THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 1 Capitole

Présentée et soutenue par

Tam LE

Le 11 octobre 2023

Calcul non-lisse et optimisation pour l'apprentissage automatique: échantillonnage au premier ordre et différentiation implicite

Ecole doctorale : **EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse**

Spécialité : **Mathématiques et Applications**

Unité de recherche :
TSE-R - Toulouse School of Economics - Recherche

Thèse dirigée par
Jérome BOLTE et Edouard PAUWELS

Jury

M. Jérôme MALICK, Rapporteur
Mme Gersende FORT, Examinatrice
M. Pascal BIANCHI, Examinateur
M. Eric MOULINES, Examinateur
M. Jérome BOLTE, Directeur de thèse
M. Edouard PAUWELS, Co-directeur de thèse

Nonsmooth calculus and optimization for machine learning: first-order sampling and implicit differentiation

Tam LE

Remerciements

J'aimerais remercier très chaleureusement mes deux directeurs de thèse pour leur accompagnement de qualité pendant ces trois années. Merci de m'avoir proposé un sujet aussi riche et captivant ! Jérôme, ta vision éclairée de la recherche, et ton attrait pour les idées simples et élégantes m'ont beaucoup plu. Ta gentillesse et ta bienveillance ont été très encourageantes et je garde de très bons souvenirs de nos échanges, qu'ils aient été d'ordre scientifique ou non. Edouard, je me souviens de ces six premiers mois de stage qui ont précédés la thèse et durant lesquels, à l'autre bout du fil, j'ai été surpris par ta grande disponibilité, ta patience et ta pédagogie. Ces dernières, mais aussi ton savoir-faire et ton enthousiasme ont été sans nul doute des moteurs essentiels à la réussite de cette thèse. J'ai vraiment été heureux pour toi de te voir être recruté en tant que Professeur à TSE, mon seul regret étant de tourner la page en même temps que toi. Avec vous deux, j'ai été marqué par l'atmosphère amicale et bienveillante qui entourait nos interactions. J'avoue qu'évoluer à vos côtés paraissait de temps à autre intimidant. Malgré tout, vous avez toujours cherché à me guider vers l'indépendance en essayant d'établir une relation de chercheurs à chercheur, et je pense que cela m'a beaucoup grandi. Merci également de m'avoir incité à participer à de nombreuses conférences scientifiques où j'ai fait de belles rencontres.

J'aimerais remercier tout particulièrement Antonio (Tony) Silveti-Falls, qui est également contributeur de cette thèse. Quand tu es arrivé pour ton post-doc afin de travailler sur la différentiation implicite, tu as été un excellent modèle pour moi dans la recherche et tout particulièrement dans ma manière de présenter. Merci de m'avoir emmené dans ce fabuleux restaurant à l'issue de ma soutenance !

Un grand merci à mes rapporteurs et mon jury de thèse. I would like to deeply thank Damek Davis, who couldn't attend the defense but accepted to review my thesis. Huge thanks for his meticulous reading and his very positive remarks. It was an honor to have you as one of my referees. Je remercie les membres de mon jury de thèse, pour la discussion riche et constructive qui a suivi la présentation, pour leur enthousiasme et leurs retours très encourageants. Merci à Jérôme Malick d'avoir accepté de rapporter ma thèse avec bienveillance et enthousiasme. Cela fait déjà deux mois que je suis arrivé dans ton équipe à Grenoble, et je m'y plais grandement. Je remercie Pascal Bianchi, Gersende Fort et Eric Moulines d'avoir été mes examinateurs. Ce fut un honneur, Eric et Gersende, de vous avoir, et de vous voir apporter à cette soutenance votre vision experte en optimisation stochastique. Pascal, il était évident pour moi de t'avoir dans le jury. Tes travaux ont été une grande source d'inspiration pour une bonne partie de nos résultats et ton esprit aiguisé était sans aucun doute essentiel au sein du jury ! J'aimerais aussi simplement tous vous remercier d'avoir pu vous rendre disponible, d'avoir fait pour certains d'entre vous le déplacement, et de m'offrir une belle soutenance avec un jury présent (en quasi-totalité) en personne. Comme l'avait rappelé Edouard, j'avais commencé ma thèse au téléphone, mes TD, séminaires et conférences se déroulaient en visio-conférence. Alors pour terminer, un peu moins d'interaction virtuelle a été plus que bienvenu !

