Small errors in random zeroth-order optimization are imaginary

Sunday Oct. 15.

SC38. Optimization Society's Award Session II.

Based on: "*Small errors in random zeroth-order optimization are imaginary*"

arXiv: https://arxiv.org/abs/2103.05478.

by **Wouter Jongeneel** (EPFL), Man-Chung Yue (HKU) and Daniel Kuhn (EPFL).

This work was supported by the (SNSF) *NCCR Automation*: https://nccr-automation.ch.

mail: wouter.jongeneel@epfl.ch,

web: wjongeneel.nl.

First-order optimization 101

For $f \in C^1(\mathcal{X} \subseteq \mathbb{R}^n; \mathbb{R})$, how to find

$$x^\star \in \text{argmin}_{x \in \mathcal{X}} f(x)\,?$$

[1] Popularity measure: in the last year (May 2022 - May 2023), searching for "SGD" was on average just a factor 1/25 as popular a searching for "Covid" (worldwide) https://trends.google.com/trends/explore?q=SGD,Covid.

[2] Still a very active research topic,
see https://www.quantamagazine.org/risky-giant-steps-can-solve-optimization-problems-faster-20230811/.

[3] Nesterov 2003, § 2.1.5.

First-order optimization 101

For $f \in C^1(\mathcal{X} \subseteq \mathbb{R}^n; \mathbb{R})$, how to find

$$x^\star \in \text{argmin}_{x \in \mathcal{X}} f(x)\,?$$

Common[1] approach: ***gradient*** *descent*

$$x_{k+1} = x_k - \mu_k \nabla f(x_k), \quad k = 1, 2, \ldots \tag{1}$$

Let f be convex with a L-Lipschitz **gradient**, *i.e.*,

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \quad \forall x, y \in \mathcal{X},$$

then, for[2] $\mu_k = 1/L$ and $x_1, x_2, \ldots, x_K$ generated by (1) one obtains[3]

$$f(x_K) - f(x^\star) \leq \mathcal{O}\left(\frac{L \cdot \|x_1 - x^\star\|_2^2}{K}\right).$$

[1] Popularity measure: in the last year (May 2022 - May 2023), searching for "SGD" was on average just a factor 1/25 as popular a searching for "Covid" (worldwide) `https://trends.google.com/trends/explore?q=SGD,Covid`.

[2] Still a very active research topic, see `https://www.quantamagazine.org/risky-giant-steps-can-solve-optimization-problems-faster-20230811/`.

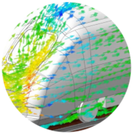
[3] Nesterov 2003, § 2.1.5.

If a gradient exists, does it mean we always *have* a gradient?

Example: DE constrained problems

Energy efficiency of transportation systems becomes increasingly important; must be optimized[4]. Good news: regularity is understood/studied.





Let $f(x)$ represent *aerodynamic performance* for x a set of *design parameters*, do we have an expression for $\nabla f(x)$?

[4] Images from: https://predatorcycling.com/, https://www.3ds.com/ and https://www.youtube.com/watch?v=FGmYpo-gkpU&ab_channel=EdwinLinders.

Example: DE constrained problems

Energy efficiency of transportation systems becomes increasingly important; must be optimized[4]. Good news: regularity is understood/studied.

Let $f(x)$ represent *aerodynamic performance* for x a set of *design parameters*, do we have an expression for $\nabla f(x)$?

○ Idea: we *can* **evaluate** $f(x')$ for some design choice x', *i.e.*, by simulation, and subsequently use $x'_1, x'_2, \ldots, f(x'_1), f(x'_2), \ldots,$ (**zeroth-order information**) for optimization.

[4] Images from: https://predatorcycling.com/, https://www.3ds.com/ and https://www.youtube.com/watch?v=FGmYpo-gkpU&ab_channel=EdwinLinders.

Zeroth-order optimization

Obtain (approximate)

$$x^\star \in \mathsf{argmin}_{x \in \mathcal{X}} f(x)$$

via function evaluations $f(x_1), f(x_2), \ldots, f(x_K)$ for some set of *selected* points $x_1, x_2, \ldots, x_K$. (For simplicity, we omit noise for now.)

[5] For references, consult the recent survey articles: Larson, Menickelly, and Wild 2019; Liu et al. 2020.

[6] See the books by Conn, Scheinberg, and Vicente 2009 and Audet and Hare 2017.

[7] Kiefer and Wolfowitz 1952; Nemirovsky and Yudin 1983; Flaxman, Kalai, and McMahan 2004; Spall 2005; Nesterov and Spokoiny 2017.

Zeroth-order optimization

Obtain (approximate)

$$x^\star \in \text{argmin}_{x \in \mathcal{X}} f(x)$$

via function evaluations $f(x_1), f(x_2), \ldots, f(x_K)$ for some set of *selected* points $x_1, x_2, \ldots, x_K$. (For simplicity, we omit noise for now.)

Two common paths[5]:

(i) *Approximate a model*: construct a local model of f, optimize using that model, *e.g.*, using a trust region method[6].

(ii) ***Approximate an algorithm***: *e.g.*, approximate ∇f directly and apply some form of gradient descent[7].

[5] For references, consult the recent survey articles: Larson, Menickelly, and Wild 2019; Liu et al. 2020.

[6] See the books by Conn, Scheinberg, and Vicente 2009 and Audet and Hare 2017.

[7] Kiefer and Wolfowitz 1952; Nemirovsky and Yudin 1983; Flaxman, Kalai, and McMahan 2004; Spall 2005; Nesterov and Spokoiny 2017.

A gradient-based approach

For any smooth $f : \mathbb{R} \to \mathbb{R}$

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

[8] d'Aspremont 2008; Devolder, Glineur, and Nesterov 2014.

A gradient-based approach

For any smooth $f : \mathbb{R} \to \mathbb{R}$

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

Then, run ***inexact*** ($\delta > 0$ fixed) gradient descent

$$x_{k+1} = x_k - \mu_k \frac{f(x_k+\delta) - f(x_k)}{\delta}.$$

◦ When does $f(x_k) \to f(x^\star)$?

[8] d'Aspremont 2008; Devolder, Glineur, and Nesterov 2014.

A gradient-based approach

For any smooth $f : \mathbb{R} \to \mathbb{R}$

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

Then, run ***inexact*** ($\delta > 0$ fixed) gradient descent

$$x_{k+1} = x_k - \mu_k \frac{f(x_k+\delta) - f(x_k)}{\delta}.$$

∘ When does $f(x_k) \to f(x^\star)$?

