k)$.

4 **Compute step size α_k and update the iterate:**

and how a new iterate is computed (Line 4). We consider Algorithm 1 for which the step size parameter α_k varies under the condition that α_k is chosen to satisfy the sufficient decrease Armijo condition,

$$f(x_k - \alpha_k g_k) \leq f(x_k) - c_1 \alpha_k \|g(x_k)\|^2, \tag{2.1}$$

where $c_1 \in (0, 1)$ is the Armijo parameter. If a trial value α_k does not satisfy (2.1), then the iteration is called *unsuccessful*; the new iterate is set to the previous iterate, i.e., $x_{k+1} = x_k$, and the step size parameter is set to a (fixed) fraction $\tau \leq 1$ of the previous value, i.e., $\alpha_{k+1} \leftarrow \tau \alpha_k$. This step makes sense particularly when g_k (and thus d_k) are random vectors and thus can be different even for the same x_k. If the trial value satisfies (2.1), then the iteration is called *successful*, the new iterate is updated based on the search direction d_k, i.e., $x_{k+1} = x_k + \alpha_k d_k$, and the step size parameter is set to $\alpha_{k+1} \leftarrow \tau^{-1} \alpha_k$. Algorithm 2, fully specifies a subroutine for computing the step size parameter and taking a step. Note that if $\tau = 1$, Algorithm 1 is a constant step size parameter DFO line search algorithm. We discuss this further in Section 2.2.

Algorithm 2: Line Search Subroutine

Inputs: Current iterate x_k, current gradient estimate $g(x_k)$, backtracking factor $\tau \in (0, 1]$, Armijo parameter $c_1 \in (0, 1)$.

1 **Check sufficient decrease:**
Check if (2.1) is satisfied

2 **if** *Condition Satisfied* ***(successful step)*** **then**
$x_{k+1} = x_k - \alpha_k g(x_k)$ and $\alpha_{k+1} \leftarrow \tau^{-1} \alpha_k$

3 **else**
$x_{k+1} = x_k$ and $\alpha_{k+1} \leftarrow \tau \alpha_k$

Outputs: New iterate x_{k+1}, new step size parameter α_{k+1}

2.1 Convergence Analysis of Line Search Algorithms

We begin by stating a condition that is used in the analysis of the line search method, which is

$$\|g(x_k) - \nabla f(x_k)\| \leq \theta \|\nabla f(x_k)\|, \quad \text{for all} \quad k = 0, 1, 2, \ldots \tag{2.2}$$

for some $\theta \in [0, \frac{1}{2})$. This condition is referred to as a *norm condition* and was introduced and studied in [4] in the context of trust-region methods with inaccurate gradients. Note, this condition implies that $g(x_k)$ is a descent direction for the function f. Clearly, unless we know $\|\nabla f(x_k)\|$, this condition is hard or impossible to verify or guarantee. There is significant amount of work that attempts to circumvent this difficulty; see e.g., [3, 5, 23]. In [3] a practical approach to estimate $\|\nabla f(x_k)\|$ is proposed and used to ensure

some approximation of (2.2) holds. In [5] and [23], (2.2) is replaced with a condition that for some $\kappa > 0$ and for each iteration $k = 0, 1, 2, \ldots$

$$\|g(x_k) - \nabla f(x_k)\| \leq \kappa \alpha_k \|g(x_k)\|,$$

holds with probability $1 - \delta$. Under this condition convergence rate analyses are derived for a line search method that has access to deterministic function values in [5] and stochastic function values (with additional assumptions) in [23]. A simple way of making condition (2.2) realizable is to replace $\|\nabla f(x_k)\|$ with ϵ, where ϵ is the desired convergence accuracy. However, if the cost of obtaining $g(x_k)$ that satisfies $\|g(x_k) - \nabla f(x_k)\| \leq \epsilon$ increases as ϵ decreases, replacing $\|\nabla f(x_k)\|$ by its global lower bound ϵ can lead to inefficient algorithms. It turns out that for the methods we consider in this paper for approximating the gradient, condition (2.2) is relatively easy to satisfy without the knowledge of $\|\nabla f(x_k)\|$. As we show in Section 3, for deterministic functions f, only the sampling radius σ has dependence on $\|\nabla f(x_k)\|$, while the number of samples N required to compute $g(x_k)$ does not. Thus, σ can be chosen to depend on the desired accuracy ϵ, while the cost of obtaining $g(x)$ that satisfies (2.2) remains constant. In the remainder of this section, we present a convergence analysis for the generic line search algorithm (Algorithm 1). The analysis is an extension of the analysis presented in [5] under condition (2.2).

We begin by introducing several definitions, key assumptions and theoretical results (from [5]) that are required for the analysis in this paper. We then provide theoretical results for convex, strongly convex and nonconvex functions. In what follows, several of the methods (gradient approximations) used will be based on random quantities. In particular, we assume that the gradient approximations $g(x_k)$ are random and that they satisfy some notion of *good quality* (which we define explicitly shortly) with probability $1 - \delta$. For this reason we define the following quantities. Let G_k denote the random gradients, and $g(x_k) = G_k(\omega_k)$ denote a realization of the random gradient at the kth iteration. This random gradient is based on several random quantities, X_k, $\mathcal{A}_k$ and $\mathcal{D}_k$, the iterate, the step size parameter and the search direction, respectively. Realizations of these random quantities are denoted by $x_k = X_k(\omega_k)$, $\alpha_k = \mathcal{A}_k(\omega)$ and $d_k = \mathcal{D}_k(\omega_k)$.

Sufficiently accurate gradients We use the following notion of *sufficiently accurate gradients*, similar to that presented in [5].

Definition 2.1. *A sequence of random gradients $\{G_k\}$ is $(1-\delta)$-probabilistically "sufficiently accurate" for Algorithm 1 for a corresponding sequence $\{\mathcal{A}_k, X_k, \mathcal{D}_k\}$, if there exists a constant $\theta \in [0, 1)$, such that the indicator variables*

$$I_k = \mathbb{1}\{\|G_k - \nabla f(X_k)\| \leq \theta \|\nabla f(X_k)\|\}$$

satisfy the following submartingale condition

$$\mathbb{P}(I_k = 1 | \mathcal{F}_{k-1}^G) \geq 1 - \delta,$$

*where $\mathcal{F}_{k-1}^G = \sigma(G_0, \ldots, G_{k-1})$ is the σ-algebra generated by $G_0, \ldots, G_{k-1}$. Moreover, we say that iteration k is a **true** iteration is the event $I_k = 1$ occurs, otherwise the iteration is called **false**.*

For further discussions about probabilistically *sufficiently accurate* gradients (or models), we refer the reader to [5, Section 2]. We should note, that we also consider gradient approximations that are not random, e.g., finite-difference approximations. In this case, using the notation in Definition 2.1, and with a specific choice of the associated parameters, one can guarantee that every iteration is a *true* iteration, i.e., $\delta = 0$. We discuss this further in Section 2.2.

For the remainder of this section, we make the following additional assumption.

Assumption 2.2. ***(Sufficiently accurate gradients)*** *The sequence of random gradients $\{G_k\}$ generated by Algorithm 1 are $(1-\delta)$-probabilistically "sufficiently accurate" for a corresponding sequence $\{\mathcal{A}_k, X_k, \mathcal{D}_k\}$, with $\delta < 1/2$.*

Number of iterations N_ϵ to reach ϵ accuracy We derive bounds on the expected number of iterations $\mathbb{E}[N_\epsilon]$ required to reach a desired level of accuracy ϵ. Since the number of iterations N_ϵ is a random variable it can be defined as a hitting time for some stochastic process. Specifically,

- If f is convex or strongly convex: N_ϵ is the hitting time for $\{f(X_k) - f^\star \leq \epsilon\}$, i.e., the number of iterations required until $f(X_k) - f^\star \leq \epsilon$ occurs for the first time. Note, $f^\star = f(x^\star)$, where $x^\star$ is a global minimizer of f.
- If f is nonconvex: N_ϵ is the hitting time for $\{\|\nabla f(X_k)\| \leq \epsilon\}$, i.e., the number of iterations required until $\|\nabla f(X_k)\| \leq \epsilon$ occurs for the first time.

Measure of progress towards optimality and upper bound Let F_k denote a measure of progress towards optimality, and let F_ϵ be an upper bound for F_k. Specifically, we construct these two quantities as described in Table 3. Note that F_k is a nondecreasing process and F_ϵ is the largest possible value that F_k can achieve.

Table 3: Definitions of F_k and F_ϵ for convex, strongly convex and nonconvex functions.

Function	$\boldsymbol{F_k}$	$\boldsymbol{F_\epsilon}$
convex	$1/(f(X_k) - f^\star)$	$1/\epsilon$
strongly convex	$\log(1/(f(X_k) - f^\star))$	$\log(1/\epsilon)$
nonconvex	$f(X_0) - f(X_k)$	$f(X_0) - \hat{f}$

Analysis of the stochastic process Let us now consider the stochastic process $\{\mathcal{A}_k, F_k\}$ generated by Algorithm 1. Under the assumption that the models are $(1-\delta)$-probabilistically *sufficiently accurate*, each iteration is *true* with probability at least $1-\delta$, conditioned on the past.

We make the following assumption about our stochastic process (similar to that in [5]). We state it here for completeness. Let $\phi_k = F_k(\omega_k)$ be a realization of the random quantity F_k.

Assumption 2.3. *[5] There exists a constant $\bar{\alpha} > 0$ and a nondecreasing function $h(\alpha) : \mathbb{R} \to \mathbb{R}$, which satisfies $h(\alpha) > 0$, for any $\alpha > 0$, such that for any realization of Algorithm 1 the following hold for all $k < N_\epsilon$:*

1. *If iteration k is **true** (i.e., $I_k = 1$) and **successful**, then $\phi_{k+1} \geq \phi_k + h(\alpha_k)$.*
2. *If $\alpha_k \leq \bar{\alpha}$ and iteration k is **true** then it is also **successful**, which implies $\alpha_{k+1} = \tau^{-1}\alpha_k$.*
3. *$\phi_{k+1} \geq \phi_k$ for all k.*

Final bound on the expected stopping time Using the above definitions, one can bound the expected stopping time; see [5, Theorem 2.1]. For completeness we state the theorem below.

Theorem 2.4. *[5] Under the condition that $\delta > 1/2$, the hitting time N_ϵ is bounded in expectation as follows:*

$$\mathbb{E}[N_\epsilon] \leq \frac{2(1-\delta)}{(1-2\delta)^2}\left(\frac{2F_\epsilon}{h(\bar{\alpha})} + \log_\tau\left(\frac{\bar{\alpha}}{\alpha_0}\right)\right).$$

Equipped with the above definitions, assumptions and theorems, we now provide convergence guarantees for a generic DFO line search algorithm (Algorithm 1), where the step size parameter is chosen using Algorithm 2, for convex, strongly convex and nonconvex objective functions.

For each *true* iteration (i.e., $I_k = 1$), we have

$$\|g(x_k) - \nabla f(x_k)\| \leq \theta \|\nabla f(x_k)\|,$$

which implies, using the triangle inequality that

$$\|g(x_k)\| \geq (1-\theta)\|\nabla f(x_k)\|. \tag{2.3}$$

We now show that Assumption 2.3 is verified. To this end, for the three classes of functions, we show that their exists an upper bound $\bar{\alpha}$ on the step length parameter, and a function $h(\alpha)$ such that the assumption is true. First, we derive an expression for the constant $\bar{\alpha}$.

Lemma 2.5. *Let Assumption 1.1 hold. For every realization of Algorithm 1, if iteration k is **true** (i.e., $I_k = 1$), and if*

$$\alpha_k \leq \bar{\alpha} = \frac{2\,(1 - 2\theta - c_1(1-\theta))}{L(1-\theta)}, \tag{2.4}$$

*then (2.1) holds. In other words, when (2.4) holds, any **true** iteration is also a **successful** iteration.*

Proof. By Assumption 1.1, we have

$$f(x_k - \alpha_k g(x_k)) \leq f(x_k) - \alpha_k g(x_k)^T \nabla f(x_k) + \frac{\alpha_k^2 L}{2}\|g(x_k)\|^2.$$

Applying the Cauchy-Schwarz inequality, (2.2) and (2.3),

$$\begin{aligned}
f(x_k - \alpha_k g(x_k)) &\leq f(x_k) - \alpha_k g(x_k)^T \left(\nabla f(x_k) - g(x_k)\right) - \alpha_k \left(1 - \frac{\alpha_k L}{2}\right)\|g(x_k)\|^2 \\
&\leq f(x_k) + \alpha_k \|g(x_k)\|\|\nabla f(x_k) - g(x_k)\| - \alpha_k \left(1 - \frac{\alpha_k L}{2}\right)\|g(x_k)\|^2 \\
&\leq f(x_k) + \alpha_k \frac{\theta}{1-\theta}\|g(x_k)\|^2 - \alpha_k \left(1 - \frac{\alpha_k L}{2}\right)\|g(x_k)\|^2 \\
&= f(x_k) - \alpha_k \left[\frac{1-2\theta}{1-\theta} - \frac{\alpha_k L}{2}\right]\|g(x_k)\|^2.
\end{aligned}$$

From this we can conclude that (2.1) holds whenever

$$f(x_k) - \alpha_k \left[\frac{1-2\theta}{1-\theta} - \frac{\alpha_k L}{2}\right]\|g(x_k)\|^2 \leq f(x_k) - c_1 \alpha_k \|g(x_k)\|^2,$$

which is equivalent to (2.4). □

We should mention that when the error in the gradient approximation is zero, i.e., $\theta = 0$, we recover the step size parameter condition for the deterministic setting.

