Small Errors in Random Zeroth Order Optimization are Imaginary

Wouter Jongeneel[a] Man-Chung Yue[b] Daniel Kuhn[a]

[a]Risk Analytics and Optimization Chair, École Polytechnique Fédérale de Lausanne,
{wouter.jongeneel,daniel.kuhn}@epfl.ch
[b]Department of Applied Mathematics, The Hong Kong Polytechnic University,
manchung.yue@polyu.edu.hk

March 10, 2021

Abstract

The vast majority of zeroth order optimization methods try to imitate first order methods via some smooth approximation of the gradient. Here, the smaller the smoothing parameter, the smaller the gradient approximation error. We show that for the majority of zeroth order methods this smoothing parameter can however not be chosen arbitrarily small as numerical cancellation errors will dominate. As such, theoretical and numerical performance could differ significantly. Using classical tools from numerical differentiation we will propose a new smoothed approximation of the gradient that can be integrated into general zeroth order algorithmic frameworks. Since the proposed smoothed approximation does not suffer from cancellation error, the smoothing parameter (and hence the approximation error) can be made arbitrarily small. Sublinear convergence rates for algorithms based on our smoothed approximation are proved. Numerical experiments are also presented to demonstrate the superiority of algorithms based on the proposed approximation.

1 Introduction

Let $f : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}$ be some objective function, $\mathcal{D}$ an open set and $\mathcal{X} \subseteq \mathcal{D}$ a closed non-empty set, then, we are interested in solving problems of the form

$$\underset{x \in \mathcal{X}}{\text{minimize}} \quad f(x). \tag{1.1}$$

Assuming some optimizer in (1.1) exists, we denote any by $x^\star$. In solving (1.1) we will not have a standard arsenal of tools at our disposal, instead, we assume to have access to function evaluations only, that is, a finite sequence of the form $f(x_0), f(x_1), \ldots, f(x_K)$. In other words, the objective function is unknown to us, but sampling its actions is possible. This approach falls under the umbrella of *zeroth order* optimization, or equivalently *derivative-free* optimization.

In motivating this setting we briefly highlight some remarks taken from the monograph [CSV09] and the paper [Nes11]. Zeroth order optimization is a framework intended to deal with problems (1.1) for which first order methods do not (easily) apply. For example, obtaining a reasonable approximation of the gradient $\nabla f(x)$ might be costly or simply impossible. The latter scenario is true for many simulation-based problems where a closed-form expression of f is unavailable, *e.g.,* for many modern engineering problems x relates to a design parameter and f to an online physics simulator only capable of providing the evaluation $f(x)$. An equally complicated setting

could be that of f being itself of a variational form, *e.g.,* in minimax optimization problems f is of the form $f(x) = \sup_{z\in\mathcal{Z}} h(x,z)$. Explicit gradients are usually not available, or worse, the inner maximization can only be solved numerically. Another setting frequently encountered in practice is that of finding the optimal values of tuning parameters in numerical algorithms. The dependence of the solution on this parameter is almost always implicit which hinders classical gradient descent. However, as function evaluations are possible, zeroth order methods *are* possible. Furthermore, as only function evaluations are used, this approach is rather crude and it is believed that this property makes it suitable for non-convex problems, *e.g.,* to escape saddle-points.

There are essentially two lines of attack when it comes to zeroth order optimization, one could use the available data to construct an explicit C^r local model of f for $r \geq 1$, say $\tilde{f}$ and apply appropriate r^{th} order optimization tools to locally improve the current iterate. Clearly, if $\tilde{f}$ is of high quality, then this is a rather powerful method. However, to construct such a $\tilde{f}$ a lot of data is usually needed.

To address this, a different and increasingly popular approach — which is also the topic of this paper — is to approximate the gradient of f directly from the data. To do so, the vast majority of these zeroth order optimization methods introduce a δ-*smoothed* version of f, denoted f_δ, for f_δ being δ-close to f. The idea is to construct an unbiased estimator $g_\delta(x)$ for $\nabla f_\delta(x)$ via sampling the objective f at selected data $\{x_k\}_{k=0}^K$ around x. As one can usually show that $\|\nabla f_\delta(x) - \nabla f(x)\| \leq O(\delta^p)$ for some $p \geq 1$, the estimator $g_\delta(x)$ can be used as a surrogate gradient in for example a gradient-descent algorithm. Compared to model-based methods, the key benefit is that less data and less assumptions are needed to find an approximation of $x^\star$ in (1.1). Nevertheless, this usually comes at the cost of weaker guarantees. Clearly, one likes to select the smallest possible smoothing parameter δ as this increases approximation quality and thereby presumably performance of the algorithm. Unfortunately, the numerically useful range of δ — and thereby of the full scheme — is usually unknown or at least not discussed.

In this work we will exploit insights from numerical differentiation to propose a new smoothed approximation f_δ such that δ, and thereby the error in the gradient approximation scheme, can be made arbitrarily small, which is not possible using conventional schemes. Algorithms based on our new smoothed approximation enjoy provable convergence rates, are more numerically stable, and most importantly, outperform those algorithms based on existing smoothed approximation in terms of both accuracy and computational speed as shown in our experiments.

Notation We denote the real and imaginary parts of a complex number $z = a + ib$ by $\Re(z) = a$ and $\Im(z) = b$, respectively. Let $\mathbb{B}^n := \{x \in \mathbb{R}^n : \|x\|_2 \leq 1\}$ and $\mathbb{S}^{n-1} := \{x \in \mathbb{R}^n : \|x\|_2 = 1\}$ be the unit Euclidean ball and sphere, respectively, in $\mathbb{R}^n$. Throughout the paper, if distributions are declared to be uniform over $\mathcal{Y}$, we simply write $y \sim \mathcal{Y}$ and additionally

$$\mathbb{E}_{y\sim\mathcal{Y}}[g(y)] = \frac{1}{\text{vol}(\mathcal{Y})}\int_{\mathcal{Y}} g(y)dy,$$

with dy denoting the Lebesgue measure of $\mathcal{Y}$. If $f : \mathcal{D} \to \mathbb{R}$ is r times continuously differentiable, we denote this by $f \in C^r(\mathcal{D})$. Given two real-valued funtions h_1 and h_2, then if there exists a function h_3 such that $h_1(x) \leq h_2(x) + h_3(x)$ with $\lim_{x\to\infty} h_3(x)/h_1(x) = 0$ we denote this asymptotic inequality by $h_1 \lesssim h_2$. At last, $C_1, C_2, \ldots, C_y, C_z$ always denote non-negative universal constants which can change from line to line.

1.1 Related work Zeroth order optimization based on a single-point oracle was arguably for the first time described in [NY83, Section 9.3]. There, the authors approximate a function $f : \mathbb{R}^n \to \mathbb{R}$ by the smoothed objective

$$f_\delta(x) = \frac{1}{\text{vol}(\mathbb{B}^n)}\int_{\mathbb{B}^n} f(x + \delta v)dv, \tag{1.2}$$

where the amount of smoothing is controlled by the smoothing parameter $\delta > 0$. Then, the key observation[1] is that $\nabla f_\delta(x)$ admits an integral representation of f as well

$$\nabla f_\delta(x) = \frac{n}{\delta} \frac{1}{\mathrm{vol}(\mathbb{S}^{n-1})} \int_{\mathbb{S}^{n-1}} f(x+\delta u) u du. \tag{1.3}$$

Hence, a natural *one-point* candidate to approximate the gradient of f would be

$$g_\delta(x) = \frac{n}{\delta} f(x+\delta u) u, \quad u \sim \mathbb{S}^{n-1} \tag{1.4}$$

as $\mathbb{E}_{u\sim\mathbb{S}^{n-1}}[g_\delta(x)] = \nabla f_\delta(x)$. The authors in [FKM04] applied this approach in the bandit context. However, already in [NY83] it was remarked that the variance of (1.4) is problematic for $\delta \downarrow 0$, which is inconvenient as a smaller δ in (1.2) leads to a smaller bias. To overcome this deficiency, the key observation from [ADX10; Nes11] is then to give (1.4) again the interpretation of a directional derivative by stressing that one should consider a *multi-point* oracle of the form

$$g'_\delta(x) = \frac{n}{\delta} \left(f(x+\delta u) - f(x)\right) u, \quad u \sim \mathbb{S}^{n-1}. \tag{1.5}$$

Although the expected values of $g_\delta(x)$ and $g'_\delta(x)$ are equivalent, their variance is different since (1.5) is bounded in δ while (1.4) is not. One problem pertains, δ needs to be made arbitrarily small to solve the minimization problem up to arbitrary precision, yet for sufficiently small δ, $f(x+\delta u)$ and $f(x)$ will be numerically indistinguishable and hence oracles of the form (1.5) are still prone to numerical complications and cannot always provide arbitrarily high precision indeed. When function evaluations are noisy this is even more true [Lia+16]. Also, see [HL14] for further generalizations of the oracle (1.4) and [Duc+15; LLZ21] for more on the optimality of multi-point schemes.

We assume that f is fixed and that the function measurements are deterministic. Using deterministic function evaluations one could try to directly approximate $\nabla f(x)$, *e.g.,* by a coordinate-wise finite-difference method. This line of work relates to inexact gradient methods [d'A08; DGN14]. However, as in [Nes11; NS17], we will use a randomized (analysis) method. One of the benefits is that we have a clear framework to understand convergence of our algorithm, namely, we will consider convergence in expectation $f(x_k) \to f(x^\star)$ whereas in fully deterministic inexact methods a bias inevitably prevails.

Although we use a randomized *method*, as in [NS17] we assume to have access to a deterministic oracle. This is in contrast with a variety of other works which assume to have access only to noisy function evaluations of the form $f(x)+\xi$, $\xi \overset{i.i.d.}{\sim} \Xi$. We note however that most methods assume the noise to be well-behaved in the sense that its distribution is light-tailed and hence the uncertainties can be well handled by averaging. If the function evaluation contains adversarial noise, negative results are obtained in [SV15].

Although our function measurements are without noise, we will randomly sample function evaluations and hence we are interested in construction a sequence of iterates $x_0, x_1, \ldots, x_{K-1}$ which minimize the following regret

$$\frac{1}{K} \sum_{k=0}^{K-1} \mathbb{E}\left[f(x_k) - f(x^\star)\right]. \tag{1.6}$$

Then, if the regret (1.6) is bounded by some $\epsilon \geq 0$ and f is convex, the *averaged* estimator $\bar{x}_{K-1} := (1/K)\sum_{k=0}^{K-1} x_k$ achieves the expected suboptimality gap (optimization error)

$$\mathbb{E}\left[f(\bar{x}_{K-1}) - f(x^\star)\right] \leq \frac{1}{K} \sum_{k=0}^{K-1} \mathbb{E}\left[f(x_k) - f(x^\star)\right] \leq \epsilon. \tag{1.7}$$

[1] See the proof of Lemma 3.4 for more on how to pass from (1.2) to (1.3).

We are interested in this suboptimality gap ϵ as a function of the samplesize K and a variety of other problem parameters such as n, the dimension of x, and the Lipschitz constants of the derivatives of f. If the problem setting allows for it, we analyze the gap $\mathbb{E}[f(x_k) - f(x^\star)]$ as well, *e.g.,* when f is strongly convex. If the samplesize K appears in the suboptimality gap one usually talks about the convergence *rate*, *e.g.,* $\epsilon = O(1/K)$.

In [Nes11; NS17] the smoothing is carried out via passing to a Gaussian kernel in combination with a multi-point oracle of the form (1.5). The authors conclude with stating that random methods are typically $O(n)$ times slower than their deterministic counterparts. For example, if f is convex and has a Lipschitz-continuous gradient, then the rate provided by [NS17] is $O(n/K)$. In [Duc+15; Sha17] these results are improved, conditioned on having access to multiple function evaluations. See also [Gas+17]. Ghadimi and Lan [GL13] consider extending the nonconvex programming part of [Nes11] and Balasubramanian and Ghadimi [BG18] look at the method proposed by [Nes11] via the Gaussian Stein identity and show that approximating second order information is also possible.

Simultaneously, in [Sha13], the author provides lower bounds and shows that in the derivative-free case, most errors cannot vanish faster than $O(n/\sqrt{K})$. However, under certain conditions like strong-convexity, a $O(n/K)$ rate is possible. Recall however that in [BM13] it is shown that by exploiting problem structure $O(1/K)$ rates are possible for some non-strongly convex problems.

Although our focus is not on the stochastic oracle setting we highlight another line of closely related work. Starting with [PT90], a variety of authors looked into exploiting higher-order smoothness. For example, in [BP16] the ideas from [PT90] are extended to the online case and in [APT20; NG21] the convergence rates are subsequently improved under increasingly weak noise assumptions. It is imperative to remark that in the stochastic setting the selection of the smoothing parameter δ has a twofold motivation. On the one hand the estimation error should be small, but on the other hand a small variance is preferred and usually this variance has a term proportional to $1/\delta$. An increasingly popular method to further reduce the variance is to make use of mini-batch methods, *e.g.,* see [Ji+19].

Also, recent work considers extending zeroth order optimization ideas to the Riemannian setting [LBM20] and even a more limited supply of information can be considered [Cai+21].

One of the appealing aspects of zeroth order optimization is its simplicity. As such we present simple algorithms with pre-determined stepsizes, but note that line-search (direct search) and model-based methods provide for an interesting alternative outlook [CSV09; SMG13; BCS19], *e.g.,* to exploit underlying low-dimensional structures [Gol+19].

For more information see the recent surveys on zeroth order (derivative-free) optimization [LMW19; Liu+20].

1.2 Numerical considerations The majority of the zeroth order optimization literature is approximating a variety of derivatives and indeed gradients but unfortunately usually without taking numerical considerations into account. For example, given some $f \in C^r(\mathbb{R})$ with $r \geq 1$, then in the approximation

$$\partial_x f(x) \approx \frac{f(x+\delta) - f(x)}{\delta} \tag{1.8}$$

one cannot make $\delta > 0$ arbitrarily small and expect to recover $\partial_x f(x)$. Theoretically this is correct, but on a machine, $f(x+\delta)$ and $f(x)$ coincide for sufficiently small $\delta > 0$. For example, let $f : \mathbb{R} \to \mathbb{R}$ be locally Lipschitz around x, then there is some constant L such that $|f(x+\delta) - f(x)| \leq L \cdot |\delta|$ for sufficiently small δ. Now we see that if $L \cdot |\delta|$ is smaller than machine-precision, numerical cancellation errors are prone to dominate, see [Ove01, Chapter 11]. Running into these machine-precision problems is inherent to finite-difference (multi-point) methods like (1.8), yet, the vast majority of zeroth order oracles is of this form whilst demanding δ to be arbitrarily small, *e.g.,* in [APT20, Theorem 3.1] the authors require $\delta_k = O(1/\sqrt{k})$ on a multi-point scheme. In a small-noise regime this will be eventually problematic.