For fixed $\delta > 0$, a ***bias*** prevails, $f(x_k) \to f(x^\star) + \mathcal{O}(\delta)$[8], *e.g.*, for $f(x) = x^2$ we effectively compute the gradient of $f(x) + x\delta$, shifting $x^\star = 0$ to $-\frac{1}{2}\delta$.
Similarly, for $f \in C^1(\mathbb{R}^n;\mathbb{R})$, one should not naïvely use

$$\sum_{i=1}^{n} \frac{f(x+\delta e_i) - f(x)}{\delta} e_i \quad \text{for} \quad (e_1, e_2, \ldots, e_n) = I_n.$$

[8] d'Aspremont 2008; Devolder, Glineur, and Nesterov 2014.

A gradient-based approach cont.

(i) For appropriate (adaptive) $\delta > 0$, apply line-search[9] using

$$\sum_{i=1}^{n} \frac{f(x+\delta b_i) - f(x)}{\delta} b_i \approx \nabla f(x), \quad \text{for} \quad \det(b_1, b_2, \ldots, b_n) \neq 0.$$

[9] Berahas, Cao, and Scheinberg 2021.

[10] Randomization can be optimal Duchi et al. 2015, but no uniformly superior method exists yet "*randomized finite difference schemes can be implemented to be n times "cheaper" [than deterministic finite difference]; but an algorithm based on them has to take at least n times more steps.*" Scheinberg 2022, see also Berahas et al. 2022.

A gradient-based approach cont.

(i) For appropriate (adaptive) $\delta > 0$, apply line-search[9] using

$$\sum_{i=1}^{n} \frac{f(x+\delta b_i) - f(x)}{\delta} b_i \approx \nabla f(x), \quad \text{for} \quad \det(b_1, b_2, \ldots, b_n) \neq 0.$$

(ii) Suppose we find a *random variable* $\xi \in \mathbb{R}^n$ such that

$$\mathbb{E}_{\xi \sim \Xi}\left[\frac{f(x+\delta\xi) - f(x)}{\delta}\xi\right] \approx \nabla f(x).$$

Consider the **randomized** algorithm

$$x_{k+1} = x_k - \mu_k \frac{f(x_k+\delta\xi) - f(x_k)}{\delta}\xi, \quad \xi \sim \Xi.$$

[9] Berahas, Cao, and Scheinberg 2021.

[10] Randomization can be optimal Duchi et al. 2015, but no uniformly superior method exists yet "*randomized finite difference schemes can be implemented to be n times "cheaper" [than deterministic finite difference]; but an algorithm based on them has to take at least n times more steps.*" Scheinberg 2022, see also Berahas et al. 2022.

A gradient-based approach cont.

(i) For appropriate (adaptive) $\delta > 0$, apply line-search[9] using

$$\sum_{i=1}^{n} \frac{f(x+\delta b_i) - f(x)}{\delta} b_i \approx \nabla f(x), \quad \text{for} \quad \det(b_1, b_2, \ldots, b_n) \neq 0.$$

(ii) Suppose we find a *random variable* $\xi \in \mathbb{R}^n$ such that

$$\mathbb{E}_{\xi \sim \Xi}\left[\frac{f(x+\delta\xi) - f(x)}{\delta}\xi\right] \approx \nabla f(x).$$

Consider the **randomized** algorithm

$$x_{k+1} = x_k - \mu_k \frac{f(x_k+\delta\xi) - f(x_k)}{\delta}\xi, \quad \xi \sim \Xi.$$

(!) Active topic of research[10].

---

[9] Berahas, Cao, and Scheinberg 2021.

[10] Randomization can be optimal Duchi et al. 2015, but no uniformly superior method exists yet "*randomized finite difference schemes can be implemented to be n times "cheaper" [than deterministic finite difference]; but an algorithm based on them has to take at least n times more steps.*" Scheinberg 2022, see also Berahas et al. 2022.

Highly-influential exercise by Nemirovski and Yudin

Let $f : \mathbb{R}^n \to \mathbb{R}$, Nemirovski and Yudin[11] consider: δ-***smoothing***

$$f_{\delta}(x) = \mathbb{E}_{y \sim \mathbb{B}^n}\left[f(x + \delta y)\right] = \mathrm{vol}(\mathbb{B}^n)^{-1} \int_{\mathbb{B}^n} f(x + \delta y) \mathrm{d}y, \tag{2a}$$

$$\nabla f_{\delta}(x) = \frac{n}{\delta} \mathbb{E}_{y \sim \mathbb{S}^{n-1}}\left[f(x + \delta y) y\right] = \frac{n}{\delta} \int_{\mathbb{S}^{n-1}} f(x + \delta y) y \, \sigma(\mathrm{d}y). \tag{2b}$$

[11] Nemirovsky and Yudin 1983, credits usually given to Flaxman, Kalai, and McMahan 2004.

[12] Agarwal, Dekel, and Xiao 2010; Nesterov 2011.

Highly-influential exercise by Nemirovski and Yudin

Let $f : \mathbb{R}^n \to \mathbb{R}$, Nemirovski and Yudin[11] consider: δ-***smoothing***

$$f_\delta(x) = \mathbb{E}_{y\sim\mathbb{B}^n}\left[f(x+\delta y)\right] = \mathrm{vol}(\mathbb{B}^n)^{-1}\int_{\mathbb{B}^n} f(x+\delta y)\mathrm{d}y, \tag{2a}$$

$$\nabla f_\delta(x) = \frac{n}{\delta}\mathbb{E}_{y\sim\mathbb{S}^{n-1}}\left[f(x+\delta y)y\right] = \frac{n}{\delta}\int_{\mathbb{S}^{n-1}} f(x+\delta y)y\,\sigma(\mathrm{d}y). \tag{2b}$$

Natural ***single-point*** candidate to approximate ∂f:

$$g_\delta(x) = \frac{n}{\delta}f(x+\delta y)y, \quad y\sim\mathbb{S}^{n-1}. \tag{3a}$$

[11] Nemirovsky and Yudin 1983, credits usually given to Flaxman, Kalai, and McMahan 2004.

[12] Agarwal, Dekel, and Xiao 2010; Nesterov 2011.