Si les conférences et les séminaires animent le monde de la recherche et permettent de faire de très belles rencontres, on évoque souvent moins les interactions qui se déroulent dans un cadre quotidien, et qui je pense, nous façonnent en grande partie. Je pense à celles que j'ai eues, le matin en arrivant très motivé, ou le soir après une longue journée de réflexion infructueuse, ainsi qu'aux discussions sans forme ni but que j'ai partagées avec mes voisins de bureau. J'aimerais remercier les nombreuses personnes que j'ai rencontrées au laboratoire de TSE. Merci à mes voisins de bureau, Dana, Tony, Joseph, Thang, Dung, Jinghua, à ceux du bureau d'en face, Lukas, Maurizio, Etienne, Ryan, Camille et Colombe. Je n'oublie pas les plus anciens du laboratoire : Merci à Jérôme (Renault), Abdelaati, Laurent, Anne, Christine, Eric, Adrien, Thibault et les autres que j'oublie

sûrement.

Je pense aussi aux autres personnes que j'ai eu la chance de rencontrer, qui font aussi parties du vaste paysage de la recherche Toulousaine, et en particulier à certaines que j'ai rencontrées bien trop tardivement à mon goût. Merci à Hippolyte, Nathanaël et tout les autres. Merci pour ce cadeau très Grenoblesque, j'en ferai bon usage !

Je remercie les nombreux colocataires qui se sont succédés au 5 rue du Tchad et qui ont rendu mon séjour à Toulouse vraiment très chouette. Merci à Paul pour ses superbes croziflettes et à Nelson pour m'avoir fait découvrir ce merveilleux plat qu'est le Calentao colombien.

J'ai également une pensée pour mes amis de la prépa et de l'ENSAE que je remercie pour leurs encouragements et les très bons moments passés ensembles.

Je remercie Elisa d'être sans cesse à mes côtés malgré la distance, et pour son énorme soutien. Enfin, j'aimerais dédier la fin de ces remerciements à mes parents et mes soeurs. Merci pour leur présence et leur soutien constant, ainsi que pour tout le reste.

Abstract

Machine learning problems often formulate as a risk minimization exhibiting nonsmoothness and nonconvexity. Important sources of nonsmoothness are the privileged use of conditional statements and the presence of sublevel problems.

Stochastic first-order methods are widely employed to address these problems due to their simplicity and scalability, making them an attractive choice for large-scale applications. While classical notions of derivatives for nonsmooth functions are hardly implementable in such contexts, we see a general trend consisting in replacing classical derivatives with the backpropagation algorithm. A recently introduced model of derivatives called conservative gradients provides a justification for such a practice by extending simple calculus rules to nonsmooth functions, such as the chain rule or the sum.

We propose two extensions of the conservative calculus finding a wide range of applications in machine learning. A first result answers the question of interchanging derivative and integral allowing to justify first-order sampling in a nonsmooth and nonconvex setting. A second result is a nonsmooth implicit differentiation formula in order to justify first-order approaches to nonsmooth bi-level problems, e.g. hyperparameter optimization, and the training of implicit layers in deep learning.

We make use of this calculus in order to set up a general nonsmooth stochastic setting that is compatible with practical implementations. A fundamental chain rule along curves allows to apply nonsmooth ODE methods in order to show the convergence of the stochastic subgradient method and its heavy ball version. Some integration results of definable functions are explored in order to ensure a Sard property for continuous distributions.

As a faithful model of practice, the conservative gradient approach yields convergence to a critical set which may depend on the calculus and lead to absurd limit points. For the stochastic subgradient method and its heavy ball version, we show that these calculus artifacts are avoided when randomizing the initialization, leading to the convergence to classical critical points.

Contents

Chapter 1

Introduction

Machine learning is now a genuine asset in our society and is employed for many complex tasks, including recommender systems, image and speech recognition, chatbots, gaming, and scene understanding. Deep learning [102] has revolutionized this field and has known a quick growth over the past decade.

The evolution of deep learning has been characterized by remarkable achievements. It began with the success of the convolutional neural network AlexNet [96] in the ImageNet 2012 Challenge. This breakthrough demonstrated the potential of deep learning in computer vision tasks. Subsequently, deep learning models like AlphaGo, which excelled in the board game Go, and AlphaFold, which made significant strides in protein structure prediction, further showcased the efficiency of deep learning in challenging domains. Other notable examples include Dall-E, an AI[1] model for generative art, and the chatbot ChatGPT. The empirical performance of these deep learning algorithms is often prioritized by practitioners over theoretical guarantees. Demonstrations of superior performance drive the current trend in the field, while unexplained aspects of deep learning models make them an active area of research.