The pioneering and celebrated construction as proposed by [NY83] alleviates — to the best of our knowledge unintentionally — part of this numerical issue by approximating the gradient of f using an oracle of the form $\frac{n}{\delta} f(x+\delta u)u$, $u \sim \mathbb{S}^{n-1}$. Since there is just one function call (single-point), the oracle has the potential to be numerically more stable. However, as explained before, the variance is ill-behaved. Therefore, we are motivated to propose a numerically stable oracle, yet, allowing for convergence rates comparable to the modern literature. We will propose a special one-point oracle which inherits the numerical properties of one-point oracles, *i.e.,* no cancellation errors, at the same time, this oracle will display the approximation quality and convergence speed of two-point oracles. All of this is achieved without introducing significant computational overhead.

Throughout we make two assumptions which deserve to be emphasized.

Assumption 1.1 (Smoothness)**.** *The objective function f is real-analytic over $\mathcal{D} \subseteq \mathbb{R}^n$, denoted $f \in C^\omega(\mathcal{D})$.*

Although somewhat non-standard in the optimization literature, looking for example at zeroth order optimization in the context of reinforcement learning [Faz+18; Mal+19], we like to remark that for these particular policy iteration problem the objective is real-analytic in the feedback gain [Pol86]. Also note that $f \in C^\omega(\mathbb{R}^n)$ does not immediately imply β^{th} order smoothness as used in for example [BP16], *e.g.*, 0^{th} order smoothness translates to f being bounded and 1^{st} order smoothness translates to f being (globally) Lipschitz.

Assumption 1.1 is mainly there to allow for the next assumption.

Assumption 1.2 (Complex oracle)**.** *Consider some unknown function $f \in C^\omega(\mathcal{D})$, we assume to have access to an oracle which can output $f(x+iy) \in \mathbb{C}$, for any $x+iy =: z \in \Omega \subseteq \mathbb{C}^n$ with $\mathcal{D} \subset \Omega$.*

Hence, the oracle must be able to process complex computations, plus, for simplicity — and to some extent realism — we assume it to be deterministic. The set Ω will be specified later on.

Contribution It was pointed out in [Duc+15; LLZ21] that, when looking at theoretical convergence rates, multi-point schemes are preferred in zeroth order optimization over algorithms using single function evaluations. However, from a numerical point of view, these single-point schemes are inherently more robust as in contrast to multi-point schemes cancellation errors can be avoided. In this work we combine the fast convergence rate and low variance of multi-point schemes with the numerical robustness of the single-point approach. All of this is made possible via passing to the complex domain.

Let $R := \|x_0 - x^\star\|_2$, $F := f(x_0) - f(x^\star)$, then, under the assumption that f is real-analytic and has a L-Lipschitz continuous gradient, we show for the convex case (unconstrained and constrained) that the optimization error scales as $O(nLR^2/K)$. For the τ-strongly convex case, the optimization error decays at a global linear rate of the form $O((1-\tau/(4nL))^K LR^2)$, whereas for the non-convex case a local $O(nLF/K)$ rate is shown. All these results are similar to the rates as in [NS17]. In fact, these rates are sharper than the noise-free rates as provided in [APT20], where higher-order smoothness — although for the perturbed setting — is explicitly taken into account. The key difference with all of the aforementioned works is that we can select our smoothing parameter as being for example of the form $\delta_k = \delta/(1+k)$ *without* the fear of numerical problems.

What is more, as highlighted throughout the recent survey article by Larson, Menickelly, and Wild [LMW19], it is not clear if there is a single-point method which is as fast as a multi-point method. This observation motivates Zhang et al. [Zha+20] to use some form of memory such that their oracle only demands a single new point each call. Nevertheless, their oracle results in a multi-point method. Although their algorithm is faster than [FKM04], it is slower than [NS17]. On the contrary, we will provide the first real single-point method which is as fast, but usually faster, than the multi-point methods by Nesterov and Spokoiny [NS17]. Numerical experiments will corroborate this claim.

Structure In Section 2 we highlight the complex-step method from numerical differentiation along with indispensable tools from Complex analysis. Then, in Section 3 we introduce our complex-step oracle and in Sections 4-6 we analyze the performance of this oracle in the context of convex, strongly convex and non-convex optimization algorithms, respectively. Then, in Section 7 we provide a variety of numerical experiments to substantiate the claims. At last, a few auxiliary results are relegated to the Appendix.

2 Preliminaries

As we assume our oracle to handle complex computations we first highlight what this means for our objective function f. Having recalled these results we can immediately clarify the power of such an oracle.

2.1 Notions from Complex analysis Given a sufficiently smooth function $f : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}$, the most important results from this section clarify what can be said about the complex extension of f. That is, what kind of structure is preserved if we evaluate f over some subset of $\mathbb{C}^n$ instead of merely over $\mathcal{D}$. To be clear, when we consider $f : \mathbb{C}^n \to \mathbb{C}$, $f : z \mapsto f(z)$, one usually identifies $\mathbb{C}^n$ with $\mathbb{R}^{2n}$ and writes $f : (x, y) \mapsto f(x + iy)$ with $f(x, y) = u(x, y) + iv(x, y)$ for some functions u and v. In what follows we mainly rely on [Kra00; KP02]

A function $f : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}$ is (real)-*analytic* on $U \subseteq \mathcal{D}$, denoted $f \in C^\omega(U)$, if for any $x' \in U$ there exists a neighbourhood $U' \ni x'$ such that for all $x \in U'$ the map f can be locally represented by the convergent power series $f(x) = \sum_{n=0}^{\infty} a_n (x - x')^n$. This means, for example, that we have a convergent Taylor series.

Let $\alpha \in \mathbb{N}^n$ be a multi-index, that is $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, $|\alpha| = \sum_{i=1}^n \alpha_i$, $\alpha! = \prod_{i=1}^n \alpha_i!$ and most importantly $\partial_x^\alpha = \prod_{i=1}^n \partial_{x_i}^{\alpha_i}$. This allows for a clean description of multivariate Taylor's series.

Theorem 2.1 (Multivariate Taylor's series [KP02, Section 2.2])**.** *Let $f \in C^\omega(\mathcal{D})$, $\mathcal{D} \subseteq \mathbb{R}^n$, then for each $x_0 \in \mathcal{D}$ there exists a neighbourhood $U \subseteq \mathcal{D}$ containing x_0 such that for each $(x_0 + h) \in U$*

$$f(x_0 + h) = \sum_{|\alpha| \in \mathbb{N}} \frac{1}{\alpha!} (\partial_x)^\alpha f(x_0) h^\alpha. \tag{2.1}$$

To indicate the difference between the real and complex setting, we denote by $f \in C^\omega(\mathcal{D})$ the fact that the real-valued function f is *analytic* over the domain $\mathcal{D} \subseteq \mathbb{R}^n$, whereas $g \in H(\Omega)$ denotes that the complex-valued function g is *holomorphic* (complex differentiable) over $\Omega \subseteq \mathbb{C}^n$.

Although at our core we interested in problems over $\mathbb{R}^n$, in general, however, we will not work over $\mathbb{R}^n$, but rather over a subset of $\mathbb{C}^n$. To that end, the following Lemma asserts the possibility of extending a real-valued domain to a complex-valued domain.

Lemma 2.2 (From real-analytic to complex-analytic)**.** *If $f : \mathbb{R}^n \to \mathbb{R}$ is real-analytic, then there exists some open set $\Omega \subseteq \mathbb{C}^n$ such that $\mathbb{R}^n \subseteq \Omega$ and f is complex-analytic (or equivalently, holomorphic) on Ω.*

Proof. To be remotely self-contained we recall this basic proof. Since $f \in C^\omega(\mathbb{R}^n)$ for any $x^\circ \in \mathbb{R}^n$, there is a $r > 0$ and sequence $(a_n)_{n \in \mathbb{N}}$ such that the local power series representation of f, that is, $f(x) = \sum_{n=0}^{\infty} a_n (x - x^\circ)^n$ converges for all x in the open ball $B_r(x^\circ)$. Recall that the radius of convergence r equals the distance (in $\mathbb{C}^n$) from $z^\circ := (x^\circ, 0 \cdot i) \in \mathbb{C}^n$ to a singularity, that is, the most nearby point to z° where f fails to be holomorphic, *e.g.*, see [Kra00, Section 2.3]. Hence, the local complex extension follows naturally by considering the same power series representation via $(a_n)_{n \in \mathbb{N}}$ yet, for all $z \in \mathbb{C}^n$ in the open ball $B_r(z^\circ)$. The result follows after applying the same construction for any x°. □

Although perhaps expected, Ω cannot always be assumed to be of the form $\Omega \simeq \mathbb{R}^n \times (-\epsilon, \epsilon)^n$, $\epsilon > 0$[2]. The set Ω is however difficult to quantify. To circumvent this we will mainly assume that $f : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}$ is holomorphic over $\mathcal{D} \times (-\bar{\delta}, \bar{\delta})$ for some $\bar{\delta} > 0$. This holds for example when $f \in C^\omega(\mathbb{R}^n)$ and $\mathcal{D} \subset \mathbb{R}^n$ is bounded. One could circumvent this implicit description of Ω by simply assuming f to be real-entire[3], but this is a rather strong assumption, for instance, consider the well-known example function $f(x) = 1/(1+x^2)$, which fails to be holomorphic at $\pm i$.

The main technical tool is one of the most celebrated tools from Complex analysis

Theorem 2.3 (Multivariate Cauchy-Riemann equations)**.** *Let $f : \Omega \subseteq \mathbb{C}^n \to \mathbb{C}$ and define $z := x + iy$ together with the functions u and v such that the map f is defined by $x + iy \mapsto f(x + iy) = u(x, y) + iv(x, y)$. Then, if f is holomorphic over Ω, the Cauchy-Riemann equations*

$$\partial_{x_i} u = \partial_{y_i} v, \quad -\partial_{x_i} v = \partial_{y_i} u, \quad i = 1, 2, \ldots, n \tag{2.2}$$

are satisfied.

For proofs and more on Theorem 2.3 see [Kra00; KW17]. The converse to Theorem 2.3 is not immediate and usually requires further assumptions [GM78].

We end with an important connection between real-analyticity and strict-convexity.

Lemma 2.4 (Generalized principle of isolated zeroes)**.** *Let $f \in C^\omega(\mathbb{R}^n)$. If f is constant over some open subset $A \subseteq \mathbb{R}^n$ with $\mathrm{vol}(A) > 0$, then f is constant over the whole of $\mathbb{R}^n$.*

Proof. The proof is a simple consequence of a far reaching result by Łojasiewicz, *e.g.*, see [Theorem 6.3.3][KP02]. □

Corollary 2.5 (Real-analytic implies local strict convexity)**.** *For any non-constant convex function $f \in C^\omega(\mathbb{R}^n)$, there is a line segment $\ell \subset \mathbb{R}^n$ such that $f|_\ell$ is strictly convex.*

Corollary 2.5 is important to take into account when comparing algorithms, we do not allow for the most generic convex objective functions.

2.2 The imaginary trick

The most straight-forward method of numerical differentiation is the *finite-difference* method [Vui+07, Chapter 3] which is based on the approximation

$$\partial_x f(x) = \frac{f(x+h) - f(x)}{h} + O(h).$$

However, as explained before, for sufficiently small h this method suffers from numerical drawbacks (subtractive cancellation errors). To overcome this we can use an elegant idea which builds upon some ideas from Complex analysis. This approach was arguably introduced in [LM67] with the first concrete *complex-step* approach appearing in [ST98] and with later elaborations in [MSA03; ASM15; Abr+18].

Instead of f mapping $\mathbb{R} \to \mathbb{R}$, enlarge its domain to $\mathbb{C}$, that is, let with some abuse of notation $f(z) = u(z) + iv(z)$ with $z = x + iy \in \mathbb{C}$ such that $f(x) = f(z)|_{y=0, v(x)=0}$. Now if f is holomorphic, that is $f \in H(\Omega)$ for $\mathbb{R} \subset \Omega$, we can appeal to the Cauchy-Riemann equations, telling us that $\partial_x u = \partial_y v$ and hence

$$\partial_x u = \lim_{h \downarrow 0} \frac{v\big(x + i(y+h)\big) - v(x+iy)}{h}. \tag{2.3}$$

[2] For example, consider $f(x) = \sum_{k=1}^{\infty} a_k \frac{1}{1+kx^2}$ for some appropriate sequence $(a_k)_k$. The set of poles contains the set of points $(1/\sqrt{k})i$, $k \in \mathbb{N}_{>0}$, such that Ω can never contain a the full strip of the form $\mathbb{R}^n \times (-\epsilon, \epsilon)^n$, $\epsilon > 0$.

[3] The function f is real-entire when it has a globally convergent power series representation.

Then since $f(x) = f(z)|_{y=0, v(x)=0}$ and $\Im(f) = v$, we obtain from (2.3)

$$\partial_x f(x) = \lim_{h \downarrow 0} \frac{\Im\big(f(x+ih)\big)}{h}, \tag{2.4}$$

or non-asymptotically, $\partial_x f \approx \Im(f(x+ih))/h$. Hence, there are no finite-differences, merely a single function evaluation. Moreover, taking the imaginary part of the Taylor series of $f(x+ih)$ around x (which we can do due to the holomorphic assumption), we find that

$$f(x+ih) = f(x) + \partial_x f(x) ih - \frac{1}{2}\partial_x^2 f(x) h^2 - \frac{1}{6}\partial_x^3 f(x) ih^3 + O(h^4) \tag{2.5}$$

and thus

$$\partial_x f(x) = \frac{\Im\big(f(x+ih)\big)}{h} + O(h^2),$$

which is a remarkably powerful observation. Indeed, one can also see the imaginary trick itself directly from (2.5). Moreover, $f(x) = \Re(f(x+ih)) + O(h^2)$. So although we do not know f, by lifting the problem to the Complex domain, we can get good approximations of both f and its gradient using essentially a single function evaluation.

One can continue this line of thinking and introduce (multi-point) finite differences again and obtain even smaller errors of the order $O(h^4)$ [ASM15]. Of course, at the cost of also introducing numerical troubles, so unless time is expensive, the current complex-step remains favourable. Moreover, the theory has been extended to the matrix case in [AMH10] and even to higher-order derivatives [LRD12], plus see [MSA03] for more on the relation to Automatic Differentiation.