Highly-influential exercise by Nemirovski and Yudin

Let $f : \mathbb{R}^n \to \mathbb{R}$, Nemirovski and Yudin[11] consider: δ-***smoothing***

$$f_\delta(x) = \mathbb{E}_{y\sim\mathbb{B}^n}\left[f(x+\delta y)\right] = \mathrm{vol}(\mathbb{B}^n)^{-1}\int_{\mathbb{B}^n} f(x+\delta y)\mathrm{d}y, \tag{2a}$$

$$\nabla f_\delta(x) = \frac{n}{\delta}\mathbb{E}_{y\sim\mathbb{S}^{n-1}}\left[f(x+\delta y)y\right] = \frac{n}{\delta}\int_{\mathbb{S}^{n-1}} f(x+\delta y)y\,\sigma(\mathrm{d}y). \tag{2b}$$

Natural ***single-point*** candidate to approximate ∂f:

$$g_\delta(x) = \frac{n}{\delta}f(x+\delta y)y, \quad y\sim\mathbb{S}^{n-1}. \tag{3a}$$

Observation[12]: avoid high-variance for $\delta \downarrow 0$ and give (3a) again the interpretation of a **directional derivative** and use a ***multi-point*** oracle like:

$$g'_\delta(x) = \frac{n}{\delta}\left(f(x+\delta y) - f(x)\right)y, \quad y\sim\mathbb{S}^{n-1}. \tag{3b}$$

---

[11] Nemirovsky and Yudin 1983, credits usually given to Flaxman, Kalai, and McMahan 2004.

[12] Agarwal, Dekel, and Xiao 2010; Nesterov 2011.

Early algorithmic analysis by Nesterov and Spokoiny

For $f : \mathbb{R}^n \to \mathbb{R}$ (locally convex), *Gaussian* smoothing[13]

$$f_{\gamma}(x) = \frac{1}{\kappa} \int_{\mathbb{R}^n} f(x + \gamma y) e^{-\frac{1}{2}\|y\|_2^2} \mathrm{d}y \tag{4a}$$

$$\nabla f_{\gamma}(x) = \frac{1}{\kappa} \int_{\mathbb{R}^n} \frac{f(x + \gamma y) - f(x - \gamma y)}{2\gamma} e^{-\frac{1}{2}\|y\|_2^2} y \mathrm{d}y \tag{4b}$$

with $\|\nabla f - \nabla f_{\gamma}\| = \mathcal{O}(n\gamma^2)$.

[13] Nesterov 2011; Nesterov and Spokoiny 2017.

Early algorithmic analysis by Nesterov and Spokoiny

For $f : \mathbb{R}^n \to \mathbb{R}$ (locally convex), *Gaussian* smoothing[13]

$$f_{\gamma}(x) = \frac{1}{\kappa} \int_{\mathbb{R}^n} f(x + \gamma y) e^{-\frac{1}{2}\|y\|_2^2} \mathrm{d}y \tag{4a}$$

$$\nabla f_{\gamma}(x) = \frac{1}{\kappa} \int_{\mathbb{R}^n} \frac{f(x + \gamma y) - f(x - \gamma y)}{2\gamma} e^{-\frac{1}{2}\|y\|_2^2} y \mathrm{d}y \tag{4b}$$

with $\|\nabla f - \nabla f_{\gamma}\| = \mathcal{O}(n\gamma^2)$.

$$\textbf{Oracle (cd):} \quad g_{\gamma}(x) = \frac{f(x + \gamma y) - f(x - \gamma y)}{2\gamma} y, \quad y \sim \mathcal{N}(0, I_n)$$

with $\mathbb{E}_{u \sim \mathcal{N}(0, I_n)} \left[\|g_{\gamma}(x)\|_2^2 \right] \leq \mathcal{O}(n^2\gamma^2 + n\|\nabla f(x)\|_2^2)$.

Algorithm: $x_{k+1} = x_k - \mu_k g_{\gamma_k}(x_k), \quad \mu_k = \mathcal{O}\left(1/nL\right).$

Performance: for $\gamma_k \to 0$ and $\bar{x}_K := 1/K \sum_{k=1}^{K} x_k$

$$\mathbb{E}[f(\bar{x}_K)] - f(x^\star) \leq \mathcal{O}\left(\frac{n \cdot L \cdot \|x_1 - x^\star\|_2^2}{K} \right) = \mathcal{O}(n) \cdot \text{ gradient descent}$$

[13] Nesterov 2011; Nesterov and Spokoiny 2017.

Numerical considerations

Analysis continued after 2011-2017, still, all common[14] oracles of the form

$$(\textit{finite difference}):\ \frac{f(x+\delta y)-f(x)}{\delta}y = \partial_x f(x) + \mathcal{O}(\delta)$$

$$(\textit{central difference}):\ \frac{f(x+\delta y)-f(x-\delta y)}{2\delta}y = \partial_x f(x) + \mathcal{O}(\delta^2),$$

$$\ldots = \partial_x f(x) + \mathcal{O}(\delta^{p\geq 1})$$

As such, many algorithms require $\delta_k \leq \mathcal{O}(1/k^q)$, with $q > 0$ for $k = 1, 2, \ldots$.

[14] Hazan and Levy 2014; Duchi et al. 2015; Nesterov and Spokoiny 2017; Gasnikov et al. 2017; Shamir 2017; Akhavan, Pontil, and Tsybakov 2020; Lam, Li, and Zhang 2021; Novitskii and Gasnikov 2021.

[15] Generally, $\mu_{\mathsf{M}} = 2^{-52} \approx 10^{-16}$.

Numerical considerations

Analysis continued after 2011-2017, still, all common[14] oracles of the form

$$(\textit{finite difference}): \frac{f(x+\delta y)-f(x)}{\delta}y = \partial_x f(x) + \mathcal{O}(\delta)$$

$$(\textit{central difference}): \frac{f(x+\delta y)-f(x-\delta y)}{2\delta}y = \partial_x f(x) + \mathcal{O}(\delta^2),$$

$$... = \partial_x f(x) + \mathcal{O}(\delta^{p\geq 1})$$

As such, many algorithms require $\delta_k \leq \mathcal{O}(1/k^q)$, with $q > 0$ for $k = 1, 2, \ldots$.

◦ However, can we ***practically*** select $\delta_k \to 0$ for $k \to +\infty$?

For sufficiently small δ, $f(x+\delta y) - f(x) \leq$ machine precision[15]
$\implies$ ***cancellation error***, *i.e.*, oracle output is nonsense.

◦ Not that frequently discussed, does it matter?

---

[14] Hazan and Levy 2014; Duchi et al. 2015; Nesterov and Spokoiny 2017; Gasnikov et al. 2017; Shamir 2017; Akhavan, Pontil, and Tsybakov 2020; Lam, Li, and Zhang 2021; Novitskii and Gasnikov 2021.

[15] Generally, $\mu_{\mathrm{M}} = 2^{-52} \approx 10^{-16}$.