Toward a deeper understanding and improvement of machine learning models, concepts and techniques from classical fields such as statistics and optimization have been rediscovered in this context. For example, the training of a neural network can be seen as an optimization problem, allowing to leverage efficient optimization algorithms [46]. Nonetheless, due to their historical development or practical applications, machine learning models often exhibit properties that pose challenges from traditional perspectives. Interpreting predictions from a neural network and determining convergence during the training process are some of the complex questions that arise in this domain.

In this thesis, we place a particular emphasis on the nonsmooth[2] aspect of machine learning problems, which presents concerns from an optimization standpoint. Common operations in machine learning, such as taking the maximum, thresholding values, or incorporating polyhedral constraints, introduce points of nondifferentiability that necessitate a specific analysis.

[1] Artificial intelligence

[2] "Nonsmooth" is a term with multiple interpretations in the literature. In our context, we will define it as a lack of differentiability at certain points.

1.1 Nonsmooth machine learning problems

Training machine learning models can be seen as a risk minimization:

$$\min_{w\in\mathbb{R}^p} F(w) := \mathbb{E}_{\xi\sim P}[f(w,\xi)], \tag{1.1}$$

where P represents a distribution of data samples, and f is a criterion to minimize in expectation. In many machine learning problems, and particularly in deep learning, the function F to minimize is nonsmooth and nonconvex. While nonsmoothness can be addressed in several contexts, for instance when accompanied with convexity or a specific structure [18, 89], the situations we consider require a specific treatment due to their general nonsmooth and nonconvex aspect.

In this thesis, we will consider two sources of nonsmoothness finding a wide range of applications. The first one is neural networks used for many prediction tasks, and the other one is bi-level problems arising for instance in hyperparameter optimization.

1.1.1 Neural networks

Supervised learning. The minimization problem (1.1) includes a wide class of problems in machine learning called *supervised learning*. In this type of problem, the goal is to predict a target variable $Y \in \mathbb{R}^I$ given an input $X \in \mathbb{R}^d$, in other words, learning a relation $h(X) \approx Y$. Y can be continuous (regression) or discrete (classification). In this context, the random variable ξ is the couple of an input and an output (X, Y), and the integrand f in (1.1) usually writes

$$f(w, X, Y) = \ell(h(w, X), Y). \tag{1.2}$$

$h(w,\cdot)$ is a prediction function parameterized by w and ℓ is a dissimilarity measure. Classical choices for the function ℓ are the square loss, in regression, $\ell(u,v) = \|u-v\|^2$, and the cross entropy in classification. The probability law P is often unknown and represents the distribution of real-world data. In practice, one can have several samples from P, $(x_i, y_i)_{i=1,\dots,n}$, assumed to be drawn independently. In this case, in order to learn a predictor, one can minimize the empirical loss,

$$\min_{w\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n} f(w, x_i, y_i). \tag{1.3}$$

In online settings, the samples can also be obtained through a stream of data, in which case the expectation (1.1) is to minimize sequentially.

Neural networks. In (1.2), the prediction function can take many forms. We focus our attention on predictors used in deep learning, called (artificial) *neural networks*. Classical neural networks are built upon composition of nonlinear functions $(\sigma_l)_{l=0,\dots,L}$, and affine transformations parameterized by $(A_l, b_l)_{l=0,\dots,L}$:

$$\begin{aligned} h(w,x) &= \sigma_L(A_L h_L + b_L) \\ h_L &= \sigma_{L-1}(A_{L-1}h_{L-1} + b_{L-1}) \\ &\dots \\ h_1 &= \sigma_0(A_0 x + b_0), \end{aligned} \tag{1.4}$$

and $L+1$ is the number of layers. When σ_l consists in a function from $\mathbb{R}$ to $\mathbb{R}$ applied componentwise, it is often referred to as an *activation function*, by analogy with the activation of a neuron.

The parameter w to optimize is the vector concatenation of all matrices $(A_l)_{l=0,\dots,L}$ and vectors $(b_l)_{l=0,\dots,L}$. From such a compositional structure one seeks to learn complex relations $h(x) \approx y$ and choosing nonlinear functions σ_l is thus essential.