2.1.1 Convex Functions

In this section, we analyze the expected complexity of Algorithm 1 in the case when f is a convex function.

Assumption 2.6. ***(Bounded level sets)*** *Suppose $f \in \mathcal{C}^1(\mathbb{R}^n)$ is convex and has bounded level sets, i.e.,*

$$\|x - x^\star\| \leq D, \quad \text{for all } x \text{ with } f(x) \leq f(x_0). \tag{2.5}$$

We bound the number of iterations taken by Algorithm 1 until $f(X_k) - f^\star \leq \epsilon$ occurs. Let

$$\Delta_k^f = f(X_k) - f^\star, \tag{2.6}$$

thus, $F_k = \frac{1}{\Delta_k^f}$. By definition, N_ϵ is the number of iterations taken until $F_k \geq \frac{1}{\epsilon} = F_\epsilon$.

By Lemma 2.5, whenever $\mathcal{A}_k \leq \bar{\alpha}$, then every *true* iteration is also *successful*. What remains to be shown is that on *true* and *successful* iterations, F_k is increased by at least some function $h(\mathcal{A}_k)$, for all $k < N_\epsilon$.

Lemma 2.7. *Let Assumptions 1.1 and 2.6 hold, and consider any realization of Algorithm 1. For every iteration that is* ***true*** *and* ***successful****, we have*

$$\phi_{k+1} \geq \phi_k + \frac{c_1 \alpha_k (1-\theta)^2}{D^2}.$$

Proof. (This proof is a modification of a similar proof from [5].) By convexity, for all $x, y \in \mathbb{R}^n$ we have

$$f(x) - f(y) \geq \nabla f(y)^T (x - y).$$

Thus, if $x = x^\star$ and $y = x_k$, we have

$$-\Delta_k^f = f(x^\star) - f(x_k) \geq \nabla f(x_k)^T (x^\star - x_k) \geq -D\|\nabla f(x_k)\|,$$

where we used the Cauchy-Schwarz inequality and (2.5). Thus, when k is a *true* iteration, by (2.3) we have

$$\frac{1}{D}\Delta_k^f \leq \|\nabla f(x_k)\| \leq \frac{\|g(x_k)\|}{(1-\theta)}.$$

If k is also a *successful* iteration, then

$$\Delta_k^f - \Delta_{k+1}^f = f(x_k) - f(x_{k+1}) \geq c_1 \alpha_k \|g(x_k)\|^2 \geq \frac{c_1 \alpha_k (1-\theta)^2 (\Delta_k^f)^2}{D^2}$$

Divinding by $\Delta_k^f \Delta_{k+1}^f$, we have that for all *true* and *successful* iterations,

$$\frac{1}{\Delta_{k+1}^f} - \frac{1}{\Delta_k^f} \geq \frac{c_1 \alpha_k (1-\theta)^2 \Delta_k^f}{D^2 \Delta_{k+1}^f} \geq \frac{c_1 \alpha_k (1-\theta)^2}{D^2},$$

since $\Delta_k^f \geq \Delta_{k+1}^f$. Using the definition of ϕ_k completes the proof. □

By Lemmas 2.5 and 2.7, for any realization of Algorithm 1 (which specifies the sequence $\{\alpha_k, f_k\}$), we have:

1. If k is a *true* and *successful* iteration, then

$$\phi_{k+1} \geq \phi_k + \frac{c_1 \alpha_k (1-\theta)^2}{D^2} \quad \text{and} \quad \alpha_{k+1} = \tau^{-1} \alpha_k.$$

2. If $\alpha_k \leq \bar{\alpha}$ and iteration k is *true*, then it is also *successful*.

Hence, Assumption 2.3 holds, with $\bar{\alpha}$ defined in (2.4) and

$$h(\mathcal{A}_k) = \frac{c_1 \mathcal{A}_k (1-\theta)^2}{D^2}.$$

We now use Theorem 2.4 and the definitions of $\bar{\alpha}$, $h(\bar{\alpha})$ and F_ϵ to bound $\mathbb{E}[N_\epsilon]$.

Theorem 2.8. *Let Assumptions 1.1, 2.2 and 2.6 hold. Then, the expected number of iterations that Algorithm 1 takes until $f(X_k) - f^\star \leq \epsilon$ occurs is bounded as follows*

$$\mathbb{E}[N_\epsilon] \leq \frac{2(1-\delta)}{(1-2\delta)^2}\left(\frac{M}{\epsilon} + \log_\tau\left(\frac{2\left(1-2\theta-c_1(1-\theta)\right)}{\alpha_0 L(1-\theta)}\right)\right),$$

where $M = \frac{D^2 L}{c_1(1-2\theta-c_1(1-\theta))(1-\theta)}$.

Remark 2.9. *If $\delta = \theta = 0$ our algorithm reduces to a deterministic line search with the exact gradients. Notice that the complexity bound has two components, the first component $\frac{2D^2L}{c_1(1-c_1)\epsilon}$ achieves its minimum value, $\frac{8D^2L}{\epsilon}$, for $c_1 = 1/2$ and is similar to complexity bounds of the fixed step gradient descent for convex functions, and the second term $\log_\tau\left(\frac{2(1-c_1)}{\alpha_0 L}\right)$ bounds the total number of unsuccessful iterations, which α_k is reduced.*

2.1.2 Strongly Convex Functions

In this section, we analyze the expected complexity of Algorithm 1 in the case when f is a strongly convex function.

Assumption 2.10. ***(Strong Convexity of f)*** *There exist a positive constant μ such that*

$$f(y) \geq f(x) + \nabla f(x)^\intercal (y-x) + \frac{\mu}{2}\|x-y\|^2, \quad \text{for all } x, y \in \mathbb{R}^n.$$

Under Assumption 2.10 , let $f^\star = f(x^\star)$, where $x^\star$ is the minimizer of f.

Recall the definition of Δ_k^f (2.6). In this setting, we bound the number of iterations taken by Algorithm 1 until $\Delta_k^f \leq \epsilon$ occurs. However, in this setting the bound is logarithmic in $\frac{1}{\epsilon}$.

Lemma 2.11. *Let Assumption 2.10 hold, and consider any realization of Algorithm 1. For every iteration that is **true** and **successful**, we have*

$$f(x_k) - f(x_{k+1}) = \Delta_k^f - \Delta_{k+1}^f \geq 2\mu c_1(1-\theta)^2 \alpha_k \Delta_k^f, \tag{2.7}$$

or equivalently,

$$\Delta_{k+1}^f \leq \left(1 - 2\mu c_1(1-\theta)^2\alpha_k\right)\Delta_k^f. \tag{2.8}$$

Proof. (This proof is also a modification of a similar proof from [5].) Assumption 2.10, implies ($x = x_k$ and $y = x^\star$), that

$$\Delta_k^f \leq \frac{1}{2\mu}\|\nabla f(x_k)\|^2;$$

see [20, Theorem 2.1.10]. Equivalently, using (2.3) we have

$$\sqrt{2\mu\Delta_k^f} \leq \|\nabla f(x_k)\| \leq \frac{\|g(x_k)\|}{(1-\theta)}.$$

The final bound is attained by using (2.1). □

Note, that from (2.7) we have that if $\Delta_k^f > 0$ and $\alpha_k > \frac{1}{2\mu c_1(1-\theta)^2}$ then the iteration is unsuccessful. Hence, for an iteration to be *successful*, we must have $\alpha_k \leq \frac{1}{2\mu c_1(1-\theta)^2}$. By Lemma 2.5 we know that a *true* iteration is also *successful* if $\alpha_k \leq \bar{\alpha}$, assuming that $\bar{\alpha} \leq \frac{1}{2\mu c_1(1-\theta)^2}$. This is not a strong assumption, since the parameter c_1 can be chosen such that this is true, e.g., choosing $c_1 < 1/4$ guarantees that $\bar{\alpha} \leq \frac{1}{2\mu c_1(1-\theta)^2}$. Thus, henceforth we assume that it is true.

By Lemmas 2.5 and 2.11, for any realization of Algorithm 1 (which specifies the sequence $\{\alpha_k, f_k\}$), we have:

1. If k is a *true* and *successful* iteration, then

$$\phi_{k+1} \geq \phi_k - \log\left(1 - 2\mu c_1(1-\theta)^2\alpha_k\right) \quad \text{and} \quad \alpha_{k+1} = \tau^{-1}\alpha_k.$$

2. If $\alpha_k \leq \bar{\alpha}$ and iteration k is *true*, then it is also *successful*.

Hence, Assumption 2.3 holds, with $\bar{\alpha}$ defined in (2.4) and

$$h(\mathcal{A}_k) = -\log\left(1 - 2\mu c_1(1-\theta)^2\mathcal{A}_k\right).$$

We now use Theorem 2.4 and the definitions of $\bar{\alpha}$, $h(\bar{\alpha})$ and F_ϵ to bound $\mathbb{E}[N_\epsilon]$.

Theorem 2.12. *Let Assumptions 1.1, 2.2 and 2.6 hold. Then, the expected number of iterations that Algorithm 1 takes until $f(X_k) - f^\star \leq \epsilon$ occurs is bounded as follows*

$$\mathbb{E}[N_\epsilon] \leq \frac{2(1-\delta)}{(1-2\delta)^2}\left(2\log_{1/M}\left(\frac{1}{\epsilon}\right) + \log_\tau\left(\frac{2\left(1-2\theta-c_1(1-\theta)\right)}{\alpha_0 L(1-\theta)}\right)\right),$$

where $M = 1 - \frac{4\mu c_1(1-2\theta-c_1(1-\theta))(1-\theta)}{L}$.

Remark 2.13. *Again, if $\delta = \theta = 0$ our algorithm reduces to a deterministic line search with the exact gradients. The complexity bound has two components, $4\log_{1/M}\left(\frac{1}{\epsilon}\right)$ where $M = 1 - \frac{4\mu c_1(1-c_1)}{L}$ achieves its minimum value, $1 - \frac{\mu}{L}$, for $c_1 = 1/2$ and is similar to complexity bounds of the fixed step gradient descent for strongly convex functions, and the second term again is the bound on the the total number of unsuccessful iterations.*

2.1.3 Nonconvex Functions

In this section, we analyze the expected complexity of Algorithm 1 in the case when f is a nonconvex function. By Lemma 2.5, the sufficient decrease condition (2.1) and (2.3), for any realization of Algorithm 1 (which specifies the sequence $\{\alpha_k, f_k\}$), we have:

1. If k is a *true* and *successful* iteration, then

$$\phi_{k+1} \geq \phi_k + c_1\alpha_k(1-\theta)^2\|\nabla f(x_k)\|^2 \quad \text{and} \quad \alpha_{k+1} = \tau^{-1}\alpha_k.$$

2. If $\alpha_k \leq \bar{\alpha}$ and iteration k is *true*, then it is also *successful*.

Hence, Assumption 2.3 holds, with $\bar{\alpha}$ defined in (2.4) and

$$h(\mathcal{A}_k) = c_1\mathcal{A}_k(1-\theta)^2\epsilon^2.$$

We now use Theorem 2.4 and the definitions of $\bar{\alpha}$, $h(\bar{\alpha})$ and F_ϵ to bound $\mathbb{E}[N_\epsilon]$.

Theorem 2.14. *Let Assumptions 1.1, 1.3 and 2.2 hold. Then, the expected number of iterations that Algorithm 1 takes until $\|\nabla f(X_k)\| \leq \epsilon$ occurs is bounded as follows*

$$\mathbb{E}[N_\epsilon] \leq \frac{2(1-\delta)}{(1-2\delta)^2}\left(\frac{M}{\epsilon^2} + \log_\tau\left(\frac{2\left(1-2\theta-c_1(1-\theta)\right)}{\alpha_0 L(1-\theta)}\right)\right),$$

where $M = \frac{(f(x_0)-\hat{f})L}{c_1(1-2\theta-c_1(1-\theta))(1-\theta)}$.

Remark 2.15. *Again, if $\delta = \theta = 0$ our algorithm reduces to a deterministic line search with the exact gradients. The complexity bound has two components, $2\frac{M}{\epsilon^2}$ where $M = \frac{(f(X_0)-\hat{f})L}{c_1(1-c_1)}$ achieves its minimum value, $4(f(X_0)-\hat{f})L$, for $c_1 = 1/2$ and is similar to complexity bounds of the fixed step gradient descent for nonconvex functions, and the second term, as before, is the bound on the the total number of unsuccessful iterations.*

2.2 General Remarks

Algorithms 1 and 2 and Assumption 2.2 fully specify the requirements for a DFO line search algorithm, for which the analysis in Section 2.1 holds. Before we proceed, we make several observations with regards to the construction of the gradient estimate, the choice of the step size parameter and the error in the gradient approximation.

We presented the analysis for the case where the gradient approximations $g(x_k)$ are possibly random, e.g., Gaussian smoothed gradients [21, 26] or sphere smoothed gradients [13, 14]. However, as a special case, we recover results for gradient approximations that are not random ($\delta = 0$), e.g., finite difference approximations [2, 15] or linear interpolation gradient approximations [11].