Intermezzo: a beautiful insight from complex analysis

As pioneered in the 60s[16], let $f : \mathbb{R} \to \mathbb{R}$ be *real analytic* (C^ω) and consider

$$f(x+i\delta) = f(x) + \partial_x f(x) i\delta - \frac{1}{2}\partial_x^2 f(x)\delta^2 - \frac{1}{6}\partial_x^3 f(x) i\delta^3 + O(\delta^4), \quad i^2 = -1.$$

such that (for $z \in \mathbb{C}$, $z = \Re(z) + \Im(z)$):

$$\Im\left(f(x+i\delta)\right) = \partial_x f(x)\delta - \frac{1}{6}\partial_x^3 f(x)\delta^3 + O(\delta^5)$$

[16] Lyness and Moler 1967; Squire and Trapp 1998; Martins, Sturdza, and Alonso 2003; Abreu et al. 2018.

[17] A value of $\delta = 10^{-100}$ (!) is successfully used in National Physical Laboratory software Cox and Harris 2004, Page 44.

Intermezzo: a beautiful insight from complex analysis

As pioneered in the 60s[16], let $f : \mathbb{R} \to \mathbb{R}$ be *real analytic* (C^ω) and consider

$$f(x+i\delta) = f(x) + \partial_x f(x) i\delta - \frac{1}{2}\partial_x^2 f(x)\delta^2 - \frac{1}{6}\partial_x^3 f(x) i\delta^3 + O(\delta^4), \quad i^2 = -1.$$

such that (for $z \in \mathbb{C}$, $z = \Re(z) + \Im(z)$):

$$\Im\left(f(x+i\delta)\right) = \partial_x f(x)\delta - \frac{1}{6}\partial_x^3 f(x)\delta^3 + O(\delta^5)$$

and thus

$$\partial_x f(x) = \frac{\Im\Big(f(x+i\delta)\Big)}{\delta} + O(\delta^2), \quad f(x) = \Re(f(x+i\delta)) + O(\delta^2).$$

Hence, consider using

$$\frac{\Im\Big(f(x+i\delta)\Big)}{\delta} \approx \partial_x f(x).$$

Cancellation errors are impossible[17]. Again, does it matter?

[16] Lyness and Moler 1967; Squire and Trapp 1998; Martins, Sturdza, and Alonso 2003; Abreu et al. 2018.

[17] A value of $\delta = 10^{-100}$ (!) is successfully used in National Physical Laboratory software Cox and Harris 2004, Page 44.

Numerical considerations cont.: an example

For $f(x) = x^3$, approximate $\nabla f(x)$ at $x \in \{-1, 0, 10\}$ using

$$\text{(forward difference):} \quad f_{\mathsf{fd}}(x, \delta) = \frac{f(x+\delta) - f(x)}{\delta}, \tag{5a}$$

$$\text{(central difference):} \quad f_{\mathsf{cd}}(x, \delta) = \frac{f(x+\delta) - f(x-\delta)}{2\delta}, \tag{5b}$$

$$\text{(complex-step):} \quad f_{\mathsf{cs}}(x, \delta) = \frac{\Im\left(f(x+i\delta)\right)}{\delta} \tag{5c}$$

and compare the error for $\delta \downarrow 0$:

Numerical considerations cont.: an example

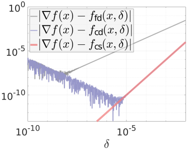
For $f(x) = x^3$, approximate $\nabla f(x)$ at $x \in \{-1, 0, 10\}$ using

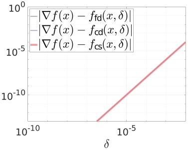
$$\text{(forward difference):} \quad f_{\mathsf{fd}}(x, \delta) = \frac{f(x+\delta) - f(x)}{\delta}, \tag{5a}$$

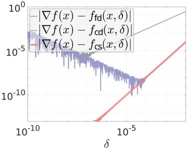
$$\text{(central difference):} \quad f_{\mathsf{cd}}(x, \delta) = \frac{f(x+\delta) - f(x-\delta)}{2\delta}, \tag{5b}$$

$$\text{(complex-step):} \quad f_{\mathsf{cs}}(x, \delta) = \frac{\Im\left(f(x + i\delta)\right)}{\delta} \tag{5c}$$

and compare the error for $\delta \downarrow 0$:







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

- Failures well before $\delta \approx \mu_{\mathsf{M}}$, so, it *does* matter.

On the necessity of leaving $\mathbb{R}$

Although single-point estimators exist[18], variance blows up for $\delta \downarrow 0$. Is this "*complex-lifting*" business really needed? Is there not a *real* analogue of

$$\partial_x f(x) = \frac{\Im(f(x + i\delta))}{\delta} + O(\delta^2)? \tag{6}$$

[18] Flaxman, Kalai, and McMahan 2004.

[19] Jongeneel 2021.

On the necessity of leaving $\mathbb{R}$

Although single-point estimators exist[18], variance blows up for $\delta \downarrow 0$. Is this "*complex-lifting*" business really needed? Is there not a *real* analogue of

$$\partial_x f(x) = \frac{\Im(f(x+i\delta))}{\delta} + O(\delta^2)? \tag{6}$$

Partial answer[19]: no.
Consider some non-empty open, convex set $\mathcal{D} \subseteq \mathbb{R}^n$ then, there does not exist a continuous map $G: \mathbb{R} \to \mathbb{R}$ such that for all real-analytic functions $f: \mathcal{D} \to \mathbb{R}$

$$G\left(f(x+\delta y)\right) = \langle \nabla f(x), y \rangle \delta + o(\delta) \quad \forall x \in \mathcal{D},\ \delta > 0,\ y \in \mathbb{S}^{n-1}. \tag{7}$$

◦ not surprising, but provides motivation.

[18] Flaxman, Kalai, and McMahan 2004.
[19] Jongeneel 2021.

Comment on Algorithmic Differentiation (AD)

- Why bother with approximations?

[20] The `Deep Learning Toolbox` in MATLAB and AD tools in Julia (See Bezanson et al. 2017; Revels, Lubin, and Papamarkou 2016; Innes 2018; Moses and Churavy 2020), *e.g.*, `ForwardDiff.jl`, `Zygote.jl` and `Enzyme.jl` or in Python, *e.g.*, JAX Bradbury et al. 2018

Comment on Algorithmic Differentiation (AD)

◦ Why bother with approximations?

Dual numbers: $a + b\varepsilon$ with $a, b \in \mathbb{R}$ and $\varepsilon \neq 0$, yet, $\varepsilon^2 = 0$, *i.e.*, elements of the quotient *ring* $\mathbb{R}[\varepsilon]/\varepsilon^2$, not a field $\implies$, *e.g.*, $\varepsilon^2/\varepsilon$ and $\sqrt{\varepsilon^2}$ not defined.
◦ $\mathbb{C}$ is an algebraically closed field.