While a neural network with two layers can approximate any continuous function [61], neural networks with more layers have shown empirically to be more successful for diverse tasks involving large data sets. Examples of these tasks include character recognition, object and speech recognition, or natural language processing. Consequently, the number of parameters and layers can be very high in practice. For instance, AlexNet [96] has 60 million parameters for 8 layers and was trained on a data set of 1.2 million images. Residual networks [86] have up to 1.7 million parameters on 110 layers. GPT-3 [49], a language model, has 175 billion parameters.

Furthermore, many neural network architectures exist. Convolutional networks such as AlexNet are often used on image data and use matrix convolutions which can be represented by circulant matrices $(A_l)_{l=1,\dots,L}$. Residual networks use skip connections and recurrent neural networks [88], used on text data, use input injection which can be represented in (1.4) by fixing some of the matrices values. The matrices $(A_l)_{l=1,\dots,L}$ can be constrained to be equal as for the trellis networks [13].

One of the most popular activation functions is the positive part, commonly referred to as "ReLU" (short for "Rectified Linear Unit") by the deep learning community (fig. 1.1). This function is recurrent in deep learning and used in many models such as convolutional and residual networks [86, 96] or transformer blocks [157] used in language models.

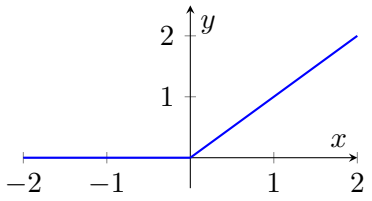


Figure 1.1: ReLU function

Another nonlinear transformation widely used on image data is MaxPooling. Given a matrix input X written as a block matrix with blocks of size d, with the usual choice $d = 2$, the MaxPooling function returns the matrix where each component is the maximum of a block, enabling image down-sampling. Here's an example of a MaxPooling function with a window of size 2:

$$\text{MaxPooling}\left(\begin{bmatrix} 3 & 1 & \mathbf{13} & 8 \\ 7 & \mathbf{20} & 6 & 2 \\ \mathbf{7} & 3 & \mathbf{5} & 4 \\ 6 & 2 & 1 & 0 \end{bmatrix}\right) = \begin{bmatrix} \mathbf{20} & \mathbf{13} \\ \mathbf{7} & \mathbf{5} \end{bmatrix}.$$

The MaxPooling function appears in many architectures such as the residual networks for image classification or YOLO models [132] for real-time object detection.

As we may notice, deep learning models are often nonsmooth. The ReLU function is not differentiable at zero, while the MaxPooling function is not differentiable whenever some components of a block are equal. This may raise some concerns if we consider the training as a continuous optimization problem, where differentiability is usually desirable. Yet, the ubiquity of nonsmoothness is not really justified and while the use of the ReLU function appears to be successful, it is not known whether nonsmoothness is strictly required. For instance, some parts of the GPT-3 model use smooth activation functions such as GeLU or softmax [49].

In fact, prior to their success, the development of artificial neural networks was rather independent of the optimization field. Early attempts to address classification tasks involved simplified representations of neural networks, notably the perceptron model introduced by Rosenblatt [138]. In these primary models, synaptic connections were represented using matrix products, and neural activations were expressed through binary outputs or thresholded values. Despite their nonsmooth-

ness, some of these simplistic aspects seem to remain even in state-of-the-art models, as illustrated by the ReLU function.

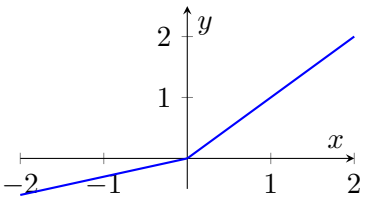


Figure 1.2: Leaky ReLU

Deep learning models continuously evolve, adapting to new problems. The field explores alternative architectures and activation functions, for instance, the sort function [7] was used to promote Lipschitzness. Many variants of the ReLU function also exist like the leaky ReLU [106], fig. 1.2. In this thesis, we will be interested in recent classes of layers called implicit layers and optimization layers. Optimization layers will be discussed in Section 1.1.2 due to their bi-level aspect.