Due to the lack of cancellation errors, the precision of the complex-step approximation can be effectively arbitrarily high. Especially the aerospace community has adopted this method actively, the complex-step approximation has been for example used in air-foil design [GWX17], more impressively it is reported in [CH04, Page 44] that a value of $h = 10^{-100}$ is successfully used in National Physical Laboratory software. More recently and more related to our goals, in [NS18] the authors consider using the complex-step coordinate-wise under complex noise and propose its use in line-search methods. However, the authors do not provide a complete analysis. The high accuracy potential of the imaginary trick (2.2) is also mentioned in [SW18; BBN19], but only by looking at (2.2). We will use the complex-step method in place of the oracle (1.5) and provide the full analysis.

Especially in the context of optimization algorithms it is desirable to be able to assert with a selected amount of accuracy that the gradient of the objective function at the current iterate is sufficiently small, *i.e.,* to assert some form of local optimality. The next example shows that most standard numerical approaches to approximate the gradient, but the imaginary step, fail to posses this desirable property.

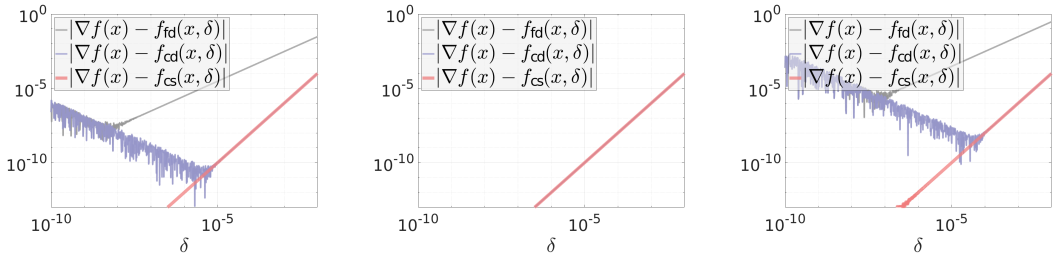
Example 2.6 (Numerical oracle quality)**.** To show the power of the complex-step and concurrently the numerical difficulties within finite-difference methods we approximate the derivative of $f(x) = x^3$ at $x \in \{-1, 0, 10\}$. We use a forward-difference (fd), central-difference (cd) and complex-step (cs) method

$$f_{\mathsf{fd}}(x, \delta) = \frac{f(x+\delta) - f(x)}{\delta}, \tag{2.6a}$$

$$f_{\mathsf{cd}}(x, \delta) = \frac{f(x+\delta) - f(x-\delta)}{2\delta}, \tag{2.6b}$$

$$f_{\mathsf{cs}}(x, \delta) = \frac{\Im\left(f(x+i\delta)\right)}{\delta} \tag{2.6c}$$

and compare the error for $\delta \downarrow 0$. In Figure 2.1 we show the three different errors as a function of δ. We see that f_{cd} and f_{cs} are of comparable quality, up to some point, indeed both have an $O(\delta^2)$



(a) $x = -1$ **(b)** $x = 0$ **(c)** $x = 10$

Figure 2.1: Comparison of the different oracles presented in (2.6) on the test setting of Example 2.6 around a variety of points x.

error. However, only the complex-step can reach machine-precision, whereas the other methods deteriorate (subtractive cancellation errors become dominant). At this point it is instrumental to mention that the majority of references use oracles of the form (2.6a)-(2.6b), hence it is clear that there is room for some numerical improvements and this indicates the potential of the imaginary trick in the context of zeroth order optimization.

2.3 Lipschitz inequalities In this part we gather a variety of inequalities which come in useful later. Note that none of them assumes the underlying function f to be convex.

Using the notation from [Nes03] a function f is said to be an element of $C_L^{k,p}(\mathcal{D})$ when f is k times continuously differentiable with additionally having its p^{th}-derivative being L-Lipschitz over $\mathcal{D} \subseteq \mathbb{R}^n$. That is, if $f \in C^{1,1}_{L_1(f)}(\mathbb{R}^n)$, then f has a Lipschitz gradient,

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L_1(f)\|x - y\|_2, \quad \forall x, y \in \mathbb{R}^n \tag{2.7}$$

and indeed, (2.7) implies that

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{1}{2L_1(f)}\|\nabla f(x) - \nabla f(y)\|_2^2, \quad \forall x, y \in \mathbb{R}^n \tag{2.8}$$

and thus for any (local) minimum $x^\star$ such that $\nabla f(x^\star) = 0$ one has $2L_1(f)\,(f(x) - f(x^\star)) \geq \|\nabla f(x)\|_2^2$, *e.g.*, see [Zho18]. Moreover, as [Nes11, Equation (6)]

$$|f(y) - f(x) - \langle \nabla f(x), y - x\rangle| \leq \frac{1}{2}L_1(f)\|x - y\|_2^2. \tag{2.9}$$

It follows from [Nes03, Lemma 1.2.4] that if $f \in C^2(\mathbb{R}^n)$ and satisfies

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_2 \leq L_2(f)\|x - y\|_2 \quad \forall x, y \in \mathbb{R}^n, \tag{2.10}$$

for some constant $L_2(f) \geq 0$, denoted $f \in C^{2,2}_{L_2}(\mathbb{R}^n)$, then,

$$|f(y) - f(x) - \langle \nabla f(x), y - x\rangle - \frac{1}{2}\langle \nabla^2 f(x)(y - x), y - x\rangle| \leq \frac{1}{6}L_2(f)\|x - y\|_2^3, \quad \forall x, y \in \mathbb{R}^n. \tag{2.11}$$

See that (2.10) is equivalent to

$$|u^\mathsf{T}\nabla^2 f(x)u - u^\mathsf{T}\nabla^2 f(y)u| \leq L_2(f)\|x - y\|_2 \quad \forall x, y \in \mathbb{R}^n,\ u \in \mathbb{S}^{n-1}, \tag{2.12}$$

which is commonly referred to as f being 3rd order smooth (*cf.* [BP16]). Now it follows directly from the definition of a derivative that $f \in C^{2,2}_{L_2(f)}(\mathbb{R}^n)$ implies that

$$\left|\partial_t^3 f(x + tu)|_{t=0}\right| \leq L_2(f), \quad \forall u \in \mathbb{S}^{n-1}. \tag{2.13}$$

3 An imaginary oracle

In this section we present the main tool of this work. Using ideas from [Nes11; NS17] we provide the complex-step generalization of the result in [NY83; FKM04].

To state the main result of this section we define a uniform complex-step δ-smoothed version of f by

$$f_\delta(x) := \mathbb{E}_{v\sim\mathbb{B}^n}\left[\Re\left(f(x+i\delta v)\right)\right]. \tag{3.1}$$

Here, the parameter $\delta \in \mathbb{R}_{>0}$ is the tuneable smoothing parameter and relates to the radius of the ball we average over. It is true that if one has prior structural knowledge of f, one could consider a different solid to smooth over. Note that if the domain $\mathcal{D} \subseteq \mathbb{R}^n$ of f is for example bounded, then, by Lemma 2.2, there exists $\bar{\delta}$ such that f_δ is well-defined for any $\delta \in [0, \bar{\delta})$. To aid readability we will mention it throughout, but from now on we will assume that Assumption 3.1 holds.

Assumption 3.1 (Holomorphic extension)**.** *The function $f : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}$ is real-analytic over the open set $\mathcal{D}$ and admits a holomorphic extension to $\mathcal{D} \times (\bar{\delta}, \bar{\delta})^n \subset \mathbb{C}^n$ for some $\bar{\delta} \in (0, 1)$.*

We remark that the 1 in the interval of $\bar{\delta}$ is larger than practically needed and merely helps in asserting that for any $\delta \in (0, \bar{\delta})$ one has $\delta^{p+1} \leq \delta^p$ for any $p \in \mathbb{N}_{\geq 0}$. Also, note that $\mathcal{D}$ need not be bounded, for example, consider $f(x) = \|Ax - b\|_2^2$ over $\mathbb{R}^n$, which is a real-entire function and as such $f \in H(\mathbb{C}^n)$.

Lemma 3.2 (Approximation quality of the complex-step function)**.** *Let $f \in C^\omega(\mathcal{D}) \cap C^{1,1}_{L_1(f)}(\mathcal{D})$ admit a holomorphic extension to $\mathcal{D} \times (-\bar{\delta}, \bar{\delta})^n \subset \mathbb{C}^n$. Then, for f_δ as in (3.1) and any $x \in \mathcal{D}$ there exists some constant $C_0 > 0$ such that for any $\delta \in [0, \bar{\delta})$*

$$|f_\delta(x) - f(x)| \leq \frac{C_0 \delta^2 L_1(f)}{2}, \tag{3.2a}$$

$$|f_\delta(x) - f(x)| \lesssim \frac{\delta^2 L_1(f)}{2}. \tag{3.2b}$$

The bound (3.2a) will be used in the generic case with $\delta \in [0, \bar{\delta})$, whereas (3.2b) provides for asymptotic (in δ) insights without universal constants.

Proof. It follows from the definition (3.1), $\mathbb{E}_{v\sim\mathbb{B}^n}[vv^\mathsf{T}] = \frac{1}{n+2} \cdot I_n$[4] and the Taylor expansion (2.5) that for all $x \in \mathcal{D}$

$$\begin{aligned} f_\delta(x) &= \mathbb{E}_{v\sim\mathbb{B}^n}\left[f(x) - \frac{1}{2}(\delta v)^\mathsf{T}\nabla^2 f(x)(\delta v) + O(\|\delta v\|_2^4)\right] \\ &= f(x) - \frac{1}{2}\delta^2 \mathbb{E}_{v\sim\mathbb{B}^n}\left[v^\mathsf{T}\nabla^2 f(x) v\right] + O(\delta^4) \\ &= f(x) - \frac{1}{2}\delta^2 \frac{1}{n+2}\mathrm{Tr}(\nabla^2 f(x)) + O(\delta^4). \end{aligned}$$

Since $\nabla^2 f(x) \preceq L_1(f) \cdot I_n$ and $\bar{\delta} < 1$, there exists some constant $C_0 > 0$ such that (3.2a) holds. Similarly, for sufficiently small δ the second-order term will dominate which yields (3.2b). □

In contrast to other works (*cf.* [Nes11]), it is important to remark that f_δ does not always belongs to the same function class as f. For example, $f(x) := 1/(1+x^2)$ has a Lipschitz continuous gradient with $L_1(f) = 2$, while $\lim_{\delta\uparrow 1} f(0+i\delta) = +\infty$ and as such this makes the function f_δ not necessarily a member of $C^{1,1}_{L_1(f_\delta)}$. Also, convexity of f does not necessarily carry over, see Example 3.3 below.

[4] To see this, the covariance matrix $\mathbb{E}_{v\sim\mathbb{B}^n}[vv^\mathsf{T}]$ must be isotropic. Hence, for any $i \in [n]$ $\mathbb{E}_{v\sim\mathbb{B}^n} v_i^2 = \frac{1}{n}\mathbb{E}_{v\sim\mathbb{B}^n}\|v\|_2^2$ equals $(C/n)\int_0^1 r^{n-1} r^2 dr = (C/n)\frac{1}{n+2}$, for r the norm (radius) of $v \in \mathbb{B}^n$. To find C, use the fact that we integrate over a probability measure, that is, $1 = C\int_0^1 r^{n-1} dr$, which implies that $(C/n) = 1$ from where the result follows.

Example 3.3 (Convexity under complex lifting)**.** Let $f : \mathbb{R} \to \mathbb{R}$ be real-entire, then we find that

$$\Re(f(x + i\delta v)) = f(x) - \sum_{k=1}^{\infty} \frac{f^{(2k)}(x)}{(2k)!}(\delta v)^{2k}.$$

If $f(x) = x^2$, then $\Re(f(x+i\delta v)) = x^2-(\delta v)^2$ which is a convex function in x, regardless of the choice of $\delta > 0$ and $v \in \mathbb{R}$. However, for $f(x) = x^4$ one finds that $\Re(f(x+i\delta v)) = x^4-12x^2(\delta v)^2-24(\delta v)^4$, which is a non-convex function in x for any sufficiently large $\delta > 0$ and non-zero v, *e.g.*, see that $\partial_x^2\Re(f(x + i\delta v)) = 12x - 12(\delta v)^2$.

As f_δ need not be convex and might not have a Lipschitz continuous gradient, analyses of algorithms based on the oracle (3.5) require some extra machinery beyond [NS17].

Now, we state one of the key contributions, which is the integral representation of ∇f_δ. Lemma 3.4 below is the complex-step version of the approach as proposed in [NY83, Section 9.3], see *e.g.*, [FKM04, Lemma 1].

Lemma 3.4 (The gradient of the complex-step function)**.** *Let $f \in C^\omega(\mathcal{D})$ admit a holomorphic extension to $\mathcal{D} \times (-\bar{\delta}, \bar{\delta})^n \subset \mathbb{C}^n$, then f_δ as in (3.1) is differentiable and for any $x \in \mathcal{D}$ we have for any $\delta \in (0, \bar{\delta})$*

$$\frac{n}{\delta} \cdot \mathbb{E}_{u\sim\mathbb{S}^{n-1}} \left[\Im\left(f(x + i\delta u)\right) u\right] = \nabla f_\delta(x). \tag{3.3}$$

Proof. Under the assumption that f is holomorphic over a subset of $\mathbb{C}^n$, one can appeal to the Cauchy-Riemann equations [Rud87, Theorem 11.2] from where it follows that for $n = 1$

$$\nabla_x \int_{-\delta}^{\delta} \Re\left(f(x + iv)\right) dv = \Im\left(f(x + iu)\right)|_{-\delta}^{\delta}.$$

The additional scaling factor n/δ is further explained below. For $n \geq 2$, recall that Stokes' theorem tells us that $\int_\Omega d\omega = \int_{\partial\Omega} \iota^\star\omega$ for $\iota : \partial\Omega \hookrightarrow \Omega$ [Theorem 7.2.8][AMR88]. After a combination of Stokes' theorem and again the Cauchy-Riemann equations we obtain a result similar to the one as stated in [NY83; FKM04]:

$$\nabla_x \int_{\delta\mathbb{B}^n} \Re\left(f(x + iv)\right) dv = \int_{\delta\mathbb{S}^{n-1}} \Im\left(f(x + iu)\right) \frac{u}{\|u\|_2} du. \tag{3.4}$$

Regarding the scaling factor $u/\|u\|_2$ see for example [Lee13, Proposition 15.21]. Moreover, due to the assumption that the vectors v and u are uniformly distributed one obtains

$$f_\delta(x) = \mathbb{E}_{v\sim\mathbb{B}^n} \left[\Re\left(f(x + i\delta v)\right)\right] = \frac{1}{\mathrm{vol}(\delta\mathbb{B}^n)} \int_{\delta\mathbb{B}^n} \Re\left(f(x + iv)\right) dv,$$

and similarly,

$$\mathbb{E}_{u\sim\mathbb{S}^{n-1}} \left[\Im\left(f(x + i\delta u)\right) u\right] = \frac{1}{\mathrm{vol}(\delta\mathbb{S}^{n-1})} \int_{\delta\mathbb{S}^{n-1}} \Im\left(f(x + iu)\right) \frac{u}{\|u\|_2} du.$$

Combining the aforementioned results with the fact that $\mathrm{vol}(\delta\mathbb{B}^n)/\mathrm{vol}(\delta\mathbb{S}^{n-1}) = \delta/n$ allows for concluding the proof. □

The proof of Lemma 3.4 clearly shows why we went through the Cauchy-Riemann equations, we critically rely on them to pass from (3.1) to (3.3). Moreover, Lemma 3.4 provides us with a natural single-point oracle candidate

$$g_\delta(x) = \frac{n}{\delta}\Im\left(f(x + i\delta u)\right) u, \quad u \sim \mathbb{S}^{n-1}. \tag{3.5}$$

Then, the following result allows for showing consistency below and implies in particular that $\mathbb{E}_{u\sim\mathbb{S}^{n-1}}[uu^\mathsf{T}] = (1/n)I_n$.