AD: for $f : \mathbb{R} \to \mathbb{R}$ is sufficiently regular, *e.g.*, $f \in C^\omega(\mathbb{R})$, then,
$f(x + \varepsilon) = f(x) + \partial_x f(x)\varepsilon$, *i.e.*, $f(x + \varepsilon)$ provides us with the pair $(f(x), \partial_x f(x))$.

[20] The `Deep Learning Toolbox` in MATLAB and AD tools in Julia (See Bezanson et al. 2017; Revels, Lubin, and Papamarkou 2016; Innes 2018; Moses and Churavy 2020), *e.g.*, `ForwardDiff.jl`, `Zygote.jl` and `Enzyme.jl` or in Python, *e.g.*, JAX Bradbury et al. 2018

Comment on Algorithmic Differentiation (AD)

∘ Why bother with approximations?

Dual numbers: $a + b\varepsilon$ with $a, b \in \mathbb{R}$ and $\varepsilon \neq 0$, yet, $\varepsilon^2 = 0$, *i.e.*, elements of the quotient *ring* $\mathbb{R}[\varepsilon]/\varepsilon^2$, not a field $\implies$, *e.g.*, $\varepsilon^2/\varepsilon$ and $\sqrt{\varepsilon^2}$ not defined.
∘ $\mathbb{C}$ is an algebraically closed field.

AD: for $f : \mathbb{R} \to \mathbb{R}$ is sufficiently regular, *e.g.*, $f \in C^\omega(\mathbb{R})$, then,
$f(x + \varepsilon) = f(x) + \partial_x f(x)\varepsilon$, *i.e.*, $f(x + \varepsilon)$ provides us with the pair $(f(x), \partial_x f(x))$.

Consider $\partial_x f(x)|_{x=0}$ for the following C^ω functions:

$$f(x) = x/x, \quad f(x) = -\sin(x)/x, \quad f(x) = -e^{-\sqrt{x^2}^2}.$$

No free lunch: most populair AD tools[20] evaluate to `NaN` whereas the complex-step derivative correctly approximates $\partial_x f(x)|_{x=0} = 0$.
Theoretical solution: *Levi-Civita field* $\sum_{q \in \mathbb{Q}} a_q \varepsilon^q$ with $a_q \in \mathbb{R}\ \forall q \in \mathbb{Q}$ (inf. dim).

[20] The `Deep Learning Toolbox` in MATLAB and AD tools in Julia (See Bezanson et al. 2017; Revels, Lubin, and Papamarkou 2016; Innes 2018; Moses and Churavy 2020), *e.g.*, `ForwardDiff.jl`, `Zygote.jl` and `Enzyme.jl` or in Python, *e.g.*, JAX Bradbury et al. 2018

A solution: the complex-step oracle[22] (exercise)

Let $f \in C^\omega(\mathcal{X} \subseteq \mathbb{R}^n; \mathbb{R})$, using *Cauchy-Riemann/Stokes* show that:

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

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

with $\|\nabla f_\delta - \nabla f\|_2 \leq \mathcal{O}(n\delta^2)$.

[21] The paper provides similar results for strong-convex and non-convex functions. This approach recently surfaced in the optimization community Nikolovski and Stojkovska 2018; Hare and Srivastava 2023 with the first complete deterministic non-asymptotic analysis appearing in Jongeneel, Yue, and Kuhn 2021. The first applications of the complex-step derivative to Reinforcement Learning appeared in Wang and Spall 2021; Wang, Zhu, and Spall 2021.

[22] Jongeneel, Yue, and Kuhn 2021.

A solution: the complex-step oracle[22] (exercise)

Let $f \in C^\omega(\mathcal{X} \subseteq \mathbb{R}^n; \mathbb{R})$, using *Cauchy-Riemann/Stokes* show that:

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

with $\|\nabla f_\delta - \nabla f\|_2 \leq \mathcal{O}(n\delta^2)$.

Oracle (cs): $g_\delta(x) = \frac{n}{\delta}\Im\left(f(x + i\delta y)\right) y, \quad y \sim \mathbb{S}^{n-1}$.

with $\mathbb{E}_{u \sim \mathbb{S}^{n-1}}\left[\|g_\delta(x)\|_2^2\right] \leq \mathcal{O}(n^2\delta^4 + n^2\delta^2\|\nabla f(x)\|_2 + n\|\nabla f(x)\|_2^2)$.

Algorithm: $x_{k+1} = x_k - \mu_k g_{\delta_k}(x_k), \quad \mu_k = \mathcal{O}\left(1/nL\right)$

Performance: for f *convex* $\delta_k = \mathcal{O}(1/k)$ and $\bar{x}_K := 1/K \sum_{k=1}^K x_k$

$$\mathbb{E}[f(\bar{x}_K)] - f(x^\star) \leq \mathcal{O}\left(\frac{n \cdot L \cdot \|x_1 - x^\star\|_2^2}{K}\right) = \mathcal{O}(n) \cdot \text{ gradient descent}^{21}.$$

[21] The paper provides similar results for strong-convex and non-convex functions. This approach recently surfaced in the optimization community Nikolovski and Stojkovska 2018; Hare and Srivastava 2023 with the first complete deterministic non-asymptotic analysis appearing in Jongeneel, Yue, and Kuhn 2021. The first applications of the complex-step derivative to Reinforcement Learning appeared in Wang and Spall 2021; Wang, Zhu, and Spall 2021.

[22] Jongeneel, Yue, and Kuhn 2021.

Solution:

∘ $f \in C^\omega$ convex $\not\Longrightarrow$ f_δ convex, *e.g.*,
for $f(x) = x^4$ we have $\Re(f(x + i\delta y)) = x^4 - 6x^2(\delta y)^2 + (\delta y)^4$.
Hence, look *beyond* typical convex proofs[23].

[23] We frequently appeal to Schmidt, Roux, and Bach 2011, Lem. 1.

[24] Although the general form is largely due to Cartan (Élie).

[25] C^ω is sufficient, but not necessary.

Solution:

∘ $f \in C^\omega$ convex $\not\Longrightarrow$ f_δ convex, *e.g.*,
for $f(x) = x^4$ we have $\Re(f(x+i\delta y)) = x^4 - 6x^2(\delta y)^2 + (\delta y)^4$.
Hence, look *beyond* typical convex proofs[23].