Implicit layers. Introduced in recent works, some neural networks use layers whose output is defined by an implicit equation [12, 80]. For instance, in deep equilibrium networks [12], the output z of a layer is defined by a fixed point equation,

$$z = \sigma(Wz + b + Ux), \tag{1.5}$$

(W, U, b) are parameters to train, x is the input and σ plays the role of an activation function. For instance, σ can be the ReLU function. This kind of layer is inspired by the model of trellis networks [13], where the matrices of the layers are constrained to be equal, resulting in fewer trainable parameters. The fixed point formulation (1.5) not only yields a model with fewer parameters but also eliminates the need to store outputs for intermediary layers, thereby reducing memory costs during training. Despite the reduction in terms of parameters, these models exhibit competitive performance when compared to traditional deep neural networks.

1.1.2 Bi-level problems

In some situations, the function F to minimize is defined upon another optimization problem, leading to nonsmoothness. In *bi-level* problems, the objective involves an argmin term:

$$\min_{w \in \mathbb{R}^p} F(w, z) \quad \text{such that } z \in \operatorname*{argmin}_{\theta \in \mathcal{C}} g(w, \theta).$$

This type of problem was studied in optimization beforehand [65, 163] and now finds a renewed interest in machine learning with several applications such as data augmentation [60, 137] and hyperparameter optimization [23, 25, 26]. Two applications will be of interest in Chapter 3 Section 3.3.3 of this thesis, hyperparameter optimization for lasso-type models [25] and convex optimization layers [6], in line with differentiable conic programming [2, 3].

Optimization layers. Optimization layers, which were recently introduced in [2, 6], represent a novel type of architecture where the layer's output is obtained as the solution of a convex optimization problem. The training of such models thus writes as a bi-level problem. These optimization layers have demonstrated their effectiveness in many applications when it comes to modeling prior knowledge and learning hard constraints [6, 78, 101].

Hyperparameter optimization. Several machine learning problems incorporate a regularization term R:

$$\min_{w \in \mathbb{R}^p} F(w) + R(\lambda, w). \tag{1.6}$$

Classical choices are the squared norm, $R(\lambda, w) = \lambda \|w\|^2$, used in deep learning under the name "weight decay" [96] or the ℓ_1 norm $\lambda \|w\|_1$. The ℓ_1-norm penalty finds a strong interest in machine learning and statistics due to its property to provide sparse solutions, allowing variables selection in high dimension, with the lasso estimator [152], or sparse signal recovery by basis pursuit [53]. A recurrent question when adding a regularization term is that of the choice of the penalty level, the hyperparameter λ. A common approach is to proceed by cross-validation and to maximize a criterion with respect to the penalty level, which can be formulated as a bi-level problem. When the hyperparameter is one-dimensional, a grid search is generally sufficient.

However, in some variants of the lasso estimator, the regularization term may contain more than one hyperparameter like in the adaptive lasso [166], which aims to reduce the estimator bias while preserving sparsity. Thus, a relevant question arises as to apply more efficient optimization algorithms starting with first-order methods [25]. In the case of the lasso estimator, the solution path is piecewise linear with respect to the hyperparameter [70], thus leading to a nonsmooth bi-level problem.

Other problems, not of interest in this thesis can also involve the value function of the sublevel problem and generate nonsmoothness as well. In min-max problems, F is defined as a pointwise maximum,

$$F(w) := \max_{\theta \in \mathcal{C}} g(w, \theta).$$

Some typical applications of this setting are generative adversarial networks for image generation [8], distributionally robust and risk-averse optimization [84, 97, 144].

1.2 First-order optimization in machine learning

For the moment, let us put aside the nonsmooth aspect. In the differentiable setting, first-order methods, like the gradient method (1.7), are often used to deal with the minimization problem (1.1).

$$\text{for } k \in \mathbb{N}, \quad w_{k+1} = w_k - \alpha_k \nabla F(w_k). \tag{1.7}$$

This popularity can be explained by their simplicity, the recent development of efficient software to compute gradients, *automatic differentiation* [79] or *backpropagation* [141], and more efficient ways to leverage computational power with the adapted use of GPUs[3] [52] in order to handle many parameters.

1.2.1 Practical implementation of first-order methods

In this part, we expose some practices when it comes to implement first-order methods in machine learning. In order to compute the gradient of a differentiable function $f : \mathbb{R}^p \to \mathbb{R}$, a simple approach would be to approximate the partial derivatives of f by finite difference:

[3] Graphics processing unit.

$$\nabla f(w) \approx \frac{1}{t} \begin{bmatrix} f(w + te_1) - f(w) \\ f(w + te_2) - f(w) \\ \vdots \\ f(w + te_p) - f(w) \end{bmatrix}$$

where t is small enough, and for $i = 1, \dots, p$, e_i is the i^{th} element of the canonical basis of $\mathbb{R}^p$. The cost of this method is approximately $p \times \text{cost}(f)$ where $\text{cost}(f)$ is the cost of computing f. In many machine learning situations such as deep learning, the parameter dimension p is high hence using this method would be unreasonable.