Moreover, in the analysis we assumed that the step size parameter was chosen using an adaptive line search procedure (Algorithm 2), i.e., where the step size parameter varies at every iteration. However, our analysis also holds for a constant step size parameter procedure. Namely, if $\alpha_0 \leq \bar{\alpha}$ and $\tau = 1$, then $\alpha_k \leq \bar{\alpha}$ for all k, and all *true* iterations are also *successful* iterations. Thus, as a special case of the analysis presented in Section 2.1, we recover results for a fixed step size parameter procedure. We should note that the second term in the results in this section is zero in the case where $\tau = 1$ and $\alpha_0 = \bar{\alpha}$.

Finally, we should mention the case where we have zero error in the gradient approximation, i.e., $\theta = 0$. In this setting, we recover the results for gradient descent.

3 Gradient Approximations and Sampling

In this section, we analyze several existing methods for constructing gradient approximations using only function information. We establish conditions under which the gradient approximations constructed via these methods satisfy the bound (2.2) for any given $\theta \in [0, 1)$.

The common feature amongst these methods is that they construct approximations $g(x)$ of the gradient $\nabla f(x)$ using function values $f(y)$ for $y \in \mathcal{X}$, where $\mathcal{X}$ is a *sample set* centered around x. These methods differ in the way they select $\mathcal{X}$ and the manner in which the function values $f(y)$, on all sample points $y \in \mathcal{X}$, are used to construct $g(x)$. The methods have different costs in terms of number of evaluations of f, as well as other associated computations. Our goal is to compare these costs when computing gradient estimates that satisfy (2.2) for some $\theta \in [0, 1)$.

3.1 Gradient Estimation via Standard Finite Differences

The first method we analyze is the standard finite difference method. The forward finite difference (FFD) approximation to the gradient of f at $x \in \mathbb{R}^n$ is computed using the sample set $\mathcal{X} = \{x + \sigma e_i\}_{i=1}^n \cup \{x\}$, where $\sigma > 0$ is the finite difference interval and $e_i \in \mathbb{R}^n$ is the ith canonical vector, as follows

$$[g(x)]_i = \frac{f(x + \sigma e_i) - f(x)}{\sigma}, \quad \text{for} \quad i = 1, \ldots, n.$$

Alternatively a more precise and stable gradient approximation is obtained using central finite differences (CFD) based on the sample set $\mathcal{X} = \{x + \sigma e_i\}_{i=1}^n \cup \{x - \sigma e_i\}_{i=1}^n$, and is computed as

$$[g(x)]_i = \frac{f(x + \sigma e_i) - f(x - \sigma e_i)}{2\sigma}, \quad \text{for} \quad i = 1, \ldots, n.$$

FFD and CFD approximations require n and $2n$ functions evaluations, respectively.

We now present two standard gradient approximation bounds.

Theorem 3.1. *Under Assumption 1.1, let $g(x)$ denote the forward finite difference (FFD) approximation to the gradient $\nabla f(x)$. Then, for all $x \in \mathbb{R}^n$,*

$$\|g(x) - \nabla f(x)\| \leq \frac{\sqrt{n} L \sigma}{2}.$$

Theorem 3.2. *Under Assumption 1.2 let $g(x)$ denote the central finite difference (CFD) approximation to the gradient $\nabla f(x)$. Then, for all $x \in \mathbb{R}^n$,*

$$\|g(x) - \nabla f(x)\| \leq \frac{\sqrt{n}M\sigma^2}{6}.$$

These results show that the gradient approximation gets better as the finite difference interval σ becomes smaller. Of course, in practice, one cannot set the finite difference interval to zero, since numerical issues arise as the approximations require dividing by σ. Our goal here is to establish the largest value of σ that ensures (2.2) and hence convergence of Algorithm 1.

In the case of FFD approximation Theorem 3.1 implies that (2.2) is satisfied if

$$0 < \sigma_k \leq \frac{2\theta\|\nabla f(x_k)\|}{\sqrt{n}L}. \tag{3.1}$$

Similarly, under Assumption 1.2, Theorem 3.2 implies that the gradient estimate obtained by CFD approximation satisfies (2.2) if

$$0 < \sigma_k \leq \sqrt{\frac{6\theta\|\nabla f(x_k)\|}{\sqrt{n}M}}. \tag{3.2}$$

As we discussed in Section 2, $\|\nabla f(x_k)\|$ is typically not known and needs to be replaced by some approximation. However, for the purposes of this paper and the clarity of the presentation, we choose to use $\|\nabla f(x_k)\|$ to establish bounds on σ_k.

With σ_k chosen accordingly, we now can establish convergence of Algorithm 1 based on the FFD and CFD approximations.

Corollary 3.3. *Suppose that Assumption 1.1 holds and $g(x_k)$ is a forward finite difference (FFD) approximation to the gradient with σ_k satisfying (3.1) Then, the convergence results of Theorems 2.8, 2.12 and 2.14 hold.*

Corollary 3.4. *Suppose that Assumption 1.2 holds and $g(x_k)$ is a central finite difference (CFD) approximation to the gradient with σ_k satisfying (3.2) Then, the convergence results of Theorems 2.8, 2.12 and 2.14 hold.*

3.2 Gradient Estimation via Linear Interpolation

We now consider a more general method of approximating gradients using polynomial interpolation that has become a popular choice for model based trust region methods in the DFO setting [8, 9, 11, 17, 24, 25, 30]. These methods construct surrogate models of the objective function using interpolation or regression. While these methods are most effective when generating quadratic models around $x \in \mathbb{R}^n$ of the form

$$m(y) = f(x) + g(x)^T(y - x) + \frac{1}{2}(y - x)^T H(x)(y - x), \tag{3.3}$$

we focus on the simplest case of linear models,

$$m(y) = f(x) + g(x)^T(y - x). \tag{3.4}$$

as the focus of this paper is on line search methods, whereas the use of (3.3) requires a trust region approach due to the general nonconvexity of $m(y)$ [11].

Let us consider the following sample set $\mathcal{X} = \{x + \sigma u_1, x + \sigma u_2, \ldots, x + \sigma u_n\}$ for some $\sigma > 0$. In other words, we have n directions denoted by u_i and we sample f along those directions, around x, using a sampling radius of size σ. We assume $f(x)$ is known (function value at x). Let $F_{\mathcal{X}} \in \mathbb{R}^n$ be a vector whose

entries are $f(x+\sigma u_i) - f(x)$, for $i = 1 \dots n$, and let $Q_{\mathcal{X}} \in \mathbb{R}^{n \times n}$ define a matrix whose rows are given by u_i for $i = 1 \dots n$. Model (3.4) is constructed to satisfy the interpolation conditions,

$$f(x+\sigma u_i) = m(x+\sigma u_i), \quad \forall i = 1, \dots, n,$$

which can be written as

$$\sigma Q_{\mathcal{X}} g = F_{\mathcal{X}}.$$

If the matrix $Q_{\mathcal{X}}$ is nonsingular, then $m(y) = f(x) + g(x)^T(y-x)$, with $g(x) = \frac{1}{\sigma} Q_{\mathcal{X}}^{-1} F_{\mathcal{X}}$, is a linear interpolation model of $f(y)$ on the sample set $\mathcal{X}$.

Next we state the bound on $\|g(x) - \nabla f(x)\|$, from [10, 11]. We modify the statement of the theorem here to fit our notation.

Theorem 3.5. *[10, 11] Suppose that Assumption 1.1 holds. Let $\mathcal{X} = \{x+\sigma u_1, \dots, x+\sigma u_n\}$ be a set of interpolation points such that $\max_{1 \le i \le n} \|u_i\| \le 1$ and $Q_{\mathcal{X}}$ is nonsingular. Then,*

$$\|g(x) - \nabla f(x)\| \le \frac{\|Q_{\mathcal{X}}^{-1}\| \sqrt{n} \sigma L}{2}.$$

This result has the implication that larger $\|Q_{\mathcal{X}}^{-1}\|$ can cause large deviation of $g(x)$ from $\nabla f(x)$. Thus, it is desirable to select $\mathcal{X}$ in such a way that the condition number of $Q_{\mathcal{X}}^{-1}$ is small, which is clearly optimized when $Q_{\mathcal{X}}$ is orthonormal. Thus, we trivially recover the theorem for FFD, and moreover, extend this result to any orthonormal set of directions $\{u_1, u_2 \dots, u_n\}$, such as those used in [7].

Aside from the condition number, the important difference between general interpolation sets and orthonormal ones is in the computational cost of evaluating $g(x)$. In particular, $g(x)$ is obtained by solving a system of linear equations given by,

$$\sigma Q_{\mathcal{X}} g = F_{\mathcal{X}},$$

which in general requires $\mathcal{O}(n^3)$ computations, but that reduces to $\mathcal{O}(n^2)$ in the case of general orthornormal matrices $Q_{\mathcal{X}}$, and further reduces to $\mathcal{O}(n)$ for $Q_{\mathcal{X}} = I$, as in the case of FFD.

On the other hand, using general sample sets allows for greater flexibility (within an optimization algorithm), in particular enabling the re-use of sample points from prior iterations. When using FFD to compute $g(x)$, n function evaluations are always required, while when using interpolation it is possible to update the interpolation set by replacing only one)or a few) sample point(s) in the set $\mathcal{X}$. It is important to note that while $\mathcal{X}$ can be fairly general, the condition number of the matrix $Q_{\mathcal{X}}$ has to remain bounded for Theorem 3.5 to be useful. The sets with bounded condition number of $Q_{\mathcal{X}}$ are called *well-poised*; see [11] for details about the construction and maintenance of interpolation sets in model based trust region DFO methods.

The bounds of Theorem 3.5 are similar to those of Theorem 3.1, hence, if the sampling radius σ is chosen to satisfy

$$0 < \sigma_k \le \frac{2\theta \|\nabla f(x_k)\|}{\sqrt{n} L \|Q_{\mathcal{X}}^{-1}\|},$$

then (2.2) holds, and so do Theorems 2.8, 2.12 and 2.14. Since the condition number of $\|Q_{\mathcal{X}}^{-1}\|$ is typically not known when σ_k is chosen, a lower bound on $\|Q_{\mathcal{X}}^{-1}\|$ is used instead; see [11].

It is possible to derive an analogue of Theorem 3.2 by including n additional sample points $\{x - \sigma u_1, \dots, x - \sigma u_n\}$ in the gradient estimation procedure. Namely, two sample sets are used, $\mathcal{X}^+ = \{x + \sigma u_1, x + \sigma u_2, \dots x + \sigma u_n\}$ and $\mathcal{X}^- = \{x - \sigma u_1, x - \sigma u_2, \dots x - \sigma u_n\}$, with corresponding matrices $Q_{\mathcal{X}^+}$ and

$Q_{\mathcal{X}^-}$. The linear model $m(y) = f(x) + g^T(y-x)$ is them computed as an average of the two interpolation models, that is

$$g = \frac{g_0^+ + g_0^-}{2} = \frac{1}{2\sigma}[Q_{\mathcal{X}^+}^{-1} F_{\mathcal{X}^+} + Q_{\mathcal{X}^-}^{-1} F_{\mathcal{X}^-}].$$

The gradient estimates are computed in this way in [6], for the case of orthonormal sets and symmetric finite difference computations. Similarly to the CFD, this results in better accuracy bounds in terms of σ_k; however, this requires additional n function evaluations at each iteration, which contradicts the original idea of using interpolation as a means for reducing the per-iteration function evaluation cost.

3.3 Gradient Estimation via Gaussian Smoothing

Gaussian smoothing has recently become a popular tool for building gradient approximations using only function values. This approach has been exploited in several recent papers [17, 21, 26, 29].

Gaussian smoothing of a given function f is obtained as follows:

$$\begin{aligned} F(x) &= \mathbb{E}_{y\sim\mathcal{N}(x,\sigma^2 I)}[f(y)] = \int_{\mathbb{R}^n} f(y)\pi(y|x,\sigma^2 I)dy \\ &= \mathbb{E}_{u\sim\mathcal{N}(0,I)}[f(x+\sigma u)] = \int_{\mathbb{R}^n} f(x+\sigma u)\pi(u|0,I)du, \end{aligned} \tag{3.5}$$

where $\mathcal{N}(x,\sigma^2 I)$ denotes the multivariate normal distribution with mean x and covariance matrix $\sigma^2 I$ and $\mathcal{N}(0,I)$ denotes the standard multivariate normal distribution. The function $\pi(y|x,\Sigma)$ is the probability density function (pdf) of $\mathcal{N}(x,\Sigma)$ evaluated at y. The gradient of F can be expressed as

$$\nabla F(x) = \frac{1}{\sigma}\mathbb{E}_{u\sim\mathcal{N}(0,I)}[f(x+\sigma u)u]. \tag{3.6}$$

The following bounds hold for the error between $\nabla F(x)$ and $\nabla f(x)$. If the function f has L-Lipschitz continuous gradients, that is if Assumption 1.1 holds, then

$$\|\nabla F(x) - \nabla f(x)\| \leq \sqrt{n}L\sigma; \tag{3.7}$$

see Appendix A.1 for the proof.[2] If the function f has M-Lipschitz continuous Hessians, that is if Assumption 1.2 holds, then

$$\|\nabla F(x) - \nabla f(x))\| \leq nM\sigma^2; \tag{3.8}$$

see Appendix A.2 for proof.