∘ **Cauchy, Riemann and Stokes**[24] meet:
for[25] $f \in C^\omega(\mathbb{R}^n;\mathbb{R})$ with $f(x+iy) = u(x,y) + iv(x,y)$, then
$\partial_{x_i} u = \partial_{y_i} v$, $\partial_{y_i} u = -\partial_{x_i} v$ $\forall i \in [n]$ (CR)
and for Ω orientable we have that $\int_\Omega \mathrm{d}\omega = \int_{\partial\Omega} \omega$ (Stokes), implication: *the divergence theorem* $\int_\Omega \mathrm{div}(X)\mathrm{dvol}_\Omega = \int_{\partial\Omega} \langle X, N\rangle \mathrm{dvol}_{\partial\Omega}$. Hence:

$$
\begin{aligned}
\nabla f_\delta(x) &\overset{(\text{def.,DCT})}{=} \mathrm{vol}(\mathbb{B}^n)^{-1} \int_{\mathbb{B}^n} \nabla_x \Re\left(f(x+i\delta y)\right) \mathrm{d}y \\
&\overset{(\text{CR})}{=} (\mathrm{vol}(\mathbb{B}^n)\delta)^{-1} \int_{\mathbb{B}^n} \nabla_y \Im\left(f(x+i\delta y)\right) \mathrm{d}y \\
&\overset{(\text{Stokes})}{=} \mathrm{vol}(\mathbb{S}^{n-1})/(\mathrm{vol}(\mathbb{B}^n)\delta) \int_{\mathbb{S}^{n-1}} \Im\left(f(x+i\delta y)\right) y\, \sigma(\mathrm{d}y) \\
&\overset{(\mathrm{vol}(\mathbb{S}^{n-1})/(\mathrm{vol}(\mathbb{B}^n))=n)}{=} (n/\delta) \cdot \mathbb{E}_{y\sim\sigma}\left[\Im\left(f(x+i\delta y)\right) y\right].
\end{aligned}
$$

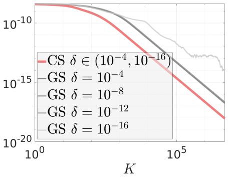
[23] We frequently appeal to Schmidt, Roux, and Bach 2011, Lem. 1.
[24] Although the general form is largely due to Cartan (Élie).
[25] C^ω is sufficient, but not necessary.

Example: worst function in the world

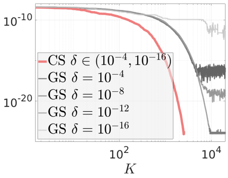
Consider the test function from Nesterov 2003, § 2.1.2

$$f_n(x) = L\left(\tfrac{1}{2}\left[(x^{(1)})^2 + \sum_{i=1}^{n-1}(x^{(i+1)} - x^{(i)})^2 + (x^{(n)})^2\right] - x^{(1)}\right) \qquad (9)$$

for $x_1 = 0$, $L = 10^{-8}$, $L_1(f) = 4L$ and $(x^\star)^{(i)} = 1 - i/(n+1)$ with $x^{(i)}$.



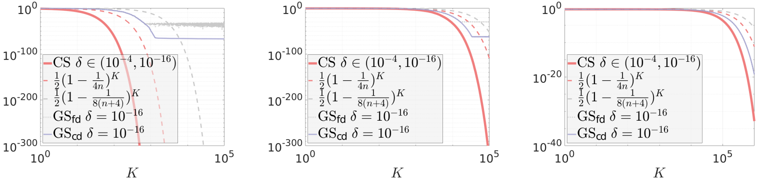
(ia) Suboptimality gap $f(\bar{x}_K) - f^\star$ for the test function (9).



(ib) Suboptimality gap $f(x_K) - f^\star$ for the test function (9).

Figure: The single-point Complex-smoothing (CS) compared to the multi-point Gaussian smoothing (GS) (fd) method from Nesterov and Spokoiny 2017.

Example: strong convexity $f(x) = \frac{1}{2}\|x\|_2^2$



(a) $f(x_K) - f^\star$, $n = 10^0$.　(b) $f(x_K) - f^\star$, $n = 10^2$.　(c) $f(x_K) - f^\star$, $n = 10^4$.

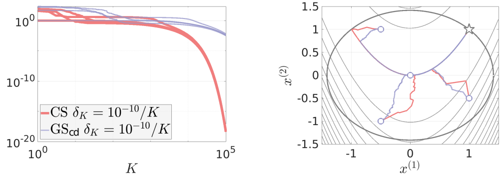
Figure: The single-point Complex-smoothing (CS) compared to the multi-point Gaussian smoothing (GS) (fd and cd) method from Nesterov and Spokoiny 2017, Eq. (55). The rate is for (GS) (cd).

Example: non-convex optimization

Consider a *Rosenbrock* optimization problem

$$\underset{x \in \sqrt{2}\mathbb{B}^2}{\text{minimize}} \quad (1 - x^{(1)})^2 + 100\left((x^{(2)} - (x^{(1)})^2\right)^2. \tag{10}$$

with $x^\star = (1, 1)$.



(a) Suboptimality gap $f(x_K) - f^\star$ for (10).

(b) Paths taken corresponding to Figure 4a.

Figure: The single-point Complex-smoothing (CS) method versus Gaussian-smoothing Nesterov and Spokoiny 2017.

What about noise?

We can handle[26] "*simulation noise*", *e.g.*, $\Im(f(z)) + \xi$, $z \in \Omega \in \mathbb{C}^n$, $\xi \sim (0, \sigma^2)$.

$$\textbf{Oracle (cs, noisy):} \quad g_\delta(x) = \frac{n}{\delta}\Im\left(f(x+i\delta y)\right)y + \frac{n}{\delta}\xi y, \quad y \sim \mathbb{S}^{n-1}. \tag{11}$$

- Handling $(n\xi/\delta)$ non-trivial. In general, we need $\mu_k = \mathcal{O}(1/k)$ and $\delta_k = o(\mu_k)$.

[26] Jongeneel 2021.

[27] Building upon Hazan, Rakhlin, and Bartlett 2008; Akhavan, Pontil, and Tsybakov 2020.

[28] Shown by building upon Shamir 2013.

What about noise?

We can handle[26] "*simulation noise*", *e.g.*, $\Im(f(z)) + \xi$, $z \in \Omega \in \mathbb{C}^n$, $\xi \sim (0, \sigma^2)$.

$$\textbf{Oracle (cs, noisy):} \quad g_\delta(x) = \frac{n}{\delta}\Im\left(f(x+i\delta y)\right)y + \frac{n}{\delta}\xi y, \quad y \sim \mathbb{S}^{n-1}. \tag{11}$$

◦ Handling $(n\xi/\delta)$ non-trivial. In general, we need $\mu_k = \mathcal{O}(1/k)$ and $\delta_k = o(\mu_k)$.