Automatic differentiation. Automatic differentiation [79], also called "backpropagation" in the deep learning community [102, 141], is an efficient algorithm for computing the gradient of a function $f : \mathbb{R}^p \to \mathbb{R}$ by automating the chain rule formula on elementary functions available in a programming language (exponential, logarithm, sinus, cosinus...). This allows for more efficient and exact computation of gradients. Automatic differentiation is available on Python libraries such as TensorFlow [1], Pytorch [126], or JAX [47].

For rational functions, a fundamental result from Baur and Strassen [17] states that the cost of automatic differentiation is at most 5 times the cost of computing the function to differentiate. This result has been extended to differentiable functions [79]. Compared to the finite difference method, the computational cost in terms of the function's cost doesn't increase with the dimension.

In deep learning, this method can raise some concerns. In particular, computing the chain rule formula for the composition (1.4) requires the intermediate values $(h_l)_{l=0,\dots,L}$ of the composition to be stored, which can be expensive in terms of memory when the number of layers L is high. Some solutions have been proposed in the literature to reduce this memory cost, such as the use of implicit layers (Section 1.1.1).

Implicit differentiation. We will have a particular focus on the differentiation of implicitly defined functions that arise for instance in implicit models, seen in Section 1.1.1. In the differentiable setting, for an equation $H(w, y) = 0$, the implicit function theorem gives under some conditions, the existence and differentiability of a solution map $y^*(w)$, leading to a derivative of y with respect to w:

$$\operatorname{Jac} y^*(w) = -(\operatorname{Jac}_y H(w, y))^{-1} \operatorname{Jac}_x H(w, y). \tag{1.8}$$

Some works [2, 3, 25] proposed to use it in order to deal with bi-level problems, by differentiating optimality conditions of the sublevel problem. In the case of simple convex problems, this method appears to be really flexible since deriving optimality conditions can be automated via disciplined convex programming [3, 4].

First-order sampling. In the stochastic minimization problem (1.1) where F writes as an expectation, computing the gradient ∇F or applying automatic differentiation is usually not tractable in large-scale settings. In training procedures involving a large data set, the empirical risk (1.3) becomes a large sum hence computing ∇F at each iteration is too expensive. In online settings, samples from an unknown distribution P arrive in succession in which case the classical gradient method can't be applied.

In order to deal with these settings, it is common to consider a stochastic gradient method, going back to the seminal work of Robbins and Monro [134]. This method writes in the differentiable setting as: for $k \in \mathbb{N}$ do

$$\begin{aligned} &\text{sample } \xi_k \sim P \\ &w_{k+1} = w_k - \alpha_k \nabla_w f(w_k, \xi_k) \end{aligned} \tag{1.9}$$

where $\nabla_w f(\cdot, \xi_k)$ is the gradient of $f(\cdot, \xi_k)$. Although not studied in this thesis, further improvements can be added to this algorithm. For instance, averaging several stochastic gradients can help to reduce variance. In practice, random reshuffling is also used in place of random sampling, leading to faster convergence [83, 92, 115].

This method is consistent with the deterministic gradient method since the expectation of the stochastic gradient $\nabla f(\cdot, \xi)$, with $\xi \sim P$, is equal to the gradient of the expectation F, thanks to the Leibniz integral rule, see e.g. [140, Chapter 9], which allows interchanging expectation and gradient.

1.2.2 On the variants of the gradient method used in machine learning

In the context of deep learning, we see the development of many variants of the stochastic gradient method with the goal to accelerate computationally intensive training phases.

Momentum methods. Introduced by Polyak [128], the heavy ball method is a variant of the gradient method which includes a momentum term:

$$w_{k+1} = w_k - \mu_k \nabla F(w_k) + \nu_k (w_k - w_{k-1}),$$

and may be considered with first-order sampling. Although no theoretical study explains its success in the nonsmooth and nonconvex setting, this method is widely used in deep learning, popularized by pioneer works [96, 151].