Lemma 3.5 (Integration over the n-sphere)**.** *Given any $x \in \mathbb{R}^n$, then*

$$\frac{n}{\mathrm{vol}(\mathbb{S}^{n-1})} \cdot \int_{\mathbb{S}^{n-1}} \langle x, u\rangle u du = x. \tag{3.6}$$

Proof. First, rewrite (3.6) as $n \cdot \int_{\mathbb{S}^{n-1}} uu^\mathsf{T} du\, x$, where uu^T represents an outer-product, that is, $uu^\mathsf{T}x = \langle x, u\rangle u$. Now we would like to show that $n \cdot \int_{\mathbb{S}^{n-1}} uu^\mathsf{T} du = \mathrm{vol}(\mathbb{S}^{n-1}) \cdot I_n$. To that end, use the geometric tracing identity $n \cdot \int_{\mathbb{S}^{n-1}} \langle Xu, u\rangle du = \mathrm{Tr}(X) \cdot \mathrm{vol}(\mathbb{S}^{n-1})$, differentiating both sides with respect to X yields $n \cdot \int_{\mathbb{S}^{n-1}} uu^\mathsf{T} du = \mathrm{vol}(\mathbb{S}^{n-1}) \cdot I_n$ indeed, which concludes the proof. □

Since f is real-analytic, $f \in C^r(\mathcal{D})$ for all $r \geq 1$ and as such the directional derivative at $x \in \mathcal{D}$ in the direction $u \in \mathbb{S}^{n-1}$ is well-defined and given by $\langle \nabla f(x), u\rangle$. Now, since

$$\langle \nabla f(x), u\rangle = \lim_{\delta \downarrow 0} \frac{1}{\delta} \Im\left(f(x + i\delta u)\right),$$

we observe from (3.3) that the approximation is asymptotically consistent, that is,

$$\lim_{\delta \downarrow 0} \nabla f_\delta(x) = \frac{n}{\mathrm{vol}(\mathbb{S}^{n-1})} \cdot \int_{\mathbb{S}^{n-1}} \langle \nabla f(x), u\rangle u du \overset{(3.6)}{=} \nabla f(x). \tag{3.7}$$

Exactly this construction was the key observation in [ADX10; Nes11] to reduce oracle variance and indeed. This observation does not hold for other single-point oracles (*cf.* [FKM04, Section 1.1]).

Proposition 3.6 (Gradient approximation quality)**.** *Let $f \in C^\omega(\mathcal{D}) \cap C^{2,2}_{L_2(f)}(\mathcal{D})$, $L_2(f) > 0$, admit a holomorphic extension to $\mathcal{D} \times (-\bar{\delta}, \bar{\delta})^n \subseteq \mathbb{C}^n$. Then, for all $x \in \mathcal{D}$ there is a constant $C_1 \geq 0$ such that for any $\delta \in [0, \bar{\delta})$ one has*

$$\|\nabla f_\delta(x) - \nabla f(x)\|_2 \leq \frac{C_1 n \delta^2 L_2(f)}{6}, \tag{3.8a}$$

$$\|\nabla f_\delta(x) - \nabla f(x)\|_2 \lesssim \frac{n\delta^2 L_2(f)}{6}. \tag{3.8b}$$

Proof. Using a similar approach as in [Nes11, Lemma 3] we see that from the Taylor series expansion of $f(x + i\delta u)$ around x that

$$\begin{aligned}
\|\nabla f_\delta(x) - \nabla f(x)\|_2 &\overset{(3.7)}{\leq} \frac{n}{\delta \cdot \mathrm{vol}(\mathbb{S}^{n-1})} \int_{\mathbb{S}^{n-1}} \left|\Im\left(f(x + i\delta u)\right) - \delta\langle \nabla f(x), u\rangle\right| \|u\|_2 du \\
&\overset{(2.13)}{\leq} \frac{n}{\delta \cdot \mathrm{vol}(\mathbb{S}^{n-1})} \int_{\mathbb{S}^{n-1}} \left(\frac{\delta^3 \cdot L_2(f)}{6} + O(\delta^5)\right) \|u\|_2 du \\
&= \frac{1}{6}\delta^2 \cdot n \cdot L_2(f) + O(n \cdot \delta^4).
\end{aligned}$$

Since $\bar{\delta} < 1$, this implies that there is a constant $C_1 \geq 0$ such that (3.8a) holds. For sufficiently small δ the second order term δ^2 will dominate which yields (3.8b). □

We see that our simple *single-point* approach allows for an error of the form $O(\delta^2)$ which is what [Nes11] could get using a Gaussian smoothed *multi*-point approach under the assumption that $f \in C^{2,2}_{L_2(f)}$. Removing the constant in (3.8a) Proposition 3.6 is possible by appealing to for example the Cauchy integral representation theorem, but at the cost of demanding non-traditional bounds on f over a complex domain.

Algorithm 1 Complex δ-smoothed zeroth order unconstrained optimization

Input: initial iterate $x_0 \in \mathbb{R}^n$, stepsizes $\{\mu_k\}_{k\geq 0}$, smoothing parameters $\{\delta_k\}_{k\geq 0}$.
for $k = 0, 1, 2, \ldots, K-1$ **do**
  generate random $u_k \sim \mathbb{S}^{n-1}$
  set $g_{\delta_k}(x_k) = \frac{n}{\delta_k}\Im\left(f(x_k + i\delta_k u_k)\right) u_k$
  set $x_{k+1} = x_k - \mu_k \cdot g_{\delta_k}(x_k)$
end for

Algorithm 2 Complex δ-smoothed zeroth order constrained optimization

Input: initial iterate $x_0 \in \mathbb{R}^n$, stepsizes $\{\mu_k\}_{k\geq 0}$, smoothing parameters $\{\delta_k\}_{k\geq 0}$.
for $k = 0, 1, 2, \ldots, K-1$ **do**
  generate random $u_k \sim \mathbb{S}^{n-1}$
  set $g_{\delta_k}(x_k) = \frac{n}{\delta_k}\Im\left(f(x_k + i\delta_k u_k)\right) u_k$
  set $x_{k+1} = \Pi_{\mathcal{K}}\left(x_k - \mu_k \cdot g_{\delta_k}(x_k)\right)$
end for

4 Convex optimization

The goal in this section is to solve convex optimization problems inspired by the zeroth order framework as proposed in [Nes11; NS17], yet, by using the complex-step oracle g_δ as given by (3.5). The algorithm for the unconstrained case is detailed in Algorithm 1. What is more, given some non-empty convex compact set $\mathcal{K} \subset \mathcal{D} \subseteq \mathbb{R}^n$, we also consider constrained convex optimization problems of the form

$$\underset{x\in\mathcal{K}}{\text{minimize}} \quad f(x), \tag{4.1}$$

again, using the zeroth order framework. This algorithm is detailed in Algorithm 2. Here, $\Pi_{\mathcal{K}} : \mathcal{D} \subseteq \mathbb{R}^n \to \mathcal{K}$ denotes the projection operator.

To characterize the effectiveness of both Algorithm 1 and Algorithm 2, we need to bound the variance of the oracle (3.5). We observe the same attractive property as in [Nes11; NS17], there is no need to assume boundedness of the second moment of our random oracle.

Lemma 4.1 (Oracle variance)**.** *Let $f \in C^\omega(\mathcal{D}) \cap C^{2,2}_{L_2(f)}(\mathcal{D})$ admit a holomorphic extension to $\mathcal{D} \times (-\bar{\delta}, \bar{\delta})^n \subseteq \mathbb{C}^n$. Then, for $g_\delta(x)$ as in (3.5) and any $\delta \in [0, \bar{\delta})$ there is a constant $C_2 \geq 0$ such that*

$$\mathbb{E}_{u\sim\mathbb{S}^{n-1}}\left[\|g_\delta(x)\|_2^2\right] \leq \frac{C_2 n^2 L_2(f)^2 \delta^4}{36} + n\|\nabla f(x)\|_2^2, \tag{4.2a}$$

$$\mathbb{E}_{u\sim\mathbb{S}^{n-1}}\left[\|g_\delta(x)\|_2^2\right] \lesssim \frac{n^2 L_2(f)^2 \delta^4}{36} + n\|\nabla f(x)\|_2^2. \tag{4.2b}$$

Proof. First, observe from Algorithm 1 that

$$\mathbb{E}_{u\sim\mathbb{S}^{n-1}}\left[\|g_\delta(x)\|_2^2\right] = \frac{n^2}{\delta^2}\mathbb{E}_{u\sim\mathbb{S}^{n-1}}\left[\left(\Im\left(f(x+i\delta u)\right)\right)^2\right].$$

Then, since $f \in C^{2,2}_{L_2}(\mathcal{D})$ one has

$$\begin{aligned}\Im\left(f(x+i\delta u)\right) &= \Im\left(f(x+i\delta u)\right) - \langle\nabla f(x), \delta u\rangle + \langle\nabla f(x), \delta u\rangle \\ &\overset{(2.13)}{\leq} \frac{1}{6}\delta^3 \cdot L_2(f) + O(\delta^5) + \langle\nabla f(x), \delta u\rangle,\end{aligned}$$

such that

$$\mathbb{E}_{u\sim\mathbb{S}^{n-1}}\left[\|g_\delta(x)\|_2^2\right] \overset{(3.6)}{\leq} \frac{1}{36}n^2L_2(f)^2\delta^4 + n\|\nabla f(x)\|_2^2 + O(n^2\cdot\delta^6).$$

Hence, since $\bar{\delta} < 1$ there is constant $C_2 \geq 0$ such that for any $\delta \in [0,\bar{\delta})$ (4.2a) holds. For sufficiently small δ, the lowest order term in δ will dominate and hence we recover (4.2b). □

Lemma 4.1 follows from the fact that the leading term in (2.5) is $O(h^3)$. This should be contrasted with standard (purely real) results where this term would be $O(h^2)$. Note, [Nes11, Theorem 4] obtains a bound similar to (4.2a) under the assumption that $f \in C^{2,2}_{L_2(f)}$, yet, using a multi-point scheme. Also note that a variety of other zeroth order schemes have $O(1/\delta^p)$ showing up in their variance bounds which is however due to the measurement noise, *i.e.*, they have only access to $f(x)+\xi$, which is not the setting considered in this paper.

Theorem 4.2 (Convergence rate of Algorithm 1 and Algorithm 2)**.** *Let $f : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}$ be a convex function that admits a holomorphic extension to $\mathcal{D}\times(-\bar{\delta},\bar{\delta})^n \subseteq \mathbb{C}^n$ for some $0 < \bar{\delta} < 1$ and let $\mathcal{K} \subset \mathcal{D}$ be a compact convex set. Suppose that f has a Lipschitz gradient and Hessian, that is, (2.7) and (2.10) hold, for non-zero constants $L_1(f)$ and $L_2(f)$, respectively. Let $\{x_k\}_{k\geq 0}$ be the sequence of iterates generated by either Algorithm 1 or Algorithm 2, with constant stepsize $\mu_k \equiv \mu = 1/(2nL_1(f))$ and smoothing parameters $\{\delta_k\}_{k\geq 0} \subset (0,\bar{\delta})$. Then, there are constants $C_1, C_2, \ldots, C_b, C_c$ such that for any $K \geq 1$,*

(i) If $\delta_k \in (0,\bar{\delta})$, then the sequence $\{x_k\}_{k=0}^{K-1}$ satisifes

$$\begin{aligned}\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[f(x_k)-f(x^\star)\right] \leq & \frac{2nL_1(f)}{K}\|x_0-x^\star\|_2^2 + \frac{C_1nL_2(f)}{K}\|x_0-x^\star\|_2\sum_{k=0}^{K-1}\delta_k^2 \\ & + \frac{nL_2(f)^2}{2L_1(f)K}\left(C_2\left(\sum_{k=0}^{K-1}\delta_k^2\right)^2 + C_3\sum_{k=0}^{K-1}\delta_k^2\left(\sum_{k=0}^{K-1}\delta_k^4\right)^{\frac{1}{2}} + C_4\sum_{k=0}^{K-1}\delta_k^4\right);\end{aligned} \tag{4.3}$$

(ii) If $\delta_k \equiv \delta$ for some $\delta \in (0,\bar{\delta})$ then $\bar{x}_{K-1} := \frac{1}{K}\sum_{k=0}^{K-1}x_k$ satisfies

$$\mathbb{E}\left[f(\bar{x}_{K-1})-f(x^\star)\right] \leq \frac{2nL_1(f)}{K}\|x_0-x^\star\|_2^2 + C_anL_2(f)\delta^2\|x_0-x^\star\|_2 + \frac{C_bnL_2(f)^2K\delta^4}{L_1(f)}; \tag{4.4}$$

(iii) While if $\delta_k = \min\{\bar{\delta}, \delta/(k+1)\}$, for some $\delta \in (0,\bar{\delta})$, then

$$\mathbb{E}\left[f(\bar{x}_{K-1})-f(x^\star)\right] \leq \frac{n}{K}\left(\sqrt{2L_1(f)}\|x_0-x^\star\|_2 + \delta^2\frac{L_2(f)C_c}{\sqrt{2L_1(f)}}\right)^2. \tag{4.5}$$

Proof. We do the proof for Algorithm 1 and show that the extension to Algorithm 2 is immediate. The abbreviations C.S. and T.I. denote the Cauchy-Schwarz and triangle inequality, respectively.