Thus, to approximate $\nabla f(x)$ one can approximate $\nabla F(x)$, with sufficient accuracy, by sample average approximations applied to (3.6), i.e.,

$$g(x) = \frac{1}{N\sigma}\sum_{i=1}^{N} f(x+\sigma u_i)u_i, \tag{3.9}$$

where $u_i \sim \mathcal{N}(0,I)$ for $i = 1,2,\ldots,N$. It can be easily verified that $g(x)$ computed via (3.9) has large variance (the variance explodes as σ goes to 0). The following simple modification,

$$g(x) = \frac{1}{N}\sum_{i=1}^{N} \frac{f(x+\sigma u_i) - f(x)}{\sigma} u_i, \tag{3.10}$$

eliminates this problem and is indeed used in practice instead of (3.9). Note that the expectation of (3.10) is also $\nabla F(x)$, since $\mathbb{E}_{u\sim\mathcal{N}(\mathbf{0},I)} f(x)u$ is an all-zero vector. In what follows we will refer to $g(x)$ computed via

[2] The bound (3.7) was presented in [17] without proof; we would like to thank the first author of [17] for providing us with this proof.

(3.10) as the Gaussian smoothed gradient (GSG). As pointed out in [21], $\frac{f(x+\sigma u_i)-f(x)}{\sigma}u_i$ can be interpreted as a forward finite difference version of the directional derivative of f at x along u_i. Moreover, one can also consider the central difference variant of (3.10)–central Gaussian smoothed gradient (cGSG)–which is computed as follows,

$$g(x) = \frac{1}{2N}\sum_{i=1}^{N}\frac{f(x+\sigma u_i) - f(x-\sigma u_i)}{\sigma}u_i. \tag{3.11}$$

The properties of (3.5) and (3.10), with $N = 1$, were analyzed in [21]. However, this analysis does not explore the effect of $N > 1$ on the variance of $g(x)$. On the other hand, in [26] the authors propose an algorithm that uses GSG estimates, (3.10) and (3.11), with large samples sizes N in a fixed step size gradient descent algorithm, but without any analysis or discussion of the choices of N, σ or α. Thus, the purpose of this section is to derive bounds on the approximation error $\|g(x) - \nabla f(x)\|$ for GSG and cGSG, and to derive conditions of σ and N under which condition (2.2) holds, and thus so do the convergence results for the line search DFO algorithm based on these approximations.

We first note that there are two sources of error: (i) approximation of the true function f by the Gaussian smoothed function F, and (ii) approximation of $\nabla F(x)$ via sample average approximations. Hence, we have that

$$\begin{aligned}\|g(x) - \nabla f(x)\| &= \|(\nabla F(x) - \nabla f(x)) + (g(x) - \nabla F(x))\| \\ &\leq \|\nabla F(x) - \nabla f(x)\| + \|g(x) - \nabla F(x)\|.\end{aligned} \tag{3.12}$$

The bound on the first term is given by (3.7) or (3.8). What remains is to bound the second term $\|g(x) - \nabla F(x)\|$, the error due to the sample average approximation.

Since (3.10) (and (3.11)) is a (mini-)batch stochastic gradient estimate of $\nabla F(x)$, the probabilistic bound on $\|g(x) - \nabla F(x)\|$ is derived by bounding the expectation, which is equivalent to bounding the variance of the (mini-)batch stochastic gradient. Existing bounds in the literature, see e.g., [28], are derived under the assumption that $\|g(x) - \nabla f(x)\|$ is uniformly bounded above almost surely, which does not hold for GSG because when u follows a Gaussian distribution, $\frac{f(x+\sigma u)-f(x)}{\sigma}u$ can be arbitrarily large with positive probability. Here, we bound $\|g(x) - \nabla F(x)\|$ only under Assumptions 1.1 or 1.2. It is shown in [21] that Assumption 1.1 implies that $\nabla F(x)$ is L-Lipschitz, and that Assumption 1.1 implies that $\nabla^2 F(x)$ is M-Lipschitz.

The variance for (3.10) can be expressed as

$$\mathrm{Var}\left\{g(x)\right\} = \frac{1}{N}\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[\left(\frac{f(x+\sigma u) - f(x)}{\sigma}\right)^2 uu^T\right] - \frac{1}{N}\nabla F(x)\nabla F(x)^T, \tag{3.13}$$

and the variance of (3.11) can be expressed as

$$\mathrm{Var}\left\{g(x)\right\} = \frac{1}{N}\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[\left(\frac{f(x+\sigma u) - f(x-\sigma u)}{2\sigma}\right)^2 uu^T\right] - \frac{1}{N}\nabla F(x)\nabla F(x)^T. \tag{3.14}$$

For a normally distributed multivariate random variable $u \in \mathbb{R}^n$, we have

$$\begin{aligned}
&\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[(a^T u)^2 uu^T\right] = a^T a I + 2aa^T \\
&\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[a^T u \cdot u^T u \cdot uu^T\right] = 0_{n\times n} \\
&\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[(u^T u)^2 uu^T\right] = (n+2)(n+4)I \\
&\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[a^T u\|u\|^3\right] = 0, \\
&\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[(u^T u)^3 uu^T\right] = (n+2)(n+4)(n+8)I,
\end{aligned} \tag{3.15}$$

where a is any vector in $\mathbb{R}^n$ independent from u. We now provide bounds for the variances of GSG and cGSG under the assumption of Lipschitz continuous gradients and Hessians, respectively.

Lemma 3.6. *Under Assumption 1.1, if $g(x)$ is calculated by (3.10), then, for all $x \in \mathbb{R}^n$,*

$$\mathrm{Var}\left\{g(x)\right\} \preceq \kappa(x) I, \quad \text{where} \quad \kappa(x) = \frac{12\|\nabla f(x)\|^2 + L^2\sigma^2(n+2)(n+4)}{4N}.$$

Alternatively, under Assumption 1.2, if $g(x)$ is calculated by (3.11), then, for all $x \in \mathbb{R}^n$,

$$\mathrm{Var}\left\{g(x)\right\} \preceq \kappa(x) I, \quad \text{where} \quad \kappa(x) = \frac{108\|\nabla f(x)\|^2 + M^2\sigma^4(n+2)(n+4)(n+8)}{36N}.$$

Proof. By (3.13), we have

$$\begin{aligned}
\mathrm{Var}\left\{g(x)\right\} &= \frac{1}{N}\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[\left(\frac{f(x+\sigma u)-f(x)}{\sigma}\right)^2 uu^T\right] - \frac{1}{N}\nabla F(x)\nabla F(x)^T \\
&\preceq \frac{1}{N\sigma^2}\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[\left(\nabla f(x)^T\sigma u + \frac{1}{2}L\sigma^2 u^T u\right)^2 uu^T\right] \\
&= \frac{1}{N\sigma^2}\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[\sigma^2(\nabla f(x)^T u)^2 uu^T + L\sigma^3\nabla f(x)^T u\cdot u^T u\cdot uu^T + \frac{1}{4}L^2\sigma^4(u^T u)^2 uu^T\right] \\
&\overset{(3.15)}{=} \frac{1}{N\sigma^2}\left(\sigma^2(\nabla f(x)^T\nabla f(x) I + 2\nabla f(x)\nabla f(x)^T) + L\sigma^3\cdot 0 + \frac{1}{4}L^2\sigma^4(n+2)(n+4)I\right) \\
&\preceq \frac{12\|\nabla f(x)\|^2 + L^2\sigma^2(n+2)(n+4)}{4N} I,
\end{aligned}$$

where the first inequality comes from the Lipschitz continuity of the gradients and $-\nabla F(x)\nabla F(x)^T/N \preceq 0$, and the last inequality is due to $\nabla f(x)\nabla f(x)^T \preceq \nabla f(x)^T\nabla f(x) I$.

For cGSG, by (3.14), we have

$$\begin{aligned}
\mathrm{Var}\left\{g(x)\right\} &= \frac{1}{N}\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[\left(\frac{f(x+\sigma u)-f(x-\sigma u)}{2\sigma}\right)^2 uu^T\right] - \frac{1}{N}\nabla F(x)\nabla F(x)^T \\
&\preceq \frac{1}{N}\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[\left(\nabla f(x)^T u + \frac{M\sigma^2}{6}\|u\|^3\right)^2 uu^T\right] \\
&\overset{(3.15)}{=} \frac{1}{N}\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[\left((\nabla f(x)^T u)^2 + \frac{M^2\sigma^4}{36}\|u\|^6\right) uu^T\right] \\
&\overset{(3.15)}{=} \frac{1}{N}(\nabla f(x)^T\nabla f(x) I + 2\nabla f(x)\nabla f(x)^T) + \frac{(n+2)(n+4)(n+8)}{36N} M^2\sigma^4 I \\
&\preceq \frac{108\|\nabla f(x)\|^2 + M^2\sigma^4(n+2)(n+4)(n+8)}{36N} I,
\end{aligned}$$

where the first inequality comes from the Lipschitz continuity of the Hessians and $-\nabla F(x)\nabla F(x)^T/N \preceq 0$, and the last inequality is due to $\nabla f(x)\nabla f(x)^T \preceq \nabla f(x)^T\nabla f(x) I$. □

Using the results of Lemma 3.6, we can now bound the quantity $\|g(x) - \nabla F(x)\|$ (3.12), in probability, using Chebyshev's inequality.

Lemma 3.7. *Let F be a Gaussian smoothed approximation of f (3.5). Under Assumption 1.1, if $g(x)$ is calculated via (3.10) with sample size*

$$N \geq \frac{3n\|\nabla f(x)\|^2}{\delta r^2} + \frac{n(n+2)(n+4)L^2\sigma^2}{4\delta r^2},$$

then, for all $x \in \mathbb{R}^n$, $\|g(x) - \nabla F(x)\| \leq r$ holds with probability at least $1-\delta$, for any $r > 0$ and $0 < \delta < 1$.

Alternatively, under Assumption 1.2, if $g(x)$ is calculated via (3.11) with sample size $2N$ where

$$N \geq \frac{3n\|\nabla f(x)\|^2}{\delta r^2} + \frac{n(n+2)(n+4)(n+6)M^2\sigma^4}{36\delta r^2},$$

then, for all $x \in \mathbb{R}^n$, $\|g(x) - \nabla F(x)\| \leq r$ holds with probability at least $1-\delta$, for any $r > 0$ and $0 < \delta < 1$.

Proof. By Chebyshev's inequality, for any $r > 0$, we have

$$\mathbb{P}\left\{\sqrt{(g(x) - \nabla F(x))^T \mathrm{Var}\left\{g(x)\right\}^{-1}(g(x) - \nabla F(x))} > r\right\} \leq \frac{n}{r^2}.$$

Since $\mathrm{Var}\left\{g(x)\right\} \preceq \kappa I$, we have $\mathrm{Var}\left\{g(x)\right\}^{-1} \succeq \kappa^{-1} I$ and

$$\sqrt{(g(x) - \nabla F(x))^T \mathrm{Var}\left\{g(x)\right\}^{-1}(g(x) - \nabla F(x))} \geq \kappa^{-\frac{1}{2}}\|g(x) - \nabla F(x)\|.$$

Therefore, we have,

$$\mathbb{P}\left\{\kappa^{-\frac{1}{2}}\|g(x) - \nabla F(x)\| > r\right\} \leq \frac{n}{r^2} \implies \mathbb{P}\left\{\|g(x) - \nabla F(x)\| > r\right\} \leq \frac{\kappa n}{r^2}.$$

By Lemma 3.6, for GSG we have

$$\mathbb{P}\left\{\|g(x) - \nabla F(x)\| > r\right\} \leq \frac{3n\|\nabla f(x)\|^2}{Nr^2} + \frac{n(n+2)(n+4)L^2\sigma^2}{4Nr^2}.$$

Thus when $N \geq \frac{3n\|\nabla f(x)\|^2}{\delta r^2} + \frac{n(n+2)(n+4)L^2\sigma^2}{4\delta r^2}$, we have $\mathbb{P}\left\{\|g(x) - \nabla F(x)\| > r\right\} \leq \delta$.

For cGSG, by Lemma 3.6, we have

$$\mathbb{P}\left\{\|g(x) - \nabla F(x)\| > r\right\} \leq \frac{3n\|\nabla f(x)\|^2}{Nr^2} + \frac{n(n+2)(n+4)(n+6)M^2\sigma^4}{36Nr^2}.$$

Thus when $N \geq \frac{6n\|\nabla f(x)\|^2}{\delta r^2} + \frac{n(n+2)(n+4)(n+6)M^2\sigma^4}{36\delta r^2}$, we have $\mathbb{P}\left\{\|g(x) - \nabla F(x)\| > r\right\} \leq \delta$. □

Now with bounds for both terms in (3.12), we can bound $\|g(x) - \nabla f(x)\|$, in probability.

Theorem 3.8. *Suppose that Assumption 1.1 holds and $g(x)$ is calculated via (3.10). If*

$$N \geq \frac{3n\|\nabla f(x)\|^2}{\delta r^2} + \frac{n(n+2)(n+4)L^2\sigma^2}{4\delta r^2},$$

then, for all $x \in \mathbb{R}^n$ and $r > 0$,

$$\|g(x) - \nabla f(x)\| \leq \sqrt{n}L\sigma + r. \tag{3.16}$$

with probability at least $1-\delta$.

Alternatively, Suppose that Assumption 1.2 holds and $g(x)$ is calculated via (3.11). If

$$N \geq \frac{3n\|\nabla f(x)\|^2}{\delta r^2} + \frac{n(n+2)(n+4)(n+6)M^2\sigma^4}{36\delta r^2},$$

then, for all $x \in \mathbb{R}^n$ and $r > 0$,

$$\|g(x) - \nabla f(x)\| \leq nM\sigma^2 + r. \tag{3.17}$$

with probability at least $1-\delta$.