◦ Non-asymptotic results[27] for: constrained/unconstrained strongly convex functions and some non-convex functions (locally).

◦ The algorithm is ***rate-optimal*** in the quadratic setting[28] (not surprising).

---

26 Jongeneel 2021.

27 Building upon Hazan, Rakhlin, and Bartlett 2008; Akhavan, Pontil, and Tsybakov 2020.

28 Shown by building upon Shamir 2013.

What about noise?

We can handle[26] "*simulation noise*", *e.g.*, $\Im(f(z)) + \xi$, $z \in \Omega \in \mathbb{C}^n$, $\xi \sim (0, \sigma^2)$.

$$\textbf{Oracle (cs, noisy):} \quad g_\delta(x) = \frac{n}{\delta}\Im\left(f(x + i\delta y)\right) y + \frac{n}{\delta}\xi y, \quad y \sim \mathbb{S}^{n-1}. \tag{11}$$

◦ Handling $(n\xi/\delta)$ non-trivial. In general, we need $\mu_k = \mathcal{O}(1/k)$ and $\delta_k = o(\mu_k)$.

◦ Non-asymptotic results[27] for: constrained/unconstrained strongly convex functions and some non-convex functions (locally).
◦ The algorithm is ***rate-optimal*** in the quadratic setting[28] (not surprising).

Why the ball $\mathbb{B}^n$ and not some other geometry
$\mathsf{M} \in \mathscr{M} = \{\mathsf{M} \subset [-1, 1]^n : \mathsf{M} \text{ diffeomorphic to } \mathbb{B}^n\}$? Optimal in the sense that

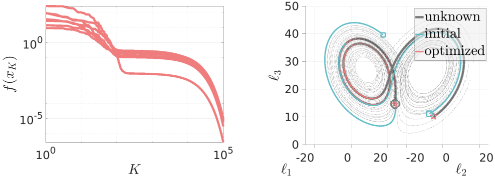
$$\text{min}_{\mathsf{M}\in\mathscr{M}} \frac{\text{vol}(\delta\partial\mathsf{M})}{\text{vol}(\delta\mathsf{M})} = \frac{n}{\delta}, \quad \mathbb{B}^n = \text{argmin}_{\mathsf{M}\in\mathscr{M}} \frac{\text{vol}(\delta\partial\mathsf{M})}{\text{vol}(\delta\mathsf{M})}, \tag{12}$$

which follows from the *isoperimetric inequality in* $\mathbb{R}^n$ Osserman 1978.

[26] Jongeneel 2021.
[27] Building upon Hazan, Rakhlin, and Bartlett 2008; Akhavan, Pontil, and Tsybakov 2020.
[28] Shown by building upon Shamir 2013.

Example: non-convex optimization (outlook)

Regularity of ODE/PDE constrained optimization problems can often be understood. We apply our zeroth-order algorithm to a ODE problem[29].



(a) Decay of $f(x_K)$ with the total number K of iterations for 10 independent simulation runs.

(b) Trajectories starting from $\ell(0)$ (unknown), x_0 (initial) and x_K for $K = 10^5$ (optimized).

Figure: Estimating the initial state $\ell(0)$ of a Lorenz system from a noisy measurement p of the state $\ell(2) = \varphi^2(\ell(0))$ (grey circle in 5b) at time 2. Even though the initial estimate x_0 is close to the optimized estimate x_K, $\varphi^2(x_0)$ is far from $\varphi^2(\ell(0))$.

[29] The complex-step derivative is implemented in an airfoil optimization package. Their underlying algorithm relies on *sequential quadratic programming* Nocedal and Wright 2006, Ch. 18, as such, the guarantees one can provide are different, see `https://mdolab-cmplxfoil.readthedocs-hosted.com/en/latest/index.html`, our work aims at providing rigorous guarantees with respect to the optimization algorithm itself.

The end

Main take away: single-point estimator where $\delta_k = \mathcal{O}(1/k)$ can be safely implemented.

Many open problems remain.

contact: `wjongeneel.nl` (slides will appear there).

Key references:

(a) Arkadi Semenovich Nemirovsky and David Borisovich Yudin (1983). *Problem complexity and method efficiency in optimization*. Wiley-Interscience

(b) Boris Teodorovich Polyak and Aleksandr Borisovich Tsybakov (1990). "Optimal order of accuracy of search algorithms in stochastic optimization". In: *Problemy Peredachi Informatsii* 26.2, pp. 45–53

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

(d) Francis Bach and Vianney Perchet (2016). "Highly-smooth zero-th order online optimization". In: *Conference on Learning Theory*, pp. 1–27

(e) Yurii Nesterov and Vladimir Spokoiny (2017). "Random gradient-free minimization of convex functions". In: *Foundations of Computational Mathematics* 17.2, pp. 527–566

(f) Albert S Berahas et al. (2022). "A theoretical and empirical comparison of gradient approximations in derivative-free optimization". In: *Foundations of Computational Mathematics* 22.2, pp. 507–560

(g) **Wouter Jongeneel, Man-Chung Yue, and Daniel Kuhn (2021). "Small Errors in Random Zeroth Order Optimization are Imaginary". In: arXiv:** 2103.05478

Appendix.

References:

Abreu, Rafael et al. (2018). "On the accuracy of the Complex-Step-Finite-Difference method". In: *Journal of Computational and Applied Mathematics* 340, pp. 390 –403.

Agarwal, Alekh, Ofer Dekel, and Lin Xiao (2010). "Optimal Algorithms for Online Convex Optimization with Multi-Point Bandit Feedback.". In: *COLT*, pp. 28–40.

Akhavan, Arya, Massimiliano Pontil, and Alexandre B Tsybakov (2020). "Exploiting higher order smoothness in derivative-free optimization and continuous bandits". In: *arXiv preprint arXiv:2006.07862*.

Audet, Charles and Warren Hare (2017). *Derivative-free and Blackbox Optimization*. Springer.

Bach, Francis and Vianney Perchet (2016). "Highly-smooth zero-th order online optimization". In: *Conference on Learning Theory*, pp. 1–27.

Berahas, Albert S, Liyuan Cao, and Katya Scheinberg (2021). "Global convergence rate analysis of a generic line search algorithm with noise". In: *SIAM Journal on Optimization* 31.2, pp. 1489–1518.

Berahas, Albert S et al. (2022). "A theoretical and empirical comparison of gradient approximations in derivative-free optimization". In: *Foundations of Computational Mathematics* 22.2, pp. 507–560.