Let $x^\star$ be any optimal solution. Define $r_k := \|x_k - x^\star\|_2$ for $k \geq 0$. Then, there are non-negative

constants C_1, C_2 such that

$$\begin{aligned}
\mathbb{E}_{u_k\sim\mathbb{S}^{n-1}}\left[r_{k+1}^2|x_k\right] &= \mathbb{E}_{u_k\sim\mathbb{S}^{n-1}}\left[r_k^2 - 2\mu_k\langle g_{\delta_k}(x_k), x_k - x^\star\rangle + \mu_k^2\|g_{\delta_k}(x_k)\|_2^2|x_k\right] \\
&\overset{(3.3)}{=} r_k^2 - 2\mu_k\langle \nabla f_{\delta_k}(x_k), x_k - x^\star\rangle + \mu_k^2\mathbb{E}_{u_k\sim\mathbb{S}^{n-1}}\left[\|g_{\delta_k}(x_k)\|_2^2|x_k\right] \\
&\overset{\text{C.S.}}{\leq} r_k^2 - 2\mu_k\langle \nabla f(x_k), x_k - x^\star\rangle + 2\mu_k\|\nabla f_{\delta_k}(x_k) - \nabla f(x_k)\|_2\|x_k - x^\star\|_2 \\
&\quad +\mu_k^2\mathbb{E}_{u_k\sim\mathbb{S}^{n-1}}\left[\|g_{\delta_k}(x_k)\|_2^2|x_k\right] \\
&\overset{(3.8a),(4.2a)}{\leq} r_k^2 - 2\mu_k\left(f(x_k) - f(x^\star)\right) + \mu_k\frac{C_1 n\delta_k^2 L_2(f)}{3}\|x_k - x^\star\|_2 \\
&\quad +\mu_k^2\left(\frac{1}{36}C_2 n^2 L_2(f)^2\delta_k^4 + n\|\nabla f(x_k)\|_2^2\right) \\
&\overset{(2.8)}{\leq} r_k^2 - 2\mu_k\left(f(x_k) - f(x^\star)\right) + \mu_k\frac{C_1 n\delta_k^2 L_2(f)}{3}\|x_k - x^\star\|_2 \\
&\quad +\mu_k^2\left(\frac{1}{36}C_2 n^2 L_2(f)^2\delta_k^4 + n2L_1(f)(f(x_k) - f(x^\star))\right),
\end{aligned}$$

which, upon rearranging and taking expectation on $u_1, \ldots, u_{k-1}$, yields

$$\begin{aligned}
\mathbb{E}\left[f(x_k) - f(x^\star)\right] &\leq \frac{1}{\mu}\left(\mathbb{E}\left[r_k^2\right] - \mathbb{E}\left[r_{k+1}^2\right]\right) + \frac{\delta_k^2 C_1 n L_2(f)}{3}\mathbb{E}\left[r_k\right] + \frac{\mu C_2 n^2 L_2(f)^2\delta_k^4}{36} \\
&\leq \frac{1}{\mu}\left(\mathbb{E}\left[r_k^2\right] - \mathbb{E}\left[r_{k+1}^2\right]\right) + \frac{\delta_k^2 C_1 n L_2(f)}{3}\sqrt{\mathbb{E}\left[r_k^2\right]} + \frac{\mu C_2 n^2 L_2(f)^2\delta_k^4}{36}.
\end{aligned}$$

Then, summing the above inequalities for k from 0 to $K-1$ gives

$$\sum_{k=0}^{K-1}\mathbb{E}\left[f(x_k) - f(x^\star)\right] \leq \frac{1}{\mu}\left(r_0^2 - \mathbb{E}\left[r_K^2\right]\right) + \frac{C_1 n L_2(f)}{3}\sum_{k=0}^{K-1}\delta_k^2\sqrt{\mathbb{E}\left[r_k^2\right]} + \frac{\mu C_2 n^2 L_2(f)^2}{36}\sum_{k=0}^{K-1}\delta_k^4. \tag{4.6}$$

For $k \geq 0$, letting

$$s_k = \sqrt{\mathbb{E}\left[r_k^2\right]},$$

we then have

$$s_K^2 \overset{(4.6)}{\leq} s_0^2 + \frac{\mu C_1 n L_2(f)}{3}\sum_{k=0}^{K-1}\delta_k^2 s_k + \frac{\mu^2 C_2 n^2 L_2(f)^2}{36}\sum_{k=0}^{K-1}\delta_k^4. \tag{4.7}$$

Applying Lemma A.1 to (4.7) yields

$$\begin{aligned}
s_K &\leq \frac{\mu C_1 n L_2(f)}{6}\sum_{k=0}^{K-1}\delta_k^2 + \left(s_0^2 + \frac{\mu^2 C_2 n^2 L_2(f)^2}{36}\sum_{k=0}^{K-1}\delta_k^4 + \left(\frac{\mu C_1 n L_2(f)}{6}\sum_{k=0}^{K-1}\delta_k^2\right)^2\right)^{\frac{1}{2}} \\
&\overset{\text{T.I.}}{\leq} \frac{\mu C_1 n L_2(f)}{3}\sum_{k=0}^{K-1}\delta_k^2 + s_0 + \left(\frac{\mu^2 C_2 n^2 L_2(f)^2}{36}\sum_{k=0}^{K-1}\delta_k^4\right)^{\frac{1}{2}}.
\end{aligned} \tag{4.8}$$

Substituting (4.8) into (4.6), we get (4.3) from

$$\begin{aligned}
&\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[f(x_k)-f(x^\star)\right]\\
\leq&\frac{1}{\mu K}\|x_0-x^\star\|_2^2+\frac{C_1nL_2(f)}{3K}\sum_{k=0}^{K-1}\delta_k^2\left(\frac{\mu C_1nL_2(f)}{3}\sum_{k=0}^{K-1}\delta_k^2+s_0+\left(\frac{\mu^2C_2n^2L_2(f)^2}{36}\sum_{k=0}^{K-1}\delta_k^4\right)^{\frac{1}{2}}\right)\\
&+\frac{\mu C_2n^2L_2(f)^2}{36K}\sum_{k=0}^{K-1}\delta_k^4\\
\leq&\frac{1}{\mu K}\|x_0-x^\star\|_2^2+\frac{C_1nL_2(f)}{3K}\|x_0-x^\star\|_2\sum_{k=0}^{K-1}\delta_k^2+\frac{\mu C_1^2n^2L_2(f)^2}{9K}\left(\sum_{k=0}^{K-1}\delta_k^2\right)^2\\
&+\frac{\mu C_1C_2^{\frac{1}{2}}n^2L_2(f)^2}{18K}\left(\sum_{k=0}^{K-1}\delta_k^2\right)\left(\sum_{k=0}^{K-1}\delta_k^4\right)^{\frac{1}{2}}+\frac{\mu C_2n^2L_2(f)^2}{36K}\sum_{k=0}^{K-1}\delta_k^4.
\end{aligned}\tag{4.9}$$

Without loss of generality, let both $C_1\geq 1$ and $C_2\geq 1$. Then, if $\delta_k\equiv\delta\in(0,\bar{\delta})$, we derive (4.4) from inequality (4.9)

$$\begin{aligned}
&\mathbb{E}\left[f(\bar{x}_{K-1})-f(x^\star)\right]\leq\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[f(x_k)-f(x^\star)\right]\\
\leq&\frac{1}{\mu K}\|x_0-x^\star\|_2^2+\frac{C_1nL_2(f)\delta^2}{3}\|x_0-x^\star\|_2+\frac{\mu C_2n^2L_2(f)^2\delta^4}{36}\\
&+\frac{\mu C_1^2n^2L_2(f)^2K\delta^4}{9}+\frac{\mu C_1C_2^{\frac{1}{2}}n^2L_2(f)^2\sqrt{K}\delta^4}{18}\\
\leq&\frac{1}{\mu K}\|x_0-x^\star\|_2^2+\frac{C_1nL_2(f)\delta^2}{3}\|x_0-x^\star\|_2+\frac{\mu C_1^2C_2n^2L_2(f)^2K\delta^4}{4}\\
=&\frac{2nL_1(f)}{K}\|x_0-x^\star\|_2^2+\frac{C_1nL_2(f)\delta^2}{3}\|x_0-x^\star\|_2+\frac{C_1^2C_2nL_2(f)^2K\delta^4}{8L_1(f)}.
\end{aligned}$$

If however $\delta_k=\min\left\{\bar{\delta},\delta/(k+1)\right\}$ for all $k\geq 0$, we derive (4.5) from inequality (4.9), the zeta function inequalities (A.1) and the convexity of f via

$$\begin{aligned}
\mathbb{E}\left[f(\bar{x}_{K-1})-f(x^\star)\right]\leq&\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[f(x_k)-f(x^\star)\right]\\
\leq&\frac{1}{\mu K}\|x_0-x^\star\|_2^2+\delta^2\frac{\pi^2C_1nL_2(f)}{18K}\|x_0-x^\star\|_2+\delta^4\frac{\mu\pi^4C_1^2n^2L_2(f)^2}{324K}\\
&+\delta^4\frac{\mu\pi^4C_1C_2^{\frac{1}{2}}n^2L_2(f)^2}{108\sqrt{90}K}+\delta^4\frac{\mu\pi^4C_2n^2L_2(f)^2}{3240K}\\
=&\frac{1}{\mu K}\|x_0-x^\star\|_2^2+\delta^2\frac{nL_2(f)C_3}{K}\|x_0-x^\star\|_2+\delta^4\frac{\mu n^2L_2(f)^2C_4}{K}\\
=&\frac{2nL_1(f)}{K}\|x_0-x^\star\|_2^2+\delta^2\frac{nL_2(f)C_3}{K}\|x_0-x^\star\|_2+\delta^4\frac{nL_2(f)^2C_4}{2L_1(f)K}\\
\leq&\frac{n}{K}\left(\sqrt{2L_1(f)}\|x_0-x^\star\|_2+\delta^2\frac{L_2(f)C_5}{\sqrt{2L_1(f)}}\right)^2.
\end{aligned}$$

This completes the proof for Algorithm 1.

Now consider Algorithm 2. Let $r_k = \|x_k - x^\star\|_2$, then since $\mathcal{K}$ is a compact convex set it follows that

$$r_{k+1}^2 \leq \|x_k - \mu_k g_{\delta_k}(x_k) - x^\star\|_2^2 = r_k^2 - 2\mu_k \langle g_{\delta_k}(x_k), x_k - x^\star \rangle + \mu_k^2 \|g_{\delta_k}(x_k)\|_2^2.$$

Hence, we can appeal to proof of Algorithm 1 and conclude on the convergence rate. □

Combining all asymptotic (in δ) error bounds (3.2b), (3.8b), (4.2b) with (4.5), then all universal constants can be taken to be 1 and as such it can be observed that if δ is sufficiently small indeed, then, one needs

$$O\left(\frac{nL_1(f)\|x_0 - x^\star\|_2^2}{\epsilon}\right) \tag{4.10}$$

iterations to have $\mathbb{E}\left[f(\bar{x}_{K-1}) - f(x^\star)\right] \leq \epsilon$. This is precisely the first-order, *i.e., with* $\nabla f(x)$ being available, complexity scaled by the dimension n [Nes03, Section 2.1.5].

5 Strongly convex optimization

In this part the previous results on unconstrained optimization are extended to the setting of f being $\tau(f)$-*strongly* convex over $\mathcal{D}$, *i.e.,* there is some $\tau(f) > 0$ such that

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\tau(f)}{2}\|y - x\|_2^2, \quad \forall x, y \in \mathcal{D}. \tag{5.1}$$

In particular (5.1) implies

$$f(x) - f(x^\star) \geq \frac{\tau(f)}{2}\|x - x^\star\|_2^2, \quad \forall x \in \mathcal{D}. \tag{5.2}$$

If additionally $f \in C^{1,1}_{L_1(f)}$, then

$$\tau(f)\|x - x^\star\|_2 \leq \|\nabla f(x)\|_2 \leq L_1(f)\|x - x^\star\|_2. \tag{5.3}$$

Theorem 5.1 (Convergence rate of Algorithm 1 under strong convexity)**.** *Let $f : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}$ be a $\tau(f)$-strongly convex function that admits a holomorphic extension to $\mathcal{D} \times (-\bar{\delta}, \bar{\delta})^n \subseteq \mathbb{C}^n$ for some $0 < \bar{\delta} < 1$. Suppose that f has a Lipschitz gradient and Hessian, that is, (2.7) and (2.10) hold, for non-zero constants $L_1(f)$ and $L_2(f)$, respectively. Let $\{x_k\}_{k\geq 0}$ be the sequence of iterates generated by Algorithm 1 with constant stepsize $\mu_k \equiv \mu = 1/(2nL_1(f))$ and the sequence of smoothing parameters defined for all $k \geq 0$ by $\delta_k = \min\{\bar{\delta}, \delta/(k+1)\}$ for some $\delta \in (0, \bar{\delta})$. Then, there are non-negative constants C_a, C_b such that if we let*

$$\nu_{\tau,\delta} = \frac{\delta^2 n L_1(f)}{\tau(f)}\left(\|x_0 - x^\star\|_2^2 \frac{C_a L_2(f)}{L_1(f)} + \delta^2 \frac{C_b L_2(f)^2}{L_1(f)^2}\right). \tag{5.4}$$

then, for any $K \geq 0$, x_{K+1} satisfies

$$\mathbb{E}[f(x_{K+1}) - f(x^\star)] \leq \frac{1}{2}L_1(f)\left(\nu_{\tau,\delta} + \left(1 - \frac{\tau(f)}{4nL_1(f)}\right)^K (\|x_0 - x^\star\|_2^2 - \nu_{\tau,\delta})\right). \tag{5.5}$$

Proof. Let $x^\star$ be any optimal solution. Define $r_k := \|x_k - x^\star\|_2$ and $\rho_{k+1} := \mathbb{E}_{u_k \sim \mathbb{S}^{n-1}}\left[r_{k+1}^2 | x_k\right]$ for $k \geq 0$. Then, it follows from the proof of Theorem 4.2 that there are non-negative constants

C_1, C_2 such that

$$\begin{aligned}\rho_{k+1} &\overset{(2.8)}{\leq} r_k^2 - 2\mu_k\left(f(x_k) - f(x^\star)\right) + \mu_k \frac{C_1 n \delta_k^2 L_2(f)}{3}\|x_k - x^\star\|_2 \\ &\quad + \mu_k^2\left(\frac{1}{36}C_2 n^2 L_2(f)^2\delta_k^4 + n2L_1(f)(f(x_k) - f(x^\star))\right) \\ &\overset{(5.2)}{\leq} \left(1 - \mu_k\frac{\tau(f)}{2}\right) r_k^2 + \mu_k \frac{C_1 n\delta_k^2 L_2(f)}{3} r_k + \mu_k^2\left(\frac{1}{36}C_2 n^2 L_2(f)^2\delta_k^4\right).\end{aligned}$$