Proof. The proof of the first part (3.16) is a straightforward combination of the bound in (3.7) and the result of the first part of Lemma 3.7. The proof for the second part (3.17) is a straightforward combination of the bound in (3.8) and the result of the second part of Lemma 3.7. □

Using the results of Theorem 3.8, we now can derive bounds on σ and N that ensure the necessary conditions on $g(x)$ presented in Section 2, i.e., conditions under which (2.2) holds with a certain probability.

To ensure (2.2), with probability $1-\delta$, using Theorem 3.8 we want the following to hold

$$\sqrt{n}L\sigma \leq \lambda\theta\|\nabla f(x)\|, \qquad r \leq (1-\lambda)\theta\|\nabla f(x)\|,$$

for some $\lambda \in (0,1)$. Let us first consider $g(x)$ calculated via (3.10). Plugging in $\sigma = \lambda\frac{\theta\|\nabla f(x)\|}{\sqrt{n}L}$ and $r = (1-\lambda)\theta\|\nabla f(x)\|$ into the first bound on N in Theorem 3.8 we have

$$N \geq \frac{3n}{\delta\theta^2(1-\lambda)^2} + \frac{(n^2+6n+8)\lambda^2}{4\delta(1-\lambda)^2}.$$

We are interested in making the lower bound on N as small as possible and hence we are concerned with its dependence on n, when n is relatively large. Henceforth, we assume that $n > 1$ and balance the two terms on the right-hand side, in terms of the dependence on n, by setting $\lambda = \frac{1}{\sqrt{n}}$, which gives

$$N \geq \frac{3n\left(\frac{\sqrt{n}}{\sqrt{n}-1}\right)^2}{\delta\theta^2} + \frac{\left(n+6+\frac{8}{n}\right)\left(\frac{\sqrt{n}}{\sqrt{n}-1}\right)^2}{4\delta}.$$

When n is large this bound shows that the number of samples needed to ensure (2.2), with probability $1-\delta$, is roughly

$$N > \frac{3n}{\delta\theta^2}.$$

We now state the convergence result for the line search DFO algorithm where the gradient approximation is computed via (3.10).

Corollary 3.9. *Suppose that Assumption 1.1 holds and $g(x_k)$ is computed via (3.10) with N and σ_k satisfying*

$$N \geq \frac{3n\left(\frac{\sqrt{n}}{\sqrt{n}-1}\right)^2}{\delta\theta^2} + \frac{\left(n+6+\frac{8}{n}\right)\left(\frac{\sqrt{n}}{\sqrt{n}-1}\right)^2}{4\delta} \qquad \text{and} \qquad \sigma_k \leq \frac{\theta\|\nabla f(x_k)\|}{nL}.$$

Then, the convergence results of Theorems 2.8, 2.12 and 2.14 hold.

Let us compare the sampling radius σ and the number of samples N with those used by forward finite differences and interpolation to guarantee (2.2). For FFD we have $\sigma \leq \frac{\theta\|\nabla f(x)\|}{\sqrt{n}L}$, which is $\sqrt{n}$ times larger than the smoothing constant used by GSG. However, this is natural because the expected length of σu where $u \sim \mathcal{N}(0, I)$ is approximately $\sqrt{n}\sigma$; in other words, in expectation the sample points evaluated by GSG have the same distance to x as the sample points evaluated by FFD. The number of samples required by GSG is clearly larger than those required by FFD and interpolation, since the latter are fixed at $n+1$. In addition, (2.2) is guaranteed only with probability $1-\delta$ and the dependence of N on δ is high. In the next section we analyze an alternative smoothing method which has significantly better dependence on δ.

We conclude this section by deriving analogous bounds on N and σ for the case when $g(x)$ is calculated via (3.11). We plug in $\sigma = \sqrt{\lambda}\sqrt{\frac{\theta\|\nabla f(x)\|}{nM}}$ and $r = (1-\lambda)\theta\|\nabla f(x)\|$ into the second bound on N in Theorem 3.8. We have,

$$N \geq \frac{3n}{\delta\theta^2(1-\lambda)^2} + \frac{\left(n^2 + 12n + 44 + \frac{48}{n}\right)\lambda^2}{36\delta(1-\lambda)^2}.$$

We again balance the two terms of the right hand side, in terms of dependence on n, by choosing $\lambda = \frac{1}{\sqrt{n}}$, which gives us

$$N \geq \frac{3n\left(\frac{\sqrt{n}}{\sqrt{n}-1}\right)^2}{\delta\theta^2} + \frac{\left(n + 12 + \frac{44}{n} + \frac{48}{n^2}\right)\left(\frac{\sqrt{n}}{\sqrt{n}-1}\right)^2}{36\delta}.$$

Hence, the number of samples needed to ensure (2.2), with probability $1-\delta$, is roughly

$$N > \frac{3n}{\delta\theta^2}.$$

Note that this is also larger than $2n$ which is sufficient to guarantee (2.2) using gradient estimates computed via CFD. Moreover, as was the case for GSG and FFD, the sampling radius of cGSG is $\sqrt{n}$ times larger than that of CFD.

We have the following corollary for the line search DFO algorithm where the gradient approximation is computed via (3.11).

Corollary 3.10. *Suppose that Assumption 1.2 holds and $g(x_k)$ is computed via (3.11) with N and σ_k satisfying*

$$N \geq \frac{3n\left(\frac{\sqrt{n}}{\sqrt{n}-1}\right)^2}{\delta\theta^2} + \frac{\left(n + 12 + \frac{44}{n} + \frac{48}{n^2}\right)\left(\frac{\sqrt{n}}{\sqrt{n}-1}\right)^2}{36\delta} \qquad \text{and} \qquad \sigma_k \leq \sqrt{\frac{\theta\|\nabla f(x_k)\|}{n^{3/2}M}}.$$

Then, the convergence results of Theorems 2.8, 2.12 and 2.14 hold.

3.4 Gradient Estimation via Smoothing on a Sphere

Similar to the Gaussian smoothing technique, one can also smooth the function f with a uniform distribution on a ball, i.e.,

$$\begin{aligned} F(x) &= \mathbb{E}_{y\sim\mathcal{U}(\mathcal{B}(x,\sigma))}[f(y)] = \int_{\mathcal{B}(x,\sigma)} f(y)\frac{1}{V_n(\sigma)}dy \\ &= \mathbb{E}_{u\sim\mathcal{U}(\mathcal{B}(0,1))}[f(x+\sigma u)] = \int_{\mathcal{B}(0,1)} f(x+\sigma u)\frac{1}{V_n(1)}du, \end{aligned} \tag{3.18}$$

where $\mathcal{U}(\mathcal{B}(x,\sigma))$ denotes the multivariate uniform distribution on a ball centered at x of radius σ and $\mathcal{U}(\mathcal{B}(0,1))$ denotes the multivariate uniform distribution on a ball centered at 0 of radius 1. The function $V_n(\sigma)$ represents the volume of a ball in $\mathbb{R}^n$ of radius σ. It was shown in [14] that the gradient of F can be expressed as

$$\nabla F(x) = \frac{n}{\sigma}\mathbb{E}_{u\sim\mathcal{U}(\mathcal{S}(0,1))}[f(x+\sigma u)u],$$

where $\mathcal{S}(0,1)$ represents a unit sphere centered at 0 of radius 1. This leads to three ways of approximating the gradient with only function evaluations using sample average approximations

$$g(x) = \frac{n}{N\sigma}\sum_{i=1}^{N} f(x+\sigma u_i)u_i, \tag{3.19}$$

$$g(x) = \frac{n}{N}\sum_{i=1}^{N} \frac{f(x+\sigma u_i)-f(x)}{\sigma}u_i, \tag{3.20}$$

$$g(x) = \frac{n}{N}\sum_{i=1}^{N} \frac{f(x+\sigma u_i)-f(x-\sigma u_i)}{2\sigma}u_i, \tag{3.21}$$

with N independently identically distributed random vectors $\{u_i\}_{i=1}^N$ following a uniform distribution on the unit sphere. Similar to the case with Gaussian smoothing, the variance of (3.19) explodes when σ goes to zero, and thus we do not consider this formula. We analyze (3.20), which we refer to as ball smoothed gradient (BSG) and (3.21) which we refer to as central BSG (cBSG).

Again, as in the Gaussian smoothed case, there are two sources of error in the gradient approximations, and namely,

$$\|g(x) - \nabla f(x)\| \leq \|\nabla F(x) - \nabla f(x)\| + \|g(x) - \nabla F(x)\|. \tag{3.22}$$

One can bound the first term as follows; if the function f has L-Lipschitz continuous gradients, that is if Assumption 1.1 holds, then

$$\|\nabla F(x) - \nabla f(x)\| \leq L\sigma, \tag{3.23}$$

and if the function f has M-Lipschitz continuous Hessians, that is if Assumption 1.2 holds, then

$$\|\nabla F(x) - \nabla f(x)\| \leq M\sigma^2. \tag{3.24}$$

The proofs are given in Appendices A.3 and A.4, respectively.

For the second error term in (3.22), similar to the case of Gaussian smoothing, we begin with the variance of $g(x)$. The variance of (3.20) can be expressed as

$$\mathrm{Var}\{g(x)\} = \frac{n^2}{N}\mathbb{E}_{u\sim\mathcal{U}(\mathcal{S}(0,1))}\left[\left(\frac{f(x+\sigma u)-f(x)}{\sigma}\right)^2 uu^T\right] - \frac{1}{N}\nabla F(x)\nabla F(x)^T, \tag{3.25}$$

and the variance of (3.21) can be expressed as

$$\mathrm{Var}\{g(x)\} = \frac{n^2}{N}\mathbb{E}_{u\sim\mathcal{U}(\mathcal{S}(0,1))}\left[\left(\frac{f(x+\sigma u)-f(x-\sigma u)}{2\sigma}\right)^2 uu^T\right] - \frac{1}{N}\nabla F(x)\nabla F(x)^T. \tag{3.26}$$

For a uniformly distributed random variable on a sphere $\mathcal{S}(0,1)$ $u \in \mathbb{R}^n$, we have

$$\begin{aligned}
&\mathbb{E}_{u\sim\mathcal{U}(\mathcal{S}(0,1))}\left[(a^Tu)^2uu^T\right] = \frac{a^Taa+2aa^T}{n(n+2)}\\
&\mathbb{E}_{u\sim\mathcal{U}(\mathcal{S}(0,1))}\left[a^Tu\cdot u^Tu\cdot uu^T\right] = 0_{n\times n}\\
&\mathbb{E}_{u\sim\mathcal{U}(\mathcal{S}(0,1))}\left[a^Tu\|u\|^3\right] = 0\\
&\mathbb{E}_{u\sim\mathcal{U}(\mathcal{S}(0,1))}\left[(u^Tu)^2uu^T\right] = \frac{1}{n}I\\
&\mathbb{E}_{u\sim\mathcal{U}(\mathcal{S}(0,1))}\left[(u^Tu)^3uu^T\right] = \frac{1}{n}I,
\end{aligned} \tag{3.27}$$

where a is any vector in $\mathbb{R}^n$ independent from u. We now provide bounds for the variances of BSG and cBSG under the assumption of Lipschitz continuous gradients and Hessians, respectively.

Lemma 3.11. *Under Assumption 1.1, if $g(x)$ is calculated by (3.20), then, for all $x \in \mathbb{R}^n$,*

$$\mathrm{Var}\{g(x)\} \preceq \kappa(x) I, \quad \text{where} \quad \kappa(x) = \frac{\frac{12n}{n+2}\|\nabla f(x)\|^2 + L^2\sigma^2 n}{4N}.$$

Alternatively, under Assumption 1.2, if $g(x)$ is calculated by (3.21), then, for all $x \in \mathbb{R}^n$,

$$\mathrm{Var}\{g(x)\} \preceq \kappa(x) I, \quad \text{where} \quad \kappa(x) = \frac{\frac{108n}{n+2}\|\nabla f(x)\|^2 + M^2\sigma^4 n}{36N}.$$

Proof. By (3.25), we have

$$\begin{aligned}
\mathrm{Var}\{g(x)\} &= \frac{n^2}{N}\mathbb{E}_{u\sim\mathcal{U}(\mathcal{S}(0,1))}\left[\left(\frac{f(x+\sigma u)-f(x)}{\sigma}\right)^2 uu^T\right] - \frac{1}{N}\nabla F(x)\nabla F(x)^T \\
&\preceq \frac{n^2}{N\sigma^2}\mathbb{E}_{u\sim\mathcal{U}(\mathcal{S}(0,1))}\left[\left(\nabla f(x)^T\sigma u + \frac{1}{2}L\sigma^2 u^T u\right)^2 uu^T\right] \\
&= \frac{n^2}{N\sigma^2}\mathbb{E}_{u\sim\mathcal{U}(\mathcal{S}(0,1))}\left[\sigma^2(\nabla f(x)^T u)^2 uu^T + L\sigma^3\nabla f(x)^T u\cdot u^T u\cdot uu^T + \frac{1}{4}L^2\sigma^4(u^T u)^2 uu^T\right] \\
&\overset{(3.27)}{=} \frac{n^2}{N\sigma^2}\left(\frac{\sigma^2}{n(n+2)}(\nabla f(x)^T\nabla f(x) I + 2\nabla f(x)\nabla f(x)^T) + L\sigma^3\cdot 0 + \frac{L^2\sigma^4}{4n} I\right) \\
&\preceq \frac{1}{N}\left(\frac{3n}{n+2}\|\nabla f(x)\|^2 + \frac{nL^2\sigma^2}{4}\right) I,
\end{aligned}$$

where the first inequality comes from the L-Lipschitz continuity of the gradients and $-\nabla F(x)\nabla F(x)^T/N \preceq 0$, and the last inequality is due to $\nabla f(x)\nabla f(x)^T \preceq \nabla f(x)^T\nabla f(x) I$.