Adaptive methods. A recurrent question coming with the gradient method is the choice of the stepsizes $(\alpha_k)_{k \in \mathbb{N}}$. In the context of deep learning models, the optimal choice of stepsizes remains unknown. Due to the long training phases, vanishing stepsizes suggested by the stochastic approximation literature [134] can lead to numerical issues which may slow down the training process. On the other hand, tuning a constant stepsize can be expensive. For these reasons, adaptive stepsizes are often preferred. For instance, AdaGrad [68, 150] defines adaptive stepsizes according to the past gradients' evaluations. Its scalar version writes as follows:

$$w_{k+1} = w_k - \frac{1}{\sqrt{\epsilon + \sum_{i=0}^{k} \|\nabla F(w_i)\|^2}} \nabla F(w_k)$$

Other variants of adaptive stepsizes exist like RMSProp [154]. Furthermore, momentum and adaptive stepsizes can also be combined as in ADAM [91] and AMSGrad [131]. The method ADAM is widely used in deep learning due to its empirical success and it is included in automatic differentiation frameworks as a built-in method [1, 126].

1.2.3 Implementation of first-order methods on nonsmooth functions

We now go back to the nonsmooth setting. Despite the ubiquitous presence of nonsmoothness in machine learning, as illustrated in the examples from Section 1.1, practitioners have not been deterred from using first-order methods. In fact, one sees an overall trend consisting in the use of automatic differentiation on nonsmooth functions, and to replace classical derivatives with automatic differentiation outputs.

Nonsmooth automatic differentiation. In deep learning, automatic differentiation can be applied to nonsmooth functions. Nondifferentiability points of functions implemented in practice are generated by conditional statements (*if* and *else*), allowing to apply automatic differentiation on each part of the computational tree. In order to understand this mechanism, we may consider the following representation of ReLU with conditional statements:

$$\operatorname{relu}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{else.} \end{cases}$$

The output of automatic differentiation will be the following map:

$$\operatorname{backprop} \operatorname{relu}(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{else.} \end{cases}$$

An important point to notice here is that the output of automatic differentiation depends on the implementation of the function. Automatic differentiation actually acts on the program representing the function, not on the function itself. For instance, changing the conditional statement $x > 0$ into $x \geq 0$ would not change the value of the function relu, but it would modify the value of automatic differentiation at zero from 0 to 1.

This procedure allows for a "formal automatic differentiation". For instance, in order to provide a first-order oracle to the composition of nonsmooth functions, the chain rule formula can be applied replacing classical Jacobians by the outputs of automatic differentiation. This way of differentiating nonsmooth functions is the basis of neural network training [102, 141]. A recent work [31] extends the complexity result of Baur and Strassen [17] for nonsmooth compositions, hence justifying its computational efficiency in deep learning applications. Some mathematical models [37, 122], to be exposed in Section 1.4, were proposed in the context of deep learning in order to justify this differentiation rule of nonsmooth functions.

We will look at two situations where automatic differentiation is applied formally: implicit differentiation, and first-order sampling.

Nonsmooth implicit differentiation. In practical cases highlighted in Section 1.1, the implicit relation H is often nonsmooth. For instance, implicit layers (1.5) can involve a nonsmooth function σ such as ReLU. Optimality conditions of sublevel problems arising in hyperparameter optimization or convex optimization layers are generally expressed by a Lipschitz and nonsmooth map.

Despite this, in practice [12,25], the implicit differentiation formula (1.8) is applied to nonsmooth functions thanks to the backpropagation algorithm. In order to provide an oracle for the solution path $y^*(w)$ of $H(w, y) = 0$, one can replace the classical Jacobian (1.8) by the backpropagation output applied to H:

$$\operatorname{Implicitdiff} y^*(w) := -(\operatorname{backprop}_y H(w, y))^{-1} \operatorname{backprop}_x H(w, y) \tag{1.10}$$

Although this formula coincides with the Jacobian of y^* in the differentiable setting, the current theory doesn't justify it in the case of nonsmooth functions. A main contribution of this thesis in Chapter 3 Section 3.3 is a nonsmooth implicit differentiation formula that justifies this practice.

Nonsmooth first-order sampling. First-order sampling can be applied with the backpropagation oracle. For instance, in supervised learning with a neural network predictor, we may consider the stochastic gradient method with backpropagation:

$$\begin{aligned} &\text{sample } \xi_k \sim P \\ &w_{k+1} = w_k - \alpha_k \operatorname{backprop}_w f(w_k, \xi_k) \end{aligned}$$

In this case, one has to justify for such a procedure. In particular, one has to justify whether sampling $\operatorname{backprop}_w(f(\cdot, \xi_k))$, which resulted from nonsmooth automatic differentiation, approximates a descent direction for F at w_k.