Therefore

$$\rho_{k+1} \leq \left(1 - \frac{\tau(f)}{4nL_1(f)}\right)\rho_k + \delta_k^2 \frac{C_1 L_2(f)}{6L_1(f)}\sqrt{\rho_k} + \delta_k^4 \frac{C_2 L_2(f)^2}{144 L_1(f)^2} \tag{5.6a}$$

$$\overset{(5.3)}{\leq} \rho_k + \delta_k^2 \frac{C_1 L_2(f)}{6L_1(f)}\sqrt{\rho_k} + \delta_k^4 \frac{C_2 L_2(f)^2}{144 L_1(f)^2} \tag{5.6b}$$

Now summing up over k and using a telescoping argument as in the proof of Theorem 4.2 yields

$$\rho_K \leq \rho_0 + \sum_{k=0}^{K-1} \delta_k^2 \frac{C_1 L_2(f)}{6L_1(f)}\sqrt{\rho_k} + \sum_{k=0}^{K-1} \delta_k^4 \frac{C_2 L_2(f)^2}{144 L_1(f)^2}. \tag{5.7}$$

Then, applying Lemma A.1 to (5.7) yields

$$\begin{aligned}\sqrt{\rho_K} &\leq \frac{1}{2}\sum_{k=0}^{K-1} \delta_k^2 \frac{C_1 L_2(f)}{6L_1(f)} + \left(\rho_0^2 + \sum_{k=0}^{K-1} \delta_k^4 \frac{C_2 L_2(f)^2}{144 L_1(f)^2} + \left(\frac{1}{2}\sum_{k=0}^{K-1} \delta_k^2 \frac{C_1 L_2(f)}{6L_1(f)}\right)^2\right)^{\frac{1}{2}} \\ &\overset{\text{T.I.}}{\leq} \sum_{k=0}^{K-1} \delta_k^2 \frac{C_1 L_2(f)}{6L_1(f)} + \rho_0 + \left(\sum_{k=0}^{K-1} \delta_k^4 \frac{C_2 L_2(f)^2}{144 L_1(f)^2}\right)^{\frac{1}{2}}.\end{aligned}$$

Plugging this bound on $\sqrt{\rho_K}$ back into (5.6a) yields

$$\begin{aligned}\rho_{K+1} \leq \left(1 - \frac{\tau(f)}{4nL_1(f)}\right)\rho_K + \delta_K^4 \frac{C_2 L_2(f)^2}{144 L_1(f)^2} \\ + \delta_K^2 \frac{C_1 L_2(f)}{6L_1(f)}\left(\sum_{k=0}^{K-1} \delta_k^2 \frac{C_1 L_2(f)}{6L_1(f)} + \rho_0 + \left(\sum_{k=0}^{K-1} \delta_k^4 \frac{C_2 L_2(f)^2}{144 L_1(f)^2}\right)^{\frac{1}{2}}\right).\end{aligned}$$

Recall the zeta bound (A.1) and let us select $\delta_k = \min\{\bar{\delta}, \delta/(k+1)\}$ such that

$$\rho_{K+1} \leq \left(1 - \frac{\tau(f)}{4nL_1(f)}\right)\rho_K + \delta^4 \frac{\pi^2 C_1^2 L_2(f)^2}{216 L_1(f)^2} + \delta^2 \rho_0 \frac{C_1 L_2(f)}{6L_1(f)} + \delta^4 \frac{C_2 L_2(f)^2}{144 L_1(f)^2} + \delta^4 \frac{\pi^2 C_1 C_2^{\frac{1}{2}} L_2(f)^2}{72\sqrt{90} L_1(f)^2}.$$

Hence, there are constants C_3 and C_4 such that

$$\rho_{K+1} \leq \left(1 - \frac{\tau(f)}{4nL_1(f)}\right)\rho_K + \delta^2 \rho_0 C_3 \frac{L_2(f)}{L_1(f)} + \delta^4 C_4 \frac{L_2(f)^2}{L_1(f)^2}.$$

Then, for $\nu_{\tau,\delta}$ as in (5.4) we get

$$(\rho_{K+1} - \nu_{\tau,\delta}) \leq \left(1 - \frac{\tau(f)}{4nL_1(f)}\right)(\rho_K - \nu_{\tau,\delta}) \leq \left(1 - \frac{\tau(f)}{4nL_1(f)}\right)^K (\rho_0 - \nu_{\tau,\delta}).$$

At last, it follows from (2.9) that $\mathbb{E}[f(x_k) - f(x^\star)] \leq \frac{1}{2}L_1(f)\rho_k$, which concludes the proof. □

Now, contrasting Theorem 4.2 with Theorem 5.1 it follows directly from (5.4) and (5.5) that to have $\mathbb{E}\left[f(x_{K+1}) - f(x^\star)\right] \leq \epsilon$ one needs a sufficiently small δ satisfying

$$\delta \leq O\left(\sqrt{\frac{\epsilon\tau(f)}{nL_1(f)^2}}\right) \tag{5.8}$$

and

$$O\left(\frac{nL_1(f)}{\tau(f)}\log\left(\frac{L_1(f)\|x_0 - x^\star\|_2^2}{\epsilon}\right)\right) \tag{5.9}$$

iterations. Comparing the rate (5.9) to (4.10), then we see — as is known, see *e.g.,* [Nes11]— that strong convexity allows for a faster scheme.

6 Non-convex optimization

At last we consider a local minima in a possibly non-convex program. Our line of proof is different from [Nes11, Section 7] as our smoothed function f_δ does not necessarily has a Lipshitz continuous gradient. In this particular setting, the set $\mathcal{D}$ functions as an open neighbourhood of a local minima.

Theorem 6.1 (Convergence rate of Algorithm 1 to a local minima)**.** *Let $f : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}$ be a — not necessarily convex — function that admits a holomorphic extension to $\mathcal{D} \times (-\bar{\delta}, \bar{\delta})^n \subseteq \mathbb{C}^n$ for some $0 < \bar{\delta} < 1$. Suppose that f has a Lipschitz gradient and Hessian, that is, (2.7) and (2.10) hold, for non-zero constants $L_1(f)$ and $L_2(f)$, respectively. Let $\{x_k\}_{k\geq 0}$ be the sequence of iterates generated by Algorithm 1 with constant stepsize $\mu_k \equiv \mu = 1/(nL_1(f))$ and the sequence of smoothing parameters defined for all $k \geq 0$ by $\delta_k = \min\{\bar{\delta}, \delta/(k+1)\}$ for some $\delta \in (0, \bar{\delta})$. Let $x^\star \in \mathcal{D}$ be a local minima of f, then, there is a non-negative constant C_a such that for all $K \geq 1$ the sequence $\{x_k\}_{k=0}^{K-1}$ satisfies*

$$\begin{aligned}\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|\nabla f(x_k)\|_2^2\right] \leq & \frac{n}{K}\Big[2L_1(f)\left(f(x_0) - f(x^\star)\right) \\ & + \delta^2 L_2(f)\left(\delta^2 L_2(f)(n+1)C_a + \sqrt{2nL_1(f)(f(x_0) - f(x^\star))}\right)\Big].\end{aligned} \tag{6.1}$$

Proof. As $f \in C^{1,1}_{L_1(f)}$ one has

$$\begin{aligned} f(x_{k+1}) &\overset{(2.9)}{\leq} f(x_k) - \mu_k\langle\nabla f(x_k), g_{\delta_k}(x_k)\rangle + \frac{1}{2}\mu_k^2 L_1(f)\|g_{\delta_k}(x_k)\|_2^2 \\ &= f(x_k) - \mu_k\|\nabla f(x_k)\|_2^2 - \mu_k\langle\nabla f(x_k), g_{\delta_k}(x_k) - \nabla f(x_k)\rangle + \frac{1}{2}\mu_k^2 L_1(f)\|g_{\delta_k}(x_k)\|_2^2. \end{aligned}$$

Now taking expectation, applying the Cauchy-Schwarz inequality and using both (3.8a) and (4.2a) results in

$$\begin{aligned}\mathbb{E}_{u_k\sim\mathbb{S}^{n-1}}\left[f(x_{k+1})|x_k\right] \leq & f(x_k) - \mu_k\|\nabla f(x_k)\|_2^2 + \mu_k\frac{C_1 n\delta_k^2 L_2(f)}{6}\|\nabla f(x_k)\|_2 \\ & + \frac{1}{2}\mu_k^2 L_1(f)\left(n\|\nabla f(x_k)\|_2^2 + \frac{C_2 n^2 L_2(f)^2\delta_k^4}{36}\right).\end{aligned}$$

Then, taking expectation over $u_1, \ldots, u_{k-1}$, plugging in our stepsize $\mu \equiv \mu_k = 1/(nL_1(f))$ applying Jensen's inequality and rearranging yields

$$\begin{aligned}\frac{\mu}{2}\mathbb{E}\left[\|\nabla f(x_k)\|_2\right]^2 \leq & \frac{\mu}{2}\mathbb{E}\left[\|\nabla f(x_k)\|_2^2\right] \\ & \leq \mathbb{E}[f(x_k) - f(x_{k+1})] + \frac{C_1\delta_k^2 L_2(f)}{6L_1(f)}\mathbb{E}\left[\|\nabla f(x_k)\|_2\right] + \frac{C_2 L_2(f)^2\delta_k^4}{72L_1(f)}.\end{aligned} \tag{6.2}$$

As we consider a local minima, assume that $f(x) \geq f(x^\star)$ and define $\phi_k := \mathbb{E}[\|\nabla f(x_k)\|_2]$, then, a telescoping argument yields

$$\phi_K^2 \leq 2nL_1(f)(f(x_0) - f(x^\star)) + \frac{nC_1L_2(f)}{3}\sum_{k=0}^{K}\delta_k^2\phi_k + \frac{nC_2L_2(f)^2}{36}\sum_{k=0}^{K}\delta_k^4.$$

Since $\|\nabla f(x_0)\|_2^2 \leq 2L_1(f)|f(x_0) - f(x^\star)|$ we can appeal to Lemma A.1 and obtain

$$\phi_K \leq \frac{nC_1L_2(f)}{6}\sum_{k=0}^{K}\delta_k^2 + \left(2nL_1(f(x_0) - f(x^\star)) + \frac{nC_2L_2(f)^2}{36}\sum_{k=0}^{K}\delta_k^4 + \left(\frac{nC_1L_2(f)}{6}\sum_{k=0}^{K}\delta_k^2\right)^2\right)^{\frac{1}{2}}.$$

Now we consider $\delta_k = \min\{\bar{\delta}, \delta/(k+1)\}$ such that after using (A.1) and an application of the triangle inequality we obtain

$$\begin{aligned}\phi_K &\leq \sqrt{2nL_1(f)(f(x_0) - f(x^\star))} + \delta^2\frac{nC_1L_2(f)\pi^2}{18} + \delta^2\frac{\sqrt{n}C_2^{\frac{1}{2}}L_2(f)\pi^2}{6\sqrt{90}}\\ &\leq \sqrt{2nL_1(f)(f(x_0) - f(x^\star))} + \delta^2\frac{nC_3L_2(f)\pi^2}{18}.\end{aligned}$$

Then, define $\theta_k^2 := \mathbb{E}\left[\|\nabla f(x_k)\|_2^2\right]$, plug ϕ_K back into (6.2) and obtain after a telescoping argument

$$\begin{aligned}\frac{1}{K+1}\sum_{k=0}^{K}\theta_k^2 \leq \frac{n}{K+1}\Big[&2L_1(f)\left(f(x_0) - f(x^\star)\right) + \delta^4C_4L_2(f)^2\\ &+ \delta^2L_2(f)\left(\delta^2nL_2(f)C_5 + \sqrt{2nL_1(f)(f(x_0) - f(x^\star))}\right)\Big].\end{aligned}$$

□

Given (6.1) we see that if δ is chosen sufficiently small, then, to achieve $(1/K)\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla f(x_k)\|_2^2] \leq \epsilon$ we need

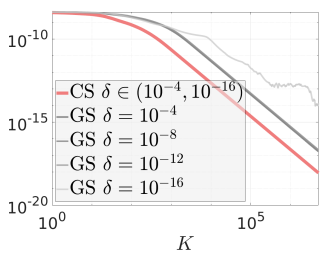
$$K \geq O\left(\frac{nL_1(f)\left(f(x_0) - f(x^\star)\right)}{\epsilon}\right).$$

7 Numerical experiments

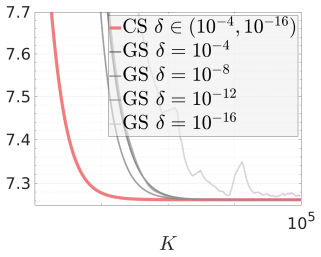
Especially in the zeroth order setting as considered here, one has usually no direct knowledge of the sharpest Lipschitz constants related to the objective function f. In such a case one needs to resort to an approximation and in fact for the Algorithms to work, the estimated Lipschitz constant should not under-approximate the real Lipschitz constant. However, as remarked throughout the literature — for deterministic oracles (*cf.* [Nes11, Equations (58, 59)])— a larger Lipschitz constant implies a smaller smoothing parameter δ to obtain a certain suboptimality gap ϵ, see also (5.8). Hence, having the freedom to pick δ arbitrarily small is preferable.

In the following experiments we will show that multi-point oracles do not allow for this option whereas our single-point method does. At the same time, if δ is sufficiently large such that multi-point methods work, we show that our method achieves a similar or faster convergence rate. All results above show that the (asymptotic) rates are the same as the ones provided in [NS17], yet we observe a slightly better dependency on n, that is, n versus $2(n+4)$. A full investigation of why our method leads to empirically faster convergence is left to future work.

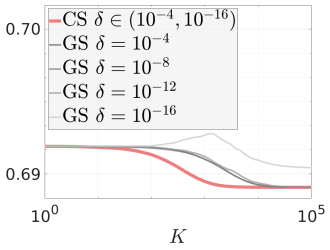
As mentioned throughout the introduction, the oracle (1.4) has severe limitations — most notably very slow convergence — which is why we omitted a comparison against it, this would



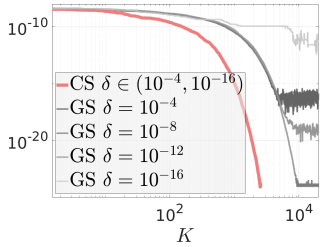
(ia) Suboptimality gap $f(\bar{x}_K) - f^\star$ for the test function (7.2).



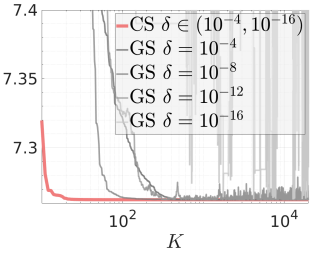
(iia) Cost $f(\bar{x}_K)$ for the Huber-like function (7.3).