For cBSG, by (3.26), we have

$$\begin{aligned}
\mathrm{Var}\{g(x)\} &= \frac{n^2}{N}\mathbb{E}_{u\sim\mathcal{U}(\mathcal{S}(0,1))}\left[\left(\frac{f(x+\sigma u)-f(x-\sigma u)}{2\sigma}\right)^2 uu^T\right] - \frac{1}{N}\nabla F(x)\nabla F(x)^T \\
&\preceq \frac{n^2}{N}\mathbb{E}_{u\sim\mathcal{U}(\mathcal{S}(0,1))}\left[\left(\nabla f(x)^T u + \frac{M\sigma^2}{6}\|u\|^3\right)^2 uu^T\right] \\
&\overset{(3.27)}{=} \frac{n^2}{N}\mathbb{E}_{u\sim\mathcal{U}(\mathcal{S}(0,1))}\left[\left((\nabla f(x)^T u)^2 + \frac{M^2\sigma^4}{36}\|u\|^6\right) uu^T\right] \\
&\overset{(3.27)}{=} \frac{n}{N(n+2)}\left(\nabla f(x)^T\nabla f(x) I + 2\nabla f(x)\nabla f(x)^T\right) + \frac{n}{36N}M^2\sigma^4 I \\
&\preceq \frac{1}{N}\left(\frac{3n}{n+2}\|\nabla f(x)\|^2 + \frac{nM^2\sigma^4}{36}\right) I,
\end{aligned}$$

where the first inequality comes from the Lipschitz continuity of the Hessians and $-\nabla F(x)\nabla F(x)^T/N \preceq 0$, and the last inequality is due to $\nabla f(x)\nabla f(x)^T \preceq \nabla f(x)^T\nabla f(x) I$. □

Using the results of Lemma 3.11, we can bound the quantity $\|g(x) - \nabla F(x)\|$ (3.22), with probability $1-\delta$, using Chebyshev's inequality, just as we did in the case of GSG. However, ball smoothed gradient approach has a significant advantage over Gaussian smoothing in that it allows the use of Bernstein's inequality [28] instead of Chebychev's and the resulting bound on N has a significantly improved dependence on δ.

Bernstein's inequality applies here, because unlike GSG (and cGSG), BSG (and cBSG) enjoys a deterministic bound on the error term $n\frac{f(x+\sigma u)-f(x)}{\sigma}u - F(x)$; see proof of Lemma 3.12.

Lemma 3.12. *Let F be a sphere smoothed approximation of f (3.18). Under Assumption 1.1, if $g(x)$ is calculated via (3.20) with sample size*

$$N \geq \left(\frac{2n}{r^2} \left(\|\nabla f(x)\|^2 + \frac{nL^2\sigma^2}{4} \right) + \frac{2n}{3r} \left(2\|\nabla f(x)\| + L\sigma \right) \right) \log \frac{n+1}{\delta},$$

then, for all $x \in \mathbb{R}^n$, $\|g(x) - \nabla F(x)\| \leq r$ holds with probability at least $1-\delta$, for any $r > 0$ and $0 < \delta < 1$.

Alternatively, under Assumption 1.2 if $g(x)$ is calculated via (3.21) with sample size $2N$ where

$$N \geq \left[\frac{2n}{r^2} \left(\|\nabla f(x)\|^2 + \frac{nM^2\sigma^4}{36} \right) + \frac{2n}{3r} \left(2\|\nabla f(x)\| + \frac{M\sigma^2}{3} \right) \right] \log \frac{n+1}{\delta},$$

then, for all $x \in \mathbb{R}^n$, $\|g(x) - \nabla F(x)\| \leq r$ holds with probability at least $1-\delta$, for any $r > 0$ and $0 < \delta < 1$.

Proof. We first note that

$$\mathbb{E}_{u_i \sim \mathcal{U}(\mathcal{S}(0,1))} \left[\frac{n}{N} \frac{f(x+\sigma u_i) - f(x)}{\sigma} u_i - \frac{1}{N} F(x) \right] = 0,$$

and

$$\begin{aligned}
\left\| \frac{n}{N} \frac{f(x+\sigma u_i) - f(x)}{\sigma} u_i - \frac{1}{N} F(x) \right\| &\leq \frac{n}{N\sigma} \mathbb{E}_{u_i \sim \mathcal{U}(\mathcal{S}(0,1))} \left[|f(x+\sigma u_i) - f(x)| \, \|u_i\| + |f(x+\sigma u_i) - f(x)| \, \|u_i\| \right] \\
&\leq \frac{n}{N\sigma} \mathbb{E}_{u_i \sim \mathcal{U}(\mathcal{S}(0,1))} \left[\nabla f(x)^T \sigma u_i + \frac{L\|\sigma u_i\|^2}{2} + \nabla f(x)^T \sigma u_i + \frac{L\|\sigma u_i\|^2}{2} \right] \\
&\leq \frac{n}{N} (2\|\nabla f(x)\| + L\sigma),
\end{aligned}$$

for all $u_i \sim \mathcal{U}(\mathcal{S}(0,1))$. The *matrix variance statistic* of $g(x) - \nabla F(x)$ is

$$\begin{aligned}
v(g(x) - \nabla F(x)) &= \max \left\{ \mathbb{E} \left[(g(x) - \nabla F(x))(g(x) - \nabla F(x))^T \right], \mathbb{E} \left[(g(x) - \nabla F(x))^T (g(x) - \nabla F(x)) \right] \right\} \\
&\leq \max \left\{ \frac{1}{N} \left(\frac{3n}{n+2} \|\nabla f(x)\|^2 + \frac{nL^2\sigma^2}{4} \right), \frac{n}{N} \left(\|\nabla f(x)\|^2 + \frac{nL^2\sigma^2}{4} \right) \right\} \\
&= \frac{n}{N} \left(\|\nabla f(x)\|^2 + \frac{nL^2\sigma^2}{4} \right).
\end{aligned}$$

By Bernstein's inequality, we have

$$\begin{aligned}
\mathbb{P}(\|g(x) - \nabla F(x)\| \geq r) &\leq (n+1) \exp \left(\frac{-r^2/2}{v(g(x) - \nabla F(x)) + \frac{nr}{3N}(2\|\nabla f(x)\| + L\sigma)} \right) \\
&\leq (n+1) \exp \left(\frac{-r^2/2}{\frac{n}{N} \left(\|\nabla f(x)\|^2 + \frac{nL^2\sigma^2}{4} \right) + \frac{nr}{3N}(2\|\nabla f(x)\| + L\sigma)} \right).
\end{aligned}$$

In order to ensure that $\mathbb{P}(\|g(x) - \nabla F(x)\| \geq r) \leq \delta$, for some $\delta \in (0,1)$, we require that

$$(n+1) \exp \left(\frac{-r^2/2}{\frac{n}{N} \left(\|\nabla f(x)\|^2 + \frac{nL^2\sigma^2}{4} \right) + \frac{nr}{3N}(2\|\nabla f(x)\| + L\sigma)} \right) \leq \delta,$$

from which we conclude that

$$N \geq \left[\frac{2n}{r^2} \left(\|\nabla f(x)\|^2 + \frac{nL^2\sigma^2}{4} \right) + \frac{2n}{3r} \left(2\|\nabla f(x)\| + L\sigma \right) \right] \log \frac{n+1}{\delta}.$$

For the cBSG case, note that

$$\mathbb{E}_{u_i\sim\mathcal{U}(\mathcal{S}(0,1))}\left[\frac{n}{N}\frac{f(x+\sigma u_i)-f(x-\sigma u_i)}{2\sigma}u_i-\frac{1}{N}F(x)\right]=0,$$

and

$$\begin{aligned}\left\|\frac{n}{N}\frac{f(x+\sigma u_i)-f(x-\sigma u_i)}{2\sigma}u_i-\frac{1}{N}F(x)\right\| &\leq \frac{n}{2N\sigma}\mathbb{E}_{u_i\sim\mathcal{U}(\mathcal{S}(0,1))}[|f(x+\sigma u_i)-f(x-\sigma\epsilon)|\,\|u_i\| \\ &\quad +|f(x+\sigma u_i)-f(x-\sigma u_i)|\,\|u_i\|] \\ &\leq \frac{n}{2N\sigma}\mathbb{E}_{u\sim\mathcal{U}(\mathcal{S}(0,1))}\left[2\nabla f(x)^T\sigma u_i+\frac{M\|\sigma u_i\|^3}{3}\right. \\ &\quad \left.+2\nabla f(x)^T\sigma u_i+\frac{M\|\sigma u_i\|^3}{3}\right] \\ &\leq \frac{n}{N}\left(2\|\nabla f(x)\|+\frac{M\sigma^2}{3}\right),\end{aligned}$$

for all $u_i\sim\mathcal{U}(\mathcal{S}(0,1))$. The *matrix variance statistic* of $g(x)-\nabla F(x)$ is

$$\begin{aligned}v(g(x)-\nabla F(x)) &= \max\left\{\mathbb{E}\left[(g(x)-\nabla F(x))(g(x)-\nabla F(x))^T\right],\mathbb{E}\left[(g(x)-\nabla F(x))^T(g(x)-\nabla F(x))\right]\right\} \\ &\leq \max\left\{\frac{1}{N}\left(\frac{3n}{n+2}\|\nabla f(x)\|^2+\frac{nM^2\sigma^4}{36}\right),\frac{n}{N}\left(\|\nabla f(x)\|^2+\frac{nM^2\sigma^4}{36}\right)\right\} \\ &= \frac{n}{N}\left(\|\nabla f(x)\|^2+\frac{nM^2\sigma^4}{36}\right).\end{aligned}$$

By Bernstein's inequality, we have

$$\begin{aligned}\mathbb{P}(\|g(x)-\nabla F(x)\|\geq r) &\leq (n+1)\exp\left(\frac{-r^2/2}{v(g(x)-\nabla F(x))+\frac{nr}{3N}(2\|\nabla f(x)\|+M\sigma^2/3)}\right) \\ &\leq (n+1)\exp\left(\frac{-r^2/2}{\frac{n}{N}\left(\|\nabla f(x)\|^2+\frac{nM^2\sigma^4}{36}\right)+\frac{nr}{3N}(2\|\nabla f(x)\|+M\sigma^2/3)}\right).\end{aligned}$$

In order to ensure that $\mathbb{P}(\|g(x)-\nabla F(x)\|\geq r)\leq\delta$, for some $\delta\in(0,1)$, we require that

$$(n+1)\exp\left(\frac{-r^2/2}{\frac{n}{N}\left(\|\nabla f(x)\|^2+\frac{nM^2\sigma^4}{36}\right)+\frac{nr}{3N}(2\|\nabla f(x)\|+M\sigma^2/3)}\right)\leq\delta$$

from which we conclude that

$$N\geq\left[\frac{2n}{r^2}\left(\|\nabla f(x)\|^2+\frac{nM^2\sigma^4}{36}\right)+\frac{2n}{3r}\left(2\|\nabla f(x)\|+\frac{M\sigma^2}{3}\right)\right]\log\frac{n+1}{\delta}.$$

□

Now, with bounds for both terms in (3.22), we can bound $\|g(x)-\nabla f(x)\|$, in probability.

Theorem 3.13. *Suppose that Assumption 1.1 holds and $g(x)$ is calculated via (3.20). If*

$$N\geq\left(\frac{2n}{r^2}\left(\|\nabla f(x)\|^2+\frac{nL^2\sigma^2}{4}\right)+\frac{2n}{3r}\left(2\|\nabla f(x)\|+L\sigma\right)\right)\log\frac{n+1}{\delta},$$

then, for all $x\in\mathbb{R}^n$ and $r>0$,

$$\|g(x)-\nabla f(x)\|\leq L\sigma+r. \tag{3.28}$$

with probability at least $1-\delta$.

Alternatively, suppose that Assumption 1.2 holds and $g(x)$ is calculated via (3.21). If

$$N \geq \left[\frac{2n}{r^2}\left(\|\nabla f(x)\|^2 + \frac{nM^2\sigma^4}{36}\right) + \frac{2n}{3r}\left(2\|\nabla f(x)\| + \frac{M\sigma^2}{3}\right)\right]\log\frac{n+1}{\delta},$$

then, for all $x \in \mathbb{R}^n$ and $r > 0$,

$$\|g(x) - \nabla f(x)\| \leq M\sigma^2 + r. \tag{3.29}$$

with probability at least $1 - \delta$.

Proof. The proof for the first part (3.28) is a straightforward combination of the bound in (3.23) and the result of the first part of Lemma 3.12. The proof for the second part (3.29) is a straightforward combination of the bound in (3.24) and the result of the second part of Lemma 3.12. □

In Theorem 3.13 one should notice the improved dependence of the sample size N on the probability δ as compared to Theorem 3.8. We should note again that we cannot derive similar results for Gaussian smoothed gradient approximations because when u is Gaussian the term $\frac{f(x+\sigma u)-f(x)}{\sigma}u$ can be arbitrarily large with positive probability.