Bezanson, Jeff et al. (2017). "Julia: A Fresh Approach to Numerical Computing". In: *SIAM Review* 59.1, pp. 65–98. DOI: 10.1137/141000671.

Bradbury, James et al. (2018). *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. URL: http://github.com/google/jax.

Conn, Andrew R., Katya Scheinberg, and Luis N. Vicente (2009). *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics.

Cox, MG and PM Harris (2004). "Software Support for Metrology Best Practice Guide No. 11". In.

d'Aspremont, Alexandre (2008). "Smooth optimization with approximate gradient". In: *SIAM Journal on Optimization* 19.3, pp. 1171–1183.

Devolder, Olivier, François Glineur, and Yurii Nesterov (2014). "First-order methods of smooth convex optimization with inexact oracle". In: *Mathematical Programming* 146.1, pp. 37–75.

Duchi, John C et al. (2015). “Optimal rates for zero-order convex optimization: The power of two function evaluations”. In: *IEEE Transactions on Information Theory* 61.5, pp. 2788–2806.

Flaxman, Abraham, Adam Tauman Kalai, and H. Brendan McMahan (2004). “Online convex optimization in the bandit setting: gradient descent without a gradient”. In: *CoRR*.

Gasnikov, Alexander V et al. (2017). “Stochastic online optimization. Single-point and multi-point non-linear multi-armed bandits. Convex and strongly-convex case”. In: *Automation and remote control* 78.2, pp. 224–234.

Hare, Warren and Kashvi Srivastava (2023). “A numerical study of applying complex-step gradient and Hessian approximations in blackbox optimization”. In: *Pacific Journal of Optimization* 19.3, pp. 391–410.

Hazan, Elad and Kfir Y Levy (2014). “Bandit Convex Optimization: Towards Tight Bounds.”. In: *NIPS*, pp. 784–792.

Hazan, Elad, Alexander Rakhlin, and Peter Bartlett (2008). “Adaptive Online Gradient Descent”. In: *Advances in Neural Information Processing Systems*, pp. 65–72.

Innes, Michael (2018). “Don’t unroll adjoint: Differentiating SSA-form programs”. In: *arXiv preprint arXiv:1810.07951*.

Jongeneel, Wouter (2021). “Imaginary Zeroth-Order Optimization”. In: arXiv: 2112.07488.

Jongeneel, Wouter, Man-Chung Yue, and Daniel Kuhn (2021). “Small Errors in Random Zeroth Order Optimization are Imaginary”. In: arXiv: 2103.05478.

Kiefer, Jack and Jacob Wolfowitz (1952). “Stochastic estimation of the maximum of a regression function”. In: *The Annals of Mathematical Statistics*, pp. 462–466.

Lam, Henry, Haidong Li, and Xuhui Zhang (2021). “Minimax efficient finite-difference stochastic gradient estimators using black-box function evaluations”. In: *Operations Research Letters* 49.1, pp. 40–47.

Larson, Jeffrey, Matt Menickelly, and Stefan M Wild (2019). “Derivative-free optimization methods”. In: *arXiv preprint arXiv:1904.11585*.

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

Lyness, J. N. and C. B. Moler (1967). “Numerical Differentiation of Analytic Functions”. In: *SIAM Journal on Numerical Analysis* 4.2, pp. 202–210.

Martins, Joaquim R. R. A., Peter Sturdza, and Juan J. Alonso (Sept. 2003). “The Complex-Step Derivative Approximation”. In: *ACM Trans. Math. Softw.* 29.3, 245–262.

Moses, William and Valentin Churavy (2020). “Instead of Rewriting Foreign Code for Machine Learning, Automatically Synthesize Fast Gradients”. In: *Advances in Neural Information Processing Systems*. Vol. 33, pp. 12472–12485.

Nemirovsky, Arkadi Semenovich and David Borisovich Yudin (1983). *Problem complexity and method efficiency in optimization*. Wiley-Interscience.

Nesterov, Yurii (2003). *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer Science & Business Media.

— (2011). *Random gradient-free minimization of convex functions*. CORE Discussion Papers 2011001.

Nesterov, Yurii and Vladimir Spokoiny (2017). “Random gradient-free minimization of convex functions”. In: *Foundations of Computational Mathematics* 17.2, pp. 527–566.

- Nikolovski, Filip and Irena Stojkovska (2018). "Complex-step derivative approximation in noisy environment". In: *Journal of Computational and Applied Mathematics* 327, pp. 64–78.
- Nocedal, Jorge and Stephen J. Wright (2006). *Numerical Optimization*. Springer Series in Operations Research. Springer.
- Novitskii, Vasilii and Alexander Gasnikov (2021). "Improved Exploiting Higher Order Smoothness in Derivative-free Optimization and Continuous Bandit". In: *arXiv preprint arXiv:2101.03821*.
- Osserman, Robert (1978). "The isoperimetric inequality". In: *Bulletin of the American Mathematical Society* 84.6, pp. 1182–1238.
- Polyak, Boris Teodorovich and Aleksandr Borisovich Tsybakov (1990). "Optimal order of accuracy of search algorithms in stochastic optimization". In: *Problemy Peredachi Informatsii* 26.2, pp. 45–53.
- Revels, Jarrett, Miles Lubin, and Theodore Papamarkou (2016). "Forward-mode automatic differentiation in Julia". In: *arXiv preprint arXiv:1607.07892*.
- Scheinberg, Katya (2022). "Finite Difference Gradient Approximation: To Randomize or Not?" In: *INFORMS Journal on Computing* 34.5, pp. 2384–2388.

- Schmidt, Mark, Nicolas Roux, and Francis Bach (2011). "Convergence rates of inexact proximal-gradient methods for convex optimization". In: *Neural Information Processing Systems*, pp. 1458–1466.
- Shamir, Ohad (2013). "On the complexity of bandit and derivative-free stochastic convex optimization". In: *Conference on Learning Theory*, pp. 3–24.
- — (2017). "An optimal algorithm for bandit and zero-order convex optimization with two-point feedback". In: *The Journal of Machine Learning Research* 18.1, pp. 1703–1713.
- Spall, James C (2005). *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. John Wiley & Sons.
- Squire, William and George Trapp (1998). "Using Complex Variables to Estimate Derivatives of Real Functions". In: *SIAM Review* 40.1, pp. 110–112.
- Wang, Long and James C Spall (2021). "Improved SPSA using complex variables with applications in optimal control problems". In: *American Control Conference*. IEEE, pp. 3519–3524.
- Wang, Long, Jingyi Zhu, and James C Spall (2021). "Model-free optimal control using SPSA with complex variables". In: *Annual Conference on Information Sciences and Systems*. IEEE, pp. 1–5.