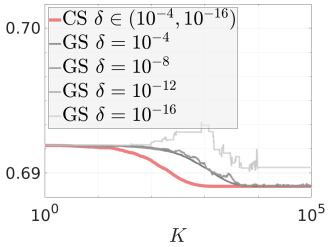
(iiia) Cost $f(\bar{x}_K)$ for logistic regression (7.5).



(ib) Suboptimality gap $f(x_K) - f^\star$ for the test function (7.2).



(iib) Cost $f(x_K)$ for the Huber-like function (7.3).



(iiib) Cost $f(x_K)$ for logistic regression (7.5).

Figure 7.1: The single-point Complex-smoothing (CS) method (Algorithm 1) compared to the multi-point Gaussian smoothing (GS) method from [NS17, Equation (54), oracle (7.1a)] on a variety of objective functions for a time-invariant smoothing parameter $\delta \equiv \delta_k$.

not be informative. Instead we compare against the Gaussian smoothing (GS) oracles from [NS17, Equation (30)], namely

$$g_{\delta,\mathsf{fd}}(x) = \frac{f(x+\delta u) - f(x)}{\delta} u, \quad u \sim \mathcal{N}(0, I_n), \tag{7.1a}$$

$$g_{\delta,\mathsf{cd}}(x) = \frac{f(x+\delta u) - f(x-\delta u)}{2\delta} u, \quad u \sim \mathcal{N}(0, I_n). \tag{7.1b}$$

As indicated, these oracles are based on finite-difference methods from numerical differentiation (2.6). When simulating (7.1a) and (7.1b) we will always use the stepsize $\mu_k \equiv 1/(4(n+4)L_1(f))$ as given by [NS17, Equation (55)].

7.1 Unconstrained convex optimization Throughout this section, we always compare against oracle (7.1a) unless indicated differently.

Example 7.1 (Quadratic test function)**.** First, we compare Algorithm 1 to [Nes11; NS17] on an ill-conditioned version of the test function[5] from [Nes03, Section 2.1.2]

$$f_n(x) = L\left(\frac{1}{2}\left[(x^{(1)})^2 + \sum_{i=1}^{n-1}(x^{(i+1)} - x^{(i)})^2 + (x^{(n)})^2\right] - x^{(1)}\right) \tag{7.2}$$

for $x_0 = 0$, $L = 10^{-8}$, $L_1(f) = 4L$ and $(x^\star)^{(i)} = 1 - i/(n+1)$ with $x^{(i)}$ denoting the i^{th} component of $x \in \mathbb{R}^n$. Indeed, on such a quadratic function, the complex-step and two-point oracle as proposed

[5] Better known as the "*worst function in the world*".

in [Nes11; NS17] have a theoretically equivalent performance. However, fixing $n = 5$ and letting the smoothing parameter approach machine precision shows the critical difference between a one-point and multi-point method. See Figure 7.1ia and Figure 7.1ib for a single realization, where we compare the performance of $\bar{x}_K$ and x_K, respectively. Especially the performance of x_K is significantly better under the complex-step algorithm.

Example 7.2 (Smooth approximate ℓ_1-regularization)**.** Following [FG16], we are interested in solving a smoothly approximated version of a ℓ_1-regularized convex program. Specifically, we consider the pseudo-Huber loss given by

$$\psi_\mu(x) = \mu \sum_{i=1}^{m} \left(\sqrt{1 + \frac{x_i^2}{\mu^2}} - 1 \right) \tag{7.3}$$

and are interested in minimizing the objective

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2 + \lambda \psi_\mu(x) \tag{7.4}$$

over $x \in \mathbb{R}^n$ for some $\lambda > 0$ and data $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Now, it follows from [FG16, Lemma 2] that $L_1(f) = \lambda/\mu + \|A^\top A\|_2$, whereas it follows from [FG16, Lemma 6] that $L_2(f) = \lambda/\mu^2$. Now, we again compare Algorithm 1 to the method proposed in [Nes11; NS17]. Here we let A and b be random with unit covariance matrices for $m = 4$, $n = 2$. Moreover, $\lambda = \mu = 10^{-4}$. We show the costs $f(\bar{x}_K)$ and $f(x_k)$ in Figure 7.1iia and Figure 7.1iib, respectively, for a decreasing smoothing parameter δ. Again, a difference in numerical stability can be observed.

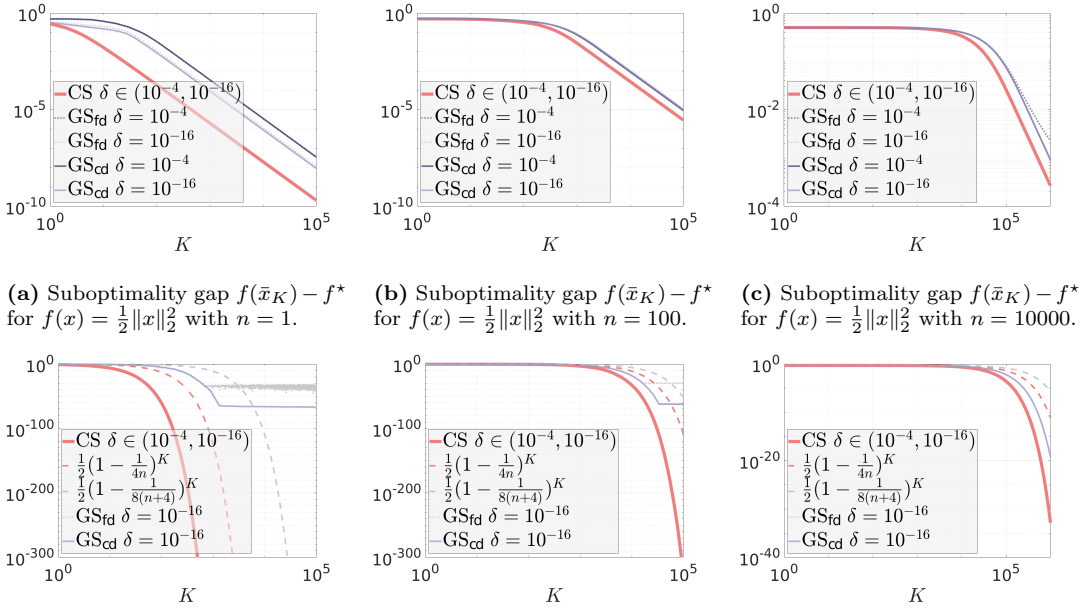
Example 7.3 (Logistic regression)**.** For the function

$$f(x) = \frac{1}{m} \sum_{i=1}^{m} \log\left(1 + \exp\left(-y_i a_i^\mathsf{T} x\right)\right) \tag{7.5}$$

with $y_i \in \{-1, 1\}$ and $a_i \sim \mathcal{N}(0, I_n)$, one can derive that $L_1(f) = \frac{1}{4}\|A\|_2$, for A the matrix filled with the vectors a_i. Let $m = 100$ and $n = 2$, then again, we can observe mild numerical issues in Figure 7.1iiia for a sufficiently small δ if one uses the averaged iterate $\bar{x}_K$. However, these are significantly more pronounced when one is interested in a faster algorithm and uses x_k as shown in Figure 7.1iiib.

Example 7.4 (The effect of dimension and strong convexity)**.** Figure 7.1 showed the effect of numerical cancellation, but also a clear difference in convergence speed. Looking back at Example 2.6 and comparing oracles (7.1a) and (7.1b) to the complex-step oracle (3.5) we expect that a speed-up should be visible be passing from the forward-difference to the central-difference oracle (7.1b). In Figure 7.2 we compare these oracles under an increase in dimension n on the function $f(x) = \frac{1}{2}\|x\|_2^2$ for $x_0 = (1/\sqrt{n})\mathbb{1}_n$. We observe that the severe numerical issues prevail, yet slightly less for higher dimensions. Concurrently, we observe throughout that averaging is more robust, but significantly slower. What is more, although the central-difference oracle (7.1b) does provide for a speed-up compared to (7.1a), the convergence remains slower than Algorithm 1 under the complex-step oracle (3.5). Also, plugging in $\tau(f) = 1$, $L_1(f) = 1$ and $\|x_0\|_2^2 = 1$ into the provided strong convexity rate from Theorem 5.1 we observe that $(1/2)(1 - 1/(4n))^K$ does capture the convergence rate of $f(x_k)$ to $f(x^\star)$. We compare this against the rate $(1/2)(1 - 1/(8(n+4)))^K$ provided by [NS17, Equation (57)].

7.2 Constrained convex optimization Next we do a control example where we iteratively update the input policy. In the unconstrained case one could consider the policy iteration schemes provided in [Faz+18; Mal+19].



(a) Suboptimality gap $f(\bar{x}_K) - f^\star$ for $f(x) = \frac{1}{2}\|x\|_2^2$ with $n = 1$.

(b) Suboptimality gap $f(\bar{x}_K) - f^\star$ for $f(x) = \frac{1}{2}\|x\|_2^2$ with $n = 100$.

(c) Suboptimality gap $f(\bar{x}_K) - f^\star$ for $f(x) = \frac{1}{2}\|x\|_2^2$ with $n = 10000$.

(d) Suboptimality gap $f(x_K) - f^\star$ for $f(x) = \frac{1}{2}\|x\|_2^2$ with $n = 1$.

(e) Suboptimality gap $f(x_K) - f^\star$ for $f(x) = \frac{1}{2}\|x\|_2^2$ with $n = 100$.

(f) Suboptimality gap $f(x_K) - f^\star$ for $f(x) = \frac{1}{2}\|x\|_2^2$ with $n = 10000$.

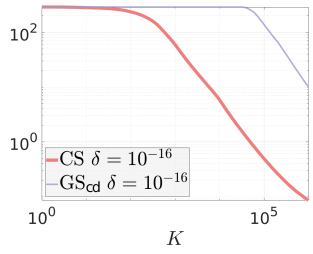
Figure 7.2: The single-point Complex-smoothing (CS) method (Algorithm 1) compared to the multi-point Gaussian smoothing (GS) method from [NS17, Equation (55)] for oracle (7.1a) (GS_{fd}) and oracle (7.1b) (GS_{cd}) as a function of the dimension n for a time-invariant smoothing parameter $\delta \equiv \delta_k$.

Example 7.5 (Policy iteration)**.** We are interested in solving the following Model Predictive Control (MPC) problem. There, given some prediction horizon $T \geq 1$ one wants to solve

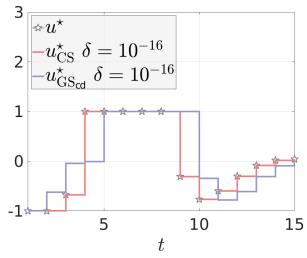
$$\begin{aligned} \underset{u := \{u_t\}_{t=0}^{T-1} \subset \mathbb{R}^{n_u}}{\text{minimize}} \quad & \sum_{t=0}^{T-1} \langle Qx_t, x_t \rangle + \langle Ru_t, u_t \rangle + \langle Qx_T, x_T \rangle \\ \text{subject to} \quad & x_{t+1} = Ax_t + Bu_t, \quad x_0 = x', \\ & \|u_t\|_\infty \leq 1. \end{aligned} \tag{7.6}$$

for the policy $u^\star(x') = u_0^\star(x'), \ldots, u_{T-1}^\star(x')$. Given $u^\star(x')$, only the first input, that is $u_0^\star(x')$ is implemented. This input maps x_0 to $x_1 = Ax_0 + Bu_0^\star(x_0)$. Now we set $x' = x_1$ and continue this iterative process for as long as we would like the control horizon to be. In this example we assume that the cost matrices $Q \in \mathcal{S}_{\succeq 0}^{n_x}$ and $R \in \mathcal{S}_{\succ 0}^{n_u}$ are given. However, we do not have access to the model parameters $A \in \mathbb{R}^{n_x \times n_x}$ and $B \in \mathbb{R}^{n_x \times n_u}$ directly, we can only simulate the performance. That is, given some policy $u' = \{u'_t\}_{t=0}^{T-1}$ we can simulate the dynamics and hence evaluate the corresponding cost in (7.6). Here, we let the tuple (A, B, Q, R) be the standard 2 dimensional Yalmip [Löf04] MPC instance[6] with $x_0 = (3, 1)$. We will construct the optimal sequence of control inputs $u_0^\star(x_0), u_0^\star(x_1), \ldots$, via Yalmip, and denote this as $u^\star$. Note that Problem (7.6) is of the form $\min_{u \in \mathcal{K}} f(u)$ for $\mathcal{K}$ a compact and convex set. Now we will show the performance of Algorithm 2 compared to oracle (7.1b) under a time-invariant $\delta_k \equiv \delta = 10^{-16}$ for $K = 10^6$ and $T = 4$. We use $u_0 = 0 \in \mathbb{R}^{n_u \cdot T}$ as our initialization and use $L_1(f) = 4 \cdot 10^4$ as our Lipschitz estimate. Note, this is a very rough estimate based on assumed knowledge of operator norms of A and B. We simulate the MPC scheme for a control horizon of length 15. In Figure 7.3ia and Figure 7.3ib we

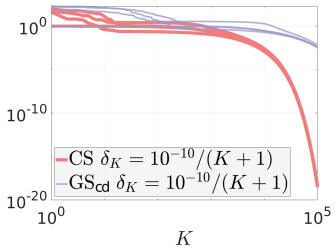
[6] Available at https://yalmip.github.io/example/standardmpc/



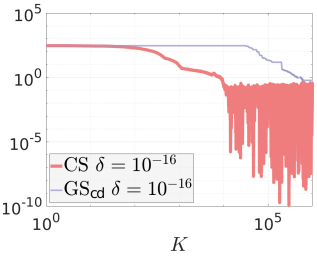
(ia) Suboptimality gap $f(\bar{u}_K) - \widehat{f}^\star$ for (7.6).



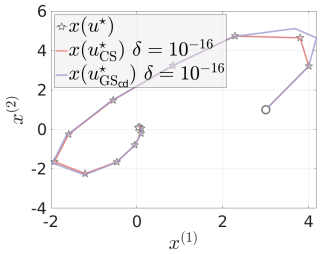
(ic) Optimal control policies for Example 7.5.



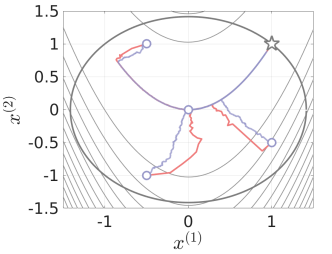
(iia) Suboptimality gap $f(x_K) - f^\star$ for (7.7).



(ib) Suboptimality gap $f(u_K) - \widehat{f}^\star$ for (7.6).



(id) Controlled trajectories resulting from the policies above.



(iib) Paths taken corresponding to Figure 7.3iia.