We now state the convergence results, similar to Corollaries 3.9 and 3.10, for the line search DFO algorithm where the gradient approximations are computed via (3.20) and (3.21), respectively. For brevity, we skip the derivations and just state the results.

Corollary 3.14. *Suppose that Assumption 1.1 holds and $g(x_k)$ is computed via (3.20) with N and σ_k satisfying*

$$N \geq \left(\frac{2n\left(\frac{\sqrt{n}}{\sqrt{n}-1}\right)^2}{\theta^2} + \frac{n\left(\frac{\sqrt{n}}{\sqrt{n}-1}\right)^2}{2} + \frac{4n\left(\frac{\sqrt{n}}{\sqrt{n}-1}\right)}{3\theta} + \frac{2\sqrt{n}\left(\frac{\sqrt{n}}{\sqrt{n}-1}\right)}{3}\right)\log\frac{n+1}{\delta} \tag{3.30}$$

$$\text{and} \qquad \sigma_k \leq \frac{\theta\|\nabla f(x_k)\|}{\sqrt{n}L}. \tag{3.31}$$

Then, the convergence results of Theorems 2.8, 2.12 and 2.14 hold.

Corollary 3.15. *Suppose that Assumption 1.2 holds and $g(x_k)$ is computed via (3.21) with N and σ_k satisfying*

$$N \geq \left(\frac{2n\left(\frac{\sqrt{n}}{\sqrt{n}-1}\right)^2}{\theta^2} + \frac{n\left(\frac{\sqrt{n}}{\sqrt{n}-1}\right)^2}{18} + \frac{4n\left(\frac{\sqrt{n}}{\sqrt{n}-1}\right)}{3\theta} + \frac{2\sqrt{n}\left(\frac{\sqrt{n}}{\sqrt{n}-1}\right)}{9}\right)\log\frac{n+1}{\delta} \tag{3.32}$$

$$\text{and} \qquad \sigma_k \leq \sqrt{\frac{\theta\|\nabla f(x_k)\|}{\sqrt{n}M}}. \tag{3.33}$$

Then, the convergence results of Theorems 2.8, 2.12 and 2.14 hold.

4 Numerical Results

In this section, we test our theoretical conclusions via numerical experiments.

4.1 Gradient Approximation Accuracy

First, we compare the numerical accuracy of the gradient approximations obtained by the methods discussed in Section 3. We compare the resulting θ, which is the relative error

$$\frac{\|\nabla f(x) - g\|}{\|\nabla f(x)\|}, \tag{4.1}$$

and report the average log of the relative error, i.e., $\log_{10}\theta$. Theory dictates that an optimization algorithm will converge if $\log_{10}\theta$ is negative, bounded away from zero, with sufficiently high probability.

Gradient estimation on a synthetic function We first conduct tests on a synthetic function,

$$f(x) = \left(\sum_{i=1}^{n/2} M\sin(x_{2i-1}) + \cos(x_{2i})\right) + \frac{L-M}{2n}x^T 1_{n\times n}x, \tag{4.2}$$

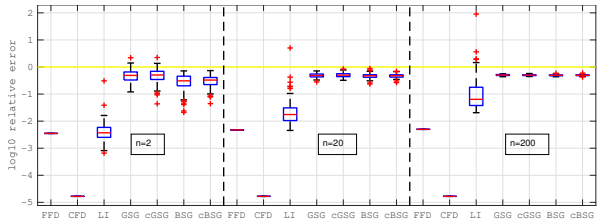
where n is an even number denoting the input dimension, $1_{n\times n}$ denotes an n by n matrix of all ones, and $L > M > 0$. We approximate the gradient of f at the origin, for which $\|\nabla f(0)\| = \sqrt{\frac{n}{2}}M$. The Lipschitz constants for the first and second derivatives are L and $\max\{M, 1\}$, respectively. The function given in (4.2) allows us to vary all the moving components in the gradient approximations, namely, the dimension n, the Lipschitz constants L and M of the gradients and Hessians, respectively, the sampling radius σ and the number of samples N, in order to evaluate the different gradient approximation methods.

For linear interpolation, the directions $\{u_i\}_{i=1}^n$ are chosen as $u_i \sim \mathcal{N}(0, I)$ for all $i = 1, 2, \ldots, n$, and then normalized so that they lie in a unit ball $u_i \leftarrow u_i / \max_{j\in\{1,\ldots,n\}} \|u_j\|$. We illustrate the performance of the gradient approximations using 5 box plots (Figure 1). The default values of the parameters are: $n = 20$, $M = 1$, $L = 2$, $\sigma = 0.01$, and $N = 4n$ (for the smoothing methods). For each box plot, we vary one of the parameters. Note, when comparing the relative errors for different values of M, the constant L is set to the same value as M. For all randomized methods, including linear interpolation, $\nabla f(0)$ is estimated 100 times, i.e., we compute 100 realizations of $g(0)$.

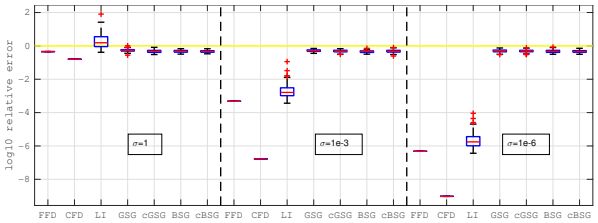
In accordance with our theory, we see in Figure 1a that the performance of most algorithms is not affected by the dimension n as long as the number of samples is chosen appropriately. The only algorithm that is affected is interpolation; this is because as the dimension increases the matrix Q formed by the sampling directions (chosen randomly) may get more ill-conditioned. In Figure 1b, we observe that the size of σ has a significant effect on the deterministic methods (FFD and CFD) and LI, while it has no effect on GSG, cGSG, BSG or cBSG because, as predicted by our theory one of the error terms in the approximation does not diminish with σ. In Figure 1c, we see that having more samples improves the accuracy achieved by GSG, cGSG, BSG or cBSG. Finally, in Figures 1d and 1e, we see how the FFD and other methods get adversely affected by large L, while CFD remain immune to these changes, and that the effect is reversed with respect to changes in M.

Gradient estimation on Schittkowski functions Next, we test the different gradient approximations on the 69 functions from the Schittkowski test set [27]. The methods we compare are the same as in the case of the synthetic function. We computed the gradient approximations for a variety of points with diverse values for $\nabla f(x_k)$ and local Lipschitz constants L. For each problem we generated points by running gradient descent with a fixed step size parameter $1/L$ for either 100 iterations or until the norm of the true gradient reached a value of 10^{-9}. Since for several problems the algorithm terminated in less than 100 iterations, the actual number of points we obtained was 6509.

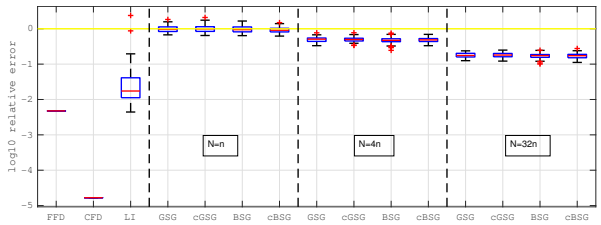
Table 4 summarizes the results of these experiments. We show the average of the log of the relative error (4.1) for 6509 points for different choices of σ ($\sigma \in \{10^{-2}, 10^{-5}, 10^{-8}\}$), and where appropriate different choices of N. The values in bold indicate values of $\theta < 1$. We observe that for the smoothing methods at least $2n$ samples are needed to reliably obtain $\log_{10}\theta < 0$ (or $\theta < 1$). Moreover, this experiment indicates that the relative errors θ for FFD, CFD and LI methods are significantly smaller than those obtained by the smoothing methods.



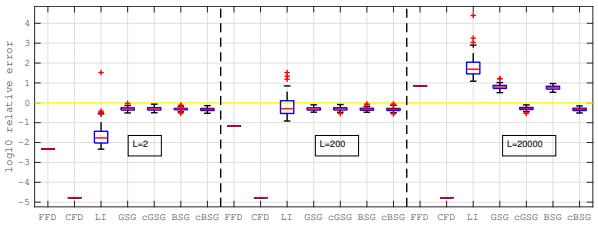
(a) Different n ($n \in \{2, 20, 200\}$).



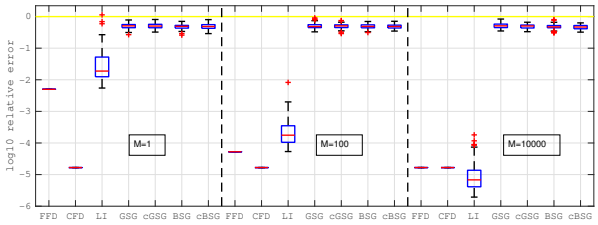
(b) Different σ ($\sigma \in \{10^0, 10^{-3}, 10^{-6}\}$).



(c) Different N ($N \in \{n, 4n, 32n\}$, smoothing methods).



(d) Different L ($L \in \{2, 200, 20000\}$).



(e) different M ($M \in \{1, 100, 10000\}$). Note, for these experiments $L = M$.

Figure 1: Log of relative error (4.1) of gradient approximations (FFD, CFD, LI, GSG, cGSG, BSG, cBSG) with different n, σ, N, L and M.

Table 4: Average (Log) Relative Error of Gradient Approximations for 6509 Problems.

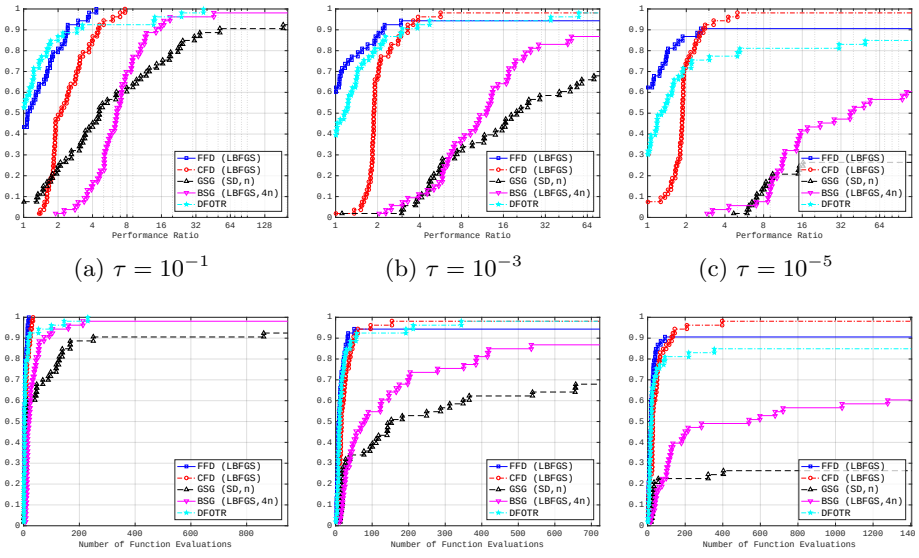
Approximation	**# of Samples (N)**	$\boldsymbol{\sigma} = 10^{-2}$	$\boldsymbol{\sigma} = 10^{-5}$	$\boldsymbol{\sigma} = 10^{-8}$
FFD	n	0.2720	**-2.6051**	**-5.3600**
CFD	$2n$	**-4.0794**	**-8.3847**	**-7.4542**
LI	n	0.7757	**-2.0992**	**-4.7714**
GSG	n	0.7449	0.0979	**-0.0254**
	$2n$	0.6527	**-0.0220**	**-0.1529**
	$4n$	0.5309	**-0.1580**	**-0.2988**
	$8n$	0.4020	**-0.3023**	**-0.4486**
cGSG	$2n$	0.1219	**-0.0453**	**-0.0403**
	$4n$	0.0022	**-0.1761**	**-0.1796**
	$8n$	**-0.1159**	**-0.3209**	**-0.3136**
	$16n$	**-0.2387**	**-0.4649**	**-0.4598**
BSG	n	0.6785	0.0193	**-0.1220**
	$2n$	0.5363	**-0.1191**	**-0.2542**
	$4n$	0.3856	**-0.2683**	**-0.4057**
	$8n$	0.2406	**-0.4242**	**-0.5563**
cBSG	$2n$	0.0375	**-0.1388**	**-0.1301**
	$4n$	**-0.0984**	**-0.2750**	**-0.2696**
	$8n$	**-0.2258**	**-0.4241**	**-0.4212**
	$16n$	**-0.3610**	**-0.5750**	**-0.5749**

4.2 Performance of Line Search DFO Algorithm with Different Gradient Approximations

The ability to approximate the gradient sufficiently accurately is a crucial ingredient of model based, and in particular line search, DFO algorithms. The numerical results presented in Section 4.1 illustrated the merits and limitations of the different gradient approximations. In this section, we investigate how these methods perform in conjunction with a line search DFO algorithm (Algorithm 1).