Figure 7.3: The single-point Complex-smoothing (CS) method applied to (i) a constrained problem (Example 7.5) using Algorithm 2 and to (ii) a non-convex problem (Example 7.6) using Theorem 6.1.

show the typical convergence for solving *one* instance of (7.6). It should be remarked that the oscillations as seen in Figure 7.3ib are due to the optimizer living on the boundary of the feasible set. Then, iteratively applying the MPC scheme, in Figure 7.3ic and Figure 7.3id we show that both the resulting control policy as well as the controlled trajectory induce by the complex-step method are close(r) to the optimal one.

7.3 Non-convex optimization We end with a classical example.

Example 7.6 (Rosenbrock function)**.** Consider optimizing a Rosenbrock function over a closed ball centred at 0, in particular, consider

$$\underset{x \in \sqrt{2}\mathbb{B}^2}{\text{minimize}} \quad (1 - x^{(1)})^2 + 100 \left((x^{(2)} - (x^{(1)})^2 \right)^2. \tag{7.7}$$

The minimizing argument of (7.7) is $x^\star = (1, 1)$, which happens to be the global optimizer over $\mathbb{R}^2$ as well. We will compare Algorithm 1 with the stepsize $\mu_k \equiv \mu = 1/(nL_1(f))$ from Theorem 6.1 against [NS17, Section 7]. Here we use oracle (7.1b) but remark that using oracle (7.1a) results in effectively the same performance on this problem. We start both algorithms from the same 4 initial conditions in $\sqrt{2}\mathbb{B}^2$ and show the convergence of the suboptimality gap in Figure 7.3iia and the paths taken by the algorithms in Figure 7.3iib. The complex-step method convergences significantly faster. We could speed-up this method slightly by further decreasing $\delta = 10^{-10}$, however, then the Gaussian smoothing method would break down.

Many more examples are possible, for example, as sketched in [CSV09, Section 1.2] one could consider tuning the regularization parameter in ridge regression. Comparing to for example the numerical experiment from [NG21] is less insightful as their smoothing parameter relies on the variance of the noise.

In conclusion, we see that the performance highlighted in Example 2.6 extrapolates to zeroth order optimization algorithms using those oracles. Here we remark that if one uses the averaged iterate $\bar{x}_{K-1} = (1/K)\sum_{k=0}^{K-1} x_k$, then, these numerical problems are less pronounced. This comes of course at the cost of slower convergence. Hence the complex-step method provides an attractive alternative if a provably fast and numerically stable algorithm is desired.

8 Future work

As highlighted in [AMH10], smoothness is sufficient but not necessary for the complex-step approximation to work. Moreover, the use of generic *dual* numbers — which are closely related to $i = \sqrt{-i}$ — as used in automatic differentiation might bring about new insights. It would also be interesting to extend the theory to the class of weakly-convex functions and to investigate the multi-batch and online settings.

A Appendix

Lemma A.1 ([SRB11, Lemma 1])**.** *Let $\{t_k\}_{k\geq 0}$ be a non-negative sequence satisfying the recursion*

$$t_K^2 \leq T_K + \sum_{k=0}^{K-m} \nu_k t_k \quad \forall K \geq 0,$$

where $m \in \{0, 1\}$ and $\{T_K\}_{K\geq 0}$ is a non-decreasing sequence with $T_0 \geq t_0^2$ and $\nu_k \geq 0$ $\forall k \geq 0$. Then,

$$t_K \leq \frac{1}{2}\sum_{k=0}^{K-m} \nu_k + \left(T_K + \left(\frac{1}{2}\sum_{k=0}^{K-m} \nu_k\right)^2\right)^{\frac{1}{2}} \quad \forall K \geq 0.$$

Proof. Up to a change in summation indices, the proof follows directly from [SRB11, Lemma 1]. □

Throughout a variety of the proofs we appeal to the following inequalities

$$\sum_{j=1}^{J} \frac{1}{j^2} \leq \frac{\pi^2}{6}, \quad \sum_{j=1}^{J} \frac{1}{j^4} \leq \frac{\pi^4}{90}, \quad \forall J \in \mathbb{N}, \tag{A.1}$$

which are truncations of the corresponding Riemann zeta functions.

Acknowledgements WJ and DK are supported by the Swiss National Science Foundation under the NCCR Automation, grant agreement 51NF40_180545. MCY is supported by the Hong Kong Research Grants Council under the grant 25302420. WJ likes to thank Prof. Nemirovski for the supplied reference material and Prof. Faulwasser for the introduction to the imaginary trick.

Bibliography

[Abr+18] R. Abreu et al. "On the accuracy of the Complex-Step-Finite-Difference method". *Journal of Computational and Applied Mathematics* 340 (2018), pp. 390 –403.

[ADX10] A. Agarwal, O. Dekel, and L. Xiao. "Optimal Algorithms for Online Convex Optimization with Multi-Point Bandit Feedback." *COLT*. 2010, pp. 28–40.

[AMH10] A. Al-Mohy and N. Higham. "The complex step approximation to the Fréchet derivative of a matrix function". *Numerical Algorithms* 53 (2010).

[AMR88] R Abraham, J. Marsden, and T Ratiu. *Manifolds, Tensor analysis, and Applications, Second Edition*. Applied Mathematical Sciences. Springer, 1988.

[APT20] A. Akhavan, M. Pontil, and A. B. Tsybakov. "Exploiting higher order smoothness in derivative-free optimization and continuous bandits" (2020). arXiv: 2006.07862.

[ASM15] R. Abreu, D. Stich, and J. Morales. "The Complex-Step-Finite-Difference method". *Geophysical Journal International* 202.1 (Apr. 2015), pp. 72–93.

[BBN19] A. S. Berahas, R. H. Byrd, and J. Nocedal. "Derivative-Free Optimization of Noisy Functions via Quasi-Newton Methods". *SIAM Journal on Optimization* 29.2 (2019), pp. 965–993.

[BCS19] A. S. Berahas, L. Cao, and K. Scheinberg. "Global convergence rate analysis of a generic line search algorithm with noise" (2019). arXiv: 1910.04055.

[BG18] K. Balasubramanian and S. Ghadimi. "Zeroth-order nonconvex stochastic optimization: Handling constraints, high-dimensionality and saddle-points" (2018). arXiv: 1809.06474.

[BM13] F. Bach and E. Moulines. "Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n)". *Advances in Neural Information Processing Systems*. Ed. by C. J. C. Burges et al. Vol. 26. Curran Associates, Inc., 2013, pp. 773–781.

[BP16] F. Bach and V. Perchet. "Highly-smooth zero-th order online optimization". *Conference on Learning Theory*. 2016, pp. 1–27.

[Cai+21] H. Cai et al. *A One-bit, Comparison-Based Gradient Estimator*. 2021. arXiv: 2010.02479.

[CH04] M. Cox and P. Harris. "Software Support for Metrology Best Practice Guide No. 11" (2004).

[CSV09] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics, 2009.

[d'A08] A. d'Aspremont. "Smooth optimization with approximate gradient". *SIAM Journal on Optimization* 19.3 (2008), pp. 1171–1183.

[DGN14] O. Devolder, F. Glineur, and Y. Nesterov. "First-order methods of smooth convex optimization with inexact oracle". *Mathematical Programming* 146.1 (2014), pp. 37–75.

[Duc+15] J. C. Duchi et al. "Optimal rates for zero-order convex optimization: The power of two function evaluations". *IEEE Transactions on Information Theory* 61.5 (2015), pp. 2788–2806.

[Faz+18] M. Fazel et al. "Global Convergence of Policy Gradient Methods for the Linear Quadratic Regulator". *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, 2018, pp. 1467–1476.

[FG16] K. Fountoulakis and J. Gondzio. "A second-order method for strongly convex ℓ_1-regularization problems". *Mathematical Programming* 156.1 (2016), pp. 189–219.

[FKM04] A. Flaxman, A. T. Kalai, and H. B. McMahan. "Online convex optimization in the bandit setting: gradient descent without a gradient". *CoRR* (2004).

[Gas+17] A. V. Gasnikov et al. "Stochastic online optimization. Single-point and multi-point non-linear multi-armed bandits. Convex and strongly-convex case". *Automation and remote control* 78.2 (2017), pp. 224–234.

[GL13] S. Ghadimi and G. Lan. "Stochastic first-and zeroth-order methods for nonconvex stochastic programming". *SIAM Journal on Optimization* 23.4 (2013), pp. 2341–2368.

[GM78] J. D. Gray and S. A. Morris. "When is a Function that Satisfies the Cauchy-Riemann Equations Analytic?" *The American Mathematical Monthly* 85.4 (1978), pp. 246–256.

[Gol+19] D. Golovin et al. "Gradientless descent: High-dimensional zeroth-order optimization" (2019). arXiv: 1911.06317.

[GWX17] Y. Gao, Y. Wu, and J. Xia. "Automatic differentiation based discrete adjoint method for aerodynamic design optimization on unstructured meshes". *Chinese Journal of Aeronautics* 30.2 (2017), pp. 611–627.

[HL14] E. Hazan and K. Y. Levy. "Bandit Convex Optimization: Towards Tight Bounds." *NIPS*. 2014, pp. 784–792.

[Ji+19] K. Ji et al. "Improved zeroth-order variance reduced algorithms and analysis for nonconvex optimization". *International Conference on Machine Learning*. PMLR. 2019, pp. 3100–3109.

[KP02] S. G. Krantz and H. R. Parks. *A Primer of Real Analytic Functions*. Birkhäuser, 2002.

[Kra00] S. G. Krantz. *Function Theory of Several Complex Variables*. AMS Chelsea Publishing, 2000.

[KW17] J. Korevaar and J. Wiegerinck. *Several complex variables*. 2017. URL: https://staff.science.uva.nl/j.j.o.o.wiegerinck/edu/scv/scvboek.pdf.

[LBM20] J. Li, K. Balasubramanian, and S. Ma. "Stochastic Zeroth-order Riemannian Derivative Estimation and Optimization" (2020). arXiv: 2003.11238.

[Lee13] J. M. Lee. *Introduction to Smooth Manifolds*. Graduate Texts in Mathematics. Springer, 2013.

[Lia+16] X. Lian et al. "A Comprehensive Linear Speedup Analysis for Asynchronous Stochastic Parallel Optimization from Zeroth-Order to First-Order". *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 2016, 3062–3070.

[Liu+20] S. Liu et al. "A Primer on Zeroth-Order Optimization in Signal Processing and Machine Learning: Principals, Recent Advances, and Applications". *IEEE Signal Processing Magazine* 37.5 (2020), pp. 43–54.

[LLZ21] H. Lam, H. Li, and X. Zhang. "Minimax efficient finite-difference stochastic gradient estimators using black-box function evaluations". *Operations Research Letters* 49.1 (2021), pp. 40–47.

[LM67] J. N. Lyness and C. B. Moler. "Numerical Differentiation of Analytic Functions". *SIAM Journal on Numerical Analysis* 4.2 (1967), pp. 202–210.

[LMW19] J. Larson, M. Menickelly, and S. M. Wild. "Derivative-free optimization methods". *arXiv preprint arXiv:1904.11585* (2019).

[Löf04] J. Löfberg. "YALMIP : A Toolbox for Modeling and Optimization in MATLAB". *In Proceedings of the CACSD Conference*. Taipei, Taiwan, 2004.

[LRD12] G. Lantoine, R. P. Russell, and T. Dargent. "Using multicomplex variables for automatic computation of high-order derivatives". *ACM Transactions on Mathematical Software* 38.3 (2012), pp. 1–21.

[Mal+19] D. Malik et al. "Derivative-free methods for policy optimization: Guarantees for linear quadratic systems". *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 2916–2925.

[MSA03] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso. "The Complex-Step Derivative Approximation". *ACM Trans. Math. Softw.* 29.3 (Sept. 2003), 245–262.

[Nes03] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer Science & Business Media, 2003.

[Nes11] Y. Nesterov. *Random gradient-free minimization of convex functions*. CORE Discussion Papers 2011001. 2011.

[NG21] V. Novitskii and A. Gasnikov. "Improved Exploiting Higher Order Smoothness in Derivative-free Optimization and Continuous Bandit" (2021). arXiv: 2101.03821.

[NS17] Y. Nesterov and V. Spokoiny. "Random gradient-free minimization of convex functions". *Foundations of Computational Mathematics* 17.2 (2017), pp. 527–566.

[NS18] F. Nikolovski and I. Stojkovska. "Complex-step derivative approximation in noisy environment". *Journal of Computational and Applied Mathematics* 327 (2018), pp. 64–78.

[NY83] A. S. Nemirovsky and D. B. Yudin. "Problem complexity and method efficiency in optimization." (1983).

[Ove01] M. L. Overton. *Numerical computing with IEEE floating point arithmetic*. SIAM, 2001.

[Pol86] J. Polderman. "A note on the structure of two subsets of the parameter space in adaptive control problems". *Systems & Control Letters* 7 (1986), pp. 25–34.

[PT90] B. T. Polyak and A. B. Tsybakov. "Optimal order of accuracy of search algorithms in stochastic optimization". *Problemy Peredachi Informatsii* 26.2 (1990), pp. 45–53.

[Rud87] W. Rudin. *Real and Complex Analysis*. Third edition. McGraw-Hill Education, 1987.

[Sha13] O. Shamir. "On the complexity of bandit and derivative-free stochastic convex optimization". *Conference on Learning Theory*. 2013, pp. 3–24.

[Sha17] O. Shamir. "An optimal algorithm for bandit and zero-order convex optimization with two-point feedback". *The Journal of Machine Learning Research* 18.1 (2017), pp. 1703–1713.

[SMG13] S. U. Stich, C. L. Muller, and B. Gartner. "Optimization of convex functions with random pursuit". *SIAM Journal on Optimization* 23.2 (2013), pp. 1284–1309.

[SRB11] M. Schmidt, N. Roux, and F. Bach. "Convergence rates of inexact proximal-gradient methods for convex optimization". *Advances in neural information processing systems* 24 (2011), pp. 1458–1466.

[ST98] W. Squire and G. Trapp. "Using Complex Variables to Estimate Derivatives of Real Functions". *SIAM Review* 40.1 (1998), pp. 110–112.

[SV15] Y. Singer and J. Vondrák. "Information-theoretic lower bounds for convex optimization with erroneous oracles". *Advances in Neural Information Processing Systems* 28 (2015), pp. 3204–3212.

[SW18] J. A. Snyman and D. N. Wilke. "Practical mathematical optimization: basic optimization theory and gradient-based algorithms" (2018).

[Vui+07] C. Vuik et al. *Numerical methods for ordinary differential equations*. VSSD, 2007.

[Zha+20] Y. Zhang et al. "Improving the Convergence Rate of One-Point Zeroth-Order Optimization using Residual Feedback" (2020). arXiv: 2006.10820.

[Zho18] X. Zhou. "On the Fenchel Duality between Strong Convexity and Lipschitz Continuous Gradient" (2018).