Several algorithms could be considered in this section. We focus on line search DFO algorithms that either compute steepest descent search directions ($d_k = -g(x_k)$) or L-BFGS search directions ($d_k = -H_k g(x_k)$). Moreover, we considered both adaptive line search variants as well as variants that used a constant, tuned step size parameter. Overall, we investigated the performance of 17 different algorithms. We considered algorithms that approximate the gradient using FFD, CFD and the four smoothing methods with steepest descent or L-BFGS search directions and an adaptive line search strategy. We also considered methods that approximate the gradient using the smoothing methods with steepest descent search directions and a constant step size parameter. Finally, as a benchmark, we compared the performance of the aforementioned methods against the popular DFOTR algorithm [1].

We tested the algorithms on the problems described in [18] (53 problems), and illustrate the performance of the methods using performance and data profiles [12, 18]. We first compare the performance of the *best variant* of each gradient approximation for different accuracy levels (τ) (Figure 2). We selected only the best performers among different possible variants by first comparing the variants among themselves. For example, for FFD and CFD the LBFGS variant outperformed the steepest descent variant. With regards to the smoothing methods, GSG with $N = n$ samples per iteration and steepest descent search directions was the best performer out of all GSG methods, and BSG with $N = 4n$ and LBFGS performed best among all BSG variants. For all the types of gradient approximations, the variants that performed the best used an adaptive step length procedure. We omit illustrations of these comparison for brevity.

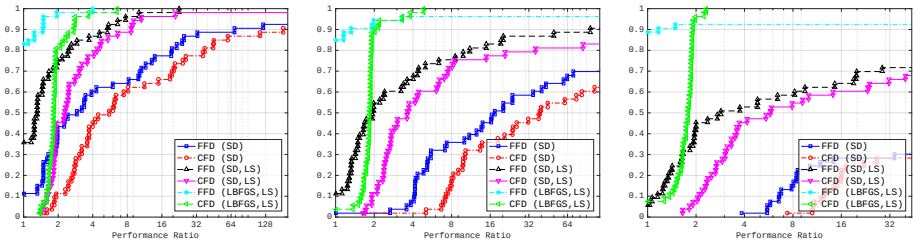


(a) $\tau = 10^{-1}$ (b) $\tau = 10^{-3}$ (c) $\tau = 10^{-5}$

(d) $\tau = 10^{-1}$ (e) $\tau = 10^{-3}$ (f) $\tau = 10^{-5}$

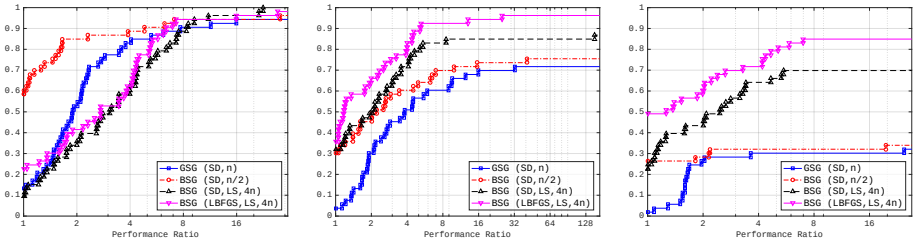
Figure 2: Performance and data profiles for best variant of each method. Top row: performance profiles; Bottom row: data profiles.

Next, we compare the adaptive step size methods against the constant step size ones (Figures 3 and 4).



(a) $\tau = 10^{-1}$ (b) $\tau = 10^{-3}$ (c) $\tau = 10^{-5}$

Figure 3: Performance profiles for Finite Difference variants with steepest descent (SD) and LBFGS search directions, and with and without a line search (LS).



(a) $\tau = 10^{-1}$ (b) $\tau = 10^{-3}$ (c) $\tau = 10^{-5}$

Figure 4: Performance profiles for best smoothed variants with steepest descent (SD) and LBFGS search directions, and with and without a line search (LS).

5 Final Remarks

We have shown that several derivative-free techniques for approximating gradients provide comparable estimates under reasonable assumptions. These approximations can be used effectively in conjunction with a line search algorithm, possibly with LBFGS search directions, provided they are sufficiently accurate. The convergence rates of the resulting methods match those of methods based on exact gradients. In this paper, all of the analysis is derived for the case of smooth functions. Our theoretical results, and related numerical experiments, show that finite difference and interpolation methods are much more efficient than smoothing methods in providing good gradient approximations. However, the situation may be different in the case of noisy, nonsmooth and stochastic functions which is it the subject for follow-up research.

References

[1] Afonso Bandeira, Katya Scheinberg, and Luis N Vicente. Computation of sparse low degree interpolating polynomials and their application to derivative-free optimization. *Mathematical Programming, Series B*, 134:223–257, 2012.

[2] Albert S Berahas, Richard H Byrd, and Jorge Nocedal. Derivative-free optimization of noisy functions via quasi-newton methods. *SIAM Journal on Optimization*, 29(2):965–993, 2019.

[3] Richard H Byrd, Gillian M Chin, Jorge Nocedal, and Yuchen Wu. Sample size selection in optimization methods for machine learning. *Mathematical programming*, 134(1):127–155, 2012.

[4] Richard G Carter. On the global convergence of trust region algorithms using inexact gradient information. *SIAM Journal on Numerical Analysis*, 28(1):251–265, 1991.

[5] Coralia Cartis and Katya Scheinberg. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Mathematical Programming*, pages 1–39, 2018.

[6] Krzysztof Choromanski, Atil Iscen, Vikas Sindhwani, Jie Tan, and Erwin Coumans. Optimizing simulations with noise-tolerant structured exploration. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2970–2977. IEEE, 2018.

[7] Krzysztof Choromanski, Mark Rowland, Vikas Sindhwani, Richard E Turner, and Adrian Weller. Structured evolution and compact parameterization for scalable policy optimization. *submitted for publication*, 2018.

[8] Andrew R Conn, Katya Scheinberg, and Philippe L Toint. On the convergence of derivative-free methods for unconstrained optimization. In A. Iserles and M. Buhmann, editors, *Approximation Theory and Optimization: Tributes to M. J. D. Powell*, pages 83–108, Cambridge, England, 1997. Cambridge University Press.

[9] Andrew R Conn, Katya Scheinberg, and Philippe L Toint. A derivative free optimization algorithm in practice. Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, Missouri, September 2-4, 1998.

[10] Andrew R Conn, Katya Scheinberg, and Luis N Vicente. Geometry of interpolation sets in derivative free optimization. *Mathematical programming*, 111(1-2):141–172, 2008.

[11] Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to Derivative-free Optimization*. MPS-SIAM Optimization series. SIAM, Philadelphia, USA, 2008.

[12] Elizabeth D Dolan and Jorge J Moré. Benchmarking Optimization Software with Performance Profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[13] Maryam Fazel, Rong Ge, Sham M Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. *arXiv preprint arXiv:1801.05039*, 2018.

[14] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394. Society for Industrial and Applied Mathematics, 2005.

[15] C T Kelley. *Implicit filtering*, volume 23. SIAM, 2011.

[16] Jeffrey Larson, Matt Menickelly, and Stefan Wild. Derivative-free optimization methods. *Submitted*, 2019.

[17] Alvaro Maggiar, Andreas Wächter, Irina S Dolinskaya, and Jeremy Staum. A derivative-free trust-region algorithm for the optimization of functions smoothed via gaussian convolution using adaptive multiple importance sampling. *SIAM Journal on Optimization*, 28(2):1478–1507, 2018.

[18] Jorge J Moré and Stefan M Wild. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20(1):172–191, 2009.

[19] Jorge J Moré and Stefan M Wild. Estimating derivatives of noisy simulations. *ACM Transactions on Mathematical Software (TOMS)*, 38(3):19, 2012.

[20] Yurii Nesterov. *Introductory Lectures on Convex Optimization*. Kluwer Academic Publsihers, Boston, MA, 2004.

[21] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

[22] Jorge Nocedal and Stephen J Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, NY, USA, 2nd edition, 2006.

[23] Courtney Paquette and Katya Scheinberg. A stochastic line search method with convergence rate analysis. *arXiv preprint arXiv:1807.07994*, 2018.

[24] Michael J D Powell. Unconstrained minimization algorithms without computation of derivatives. *Bollettino delle Unione Matematica Italiana*, 9:60–69, 1974.

[25] Michael J D Powell. The NEWUOA software for unconstrained optimization without derivatives. In *Large-Scale Nonlinear Optimization*, volume 83, pages 255–297. Springer, US, 2006.

[26] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. Technical Report arXiv:1703.03864, 2016.

[27] Klaus Schittkowski. *More test examples for nonlinear programming codes*, volume 282. Springer Science & Business Media, 2012.

[28] Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I Jordan. Stochastic cubic regularization for fast nonconvex optimization. *arXiv preprint arXiv:1711.02838*, 2017.

[29] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

[30] Stefan M Wild, Rommel G Regis, and Christine A Shoemaker. ORBIT: optimization by radial basis function interpolation in trust-regions. *SIAM Journal on Scientific Computing*, 30(6):3197–3219, 2008.

A Derivations

A.1 Derivation of (3.7)

$$\begin{aligned}\|\nabla F(x) - \nabla f(x))\| &= \|\mathbb{E}_{u\sim\mathcal{N}(0,I)}[\nabla f(x+\sigma u)] - \nabla f(x)\| \\ &\le \mathbb{E}_{u\sim\mathcal{N}(0,I)}[\|\nabla f(x+\sigma u) - \nabla f(x)\|] \\ &\le L\sigma\mathbb{E}_{u\sim\mathcal{N}(0,I)}[\|u\|] \\ &= L\sigma\sqrt{2}\frac{\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n}{2})} \\ &\le \sqrt{n}L\sigma.\end{aligned}$$

A.2 Derivation of (3.8)

$$\begin{aligned}\|\nabla F(x) - \nabla f(x))\| &= \left\|\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[\frac{1}{2}\nabla f(x+\sigma u) + \frac{1}{2}\nabla f(x-\sigma u) - \nabla f(x)\right]\right\| \\ &\le \frac{1}{2}\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[\|\left(\nabla f(x+\sigma u) - \nabla f(x)\right) - \left(\nabla f(x) - f(x-\sigma u)\right)\|\right] \\ &= \frac{1}{2}\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[\|\left(\nabla^2 f(x+\xi_1 u) - \nabla^2 f(x-\xi_2 u)\right)\sigma u\|\right],\end{aligned}$$

for some $0 \le \xi_1 \le \sigma$ and $0 \le \xi_2 \le \sigma$ by the intermediate value theorem. Then

$$\begin{aligned}\|\nabla F(x) - \nabla f(x))\| &\le \frac{1}{2}\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[\|\nabla^2 f(x+\xi_1 u) - \nabla^2 f(x-\xi_2 u)\|\|\sigma u\|\right] \\ &\le \frac{1}{2}\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[M\|\xi_1 u + \xi_2 u\| \cdot \sigma\|u\|\right] \\ &= \frac{1}{2}\mathbb{E}_{u\sim\mathcal{N}(0,I)}\left[|\xi_1 + \xi_2| \cdot \|u\|^2 M\sigma\right] \\ &\le nM\sigma^2.\end{aligned}$$

A.3 Derivation of (3.23)

$$\begin{aligned}\|\nabla F(x) - \nabla f(x))\| &= \|\mathbb{E}_{u\sim\mathcal{U}(\mathcal{B}(0,1))}[\nabla f(x+\sigma u)] - \nabla f(x)\| \\ &\le \mathbb{E}_{u\sim\mathcal{U}(\mathcal{B}(0,1))}[\|\nabla f(x+\sigma u) - \nabla f(x)\|] \\ &\le L\sigma\mathbb{E}_{u\sim\mathcal{U}(\mathcal{B}(0,1))}[\|u\|] \\ &\le L\sigma.\end{aligned}$$

A.4 Derivation of (3.24)

$$\begin{aligned}\|\nabla F(x) - \nabla f(x))\| &= \left\|\mathbb{E}_{u\sim\mathcal{U}(\mathcal{B}(0,1))}\left[\frac{1}{2}\nabla f(x+\sigma u) + \frac{1}{2}\nabla f(x-\sigma u) - \nabla f(x)\right]\right\| \\ &\le \frac{1}{2}\mathbb{E}_{u\sim\mathcal{U}(\mathcal{B}(0,1))}\left[\|\left(\nabla f(x+\sigma u) - \nabla f(x)\right) - \left(\nabla f(x) - f(x-\sigma u)\right)\|\right] \\ &= \frac{1}{2}\mathbb{E}_{u\sim\mathcal{U}(\mathcal{B}(0,1))}\left[\|\left(\nabla^2 f(x+\xi_1 u) - \nabla^2 f(x-\xi_2 u)\right)\sigma u\|\right],\end{aligned}$$

for some $0 \le \xi_1 \le \sigma$ and $0 \le \xi_2 \le \sigma$ by the intermediate value theorem. Then

$$\begin{aligned}
\|\nabla F(x) - \nabla f(x))\| &\le \frac{1}{2}\mathbb{E}_{u\sim\mathcal{U}(\mathcal{B}(0,1))}\left[\|\nabla^2 f(x+\xi_1 u) - \nabla^2 f(x-\xi_2 u)\|\|\sigma u\|\right] \\
&\le \frac{1}{2}\mathbb{E}_{u\sim\mathcal{U}(\mathcal{B}(0,1))}\left[M\|\xi_1 u + \xi_2 u\| \cdot \sigma\|u\|\right] \\
&= \frac{1}{2}\mathbb{E}_{u\sim\mathcal{U}(\mathcal{B}(0,1))}\left[|\xi_1 + \xi_2| \cdot \|u\|^2 M\sigma\right] \\
&\le M\sigma^2.
\end{aligned}$$