In this thesis, we establish a result in Chapter 3 Section 3.2, that allows the interchange of the integral and derivative operations for nonsmooth functions. This result aims to provide a theoretical basis to justify first-order sampling in the nonsmooth nonconvex setting. We are motivated by the ubiquity of stochastic settings in data-driven approaches and a growing variety of online scenarios in machine learning, including reinforcement learning [76], decentralized and federated learning [82, 93]. Other general stochastic settings may be the use of loss functions that are averaged over an absolutely continuous distribution, such as in Bayesian inference where this procedure enables to incorporate uncertainty in model predictions [30].

By considering the procedures described above, first-order methods find their implementable nonsmooth counterparts thanks to automatic differentiation. This way, the gradient method and its variants seen in Section 1.2.2 can be transposed to the nonsmooth setting.

1.3 Analysis of first-order algorithms

In this thesis, we seek to justify the convergence of first-order methods in machine learning from the point of view of optimization. In particular, we want to study first-order methods, as implemented in practice, for instance, the stochastic gradient method with backpropagation (1.11) or with the use of implicit differentiation (1.10). Our main questions will be: Do the iterates converge to "critical points", in some sense? Does the objective function converge?

As we will see, addressing these questions in the nonsmooth setting requires the use of specific tools. In order to grasp these, it is important to understand that the analysis in the nonsmooth setting relies on similar mechanisms as those employed in the smooth setting.

1.3.1 Lyapunov analysis in the smooth nonconvex setting

In the smooth setting, a common approach to analyze first-order methods is to view them as approximations of a continuous-time dynamical system. This approach, often called the ODE[4] method, was first introduced in the early works [98, 105] and has been extensively explored in the stochastic optimization literature [15, 19, 27, 66, 77, 99]. In [19], the author considers a general stochastic process written as

$$w_{k+1} = w_k + \alpha_k(H(w_k) + \epsilon_k)$$

[4]Ordinary differential equation.

with vanishing stepsizes $\alpha_k \to 0$ and a centered noise term ϵ_k. He shows that this process can be studied through the solutions of the differential equation $\dot{w} = H(w)$ as it satisfies the property to be an *asymptotic pseudo-trajectory*.

The term $H(w_k) + \epsilon_k$ represents a noisy measurement of $H(w_k)$. For instance in the case of the stochastic gradient method (1.9), $H(w_k)$ is the gradient of the risk, $\nabla F(w_k)$, and $H(w_k) + \epsilon_k$ is the stochastic gradient $\nabla_w f(w_k, \xi_k)$. This method can then be studied through the limiting gradient flow,

$$\dot{w} = -\nabla F(w). \tag{1.11}$$

It allows to justify the convergence of the stochastic gradient method to the critical set $\{w : 0 = \nabla F(w)\}$. This method applies more generally to first-order algorithms admitting a Lyapunov function that decreases along the trajectories of the continuous-time dynamical system. In the case of the gradient flow, F acts as a Lyapunov function and decreases along the gradient curves (1.11) outside the critical set. The heavy ball method can be studied as in [77] by considering the Lyapunov function $E(w, \dot{w}) = F(w) + \frac{1}{2}\|\dot{w}\|$. This approach was also employed to study adaptive algorithms [15].

On the Sard property. In order to conduct a proper Lyapunov analysis, a recurrent condition requires the critical values to have an empty interior [19,66,77]. This requirement can be referred to as Sard property, from Sard's theorem [146] which asserts that a sufficient degree of differentiability of the function allows to satisfy this condition. In the case of the gradient method for instance, this condition enables to show the convergence of the objective function and that the accumulation points of the iterates are critical points.

The ergodic approach. Algorithms can also be examined from the point of view of measures [19, 22]. In particular, the trajectory of the iterates can be represented as a continuum of Dirac measures, which is referred to as the *occupation measure*. The notions of convergence and limit are then understood in the space of measure. For instance, the accumulation points of the algorithm are associated to limit measures supported on the stationary points of the flow.

This methodology allows the derivation of weak convergence results in a more general setting. In [19], the author considers situations with slower stepsizes, while the authors of [27] treats the case of constant stepsizes. Notably, this method enables the derivation of convergence results even in the absence of the Sard property.

Toward the nonsmooth setting. The analysis of nonsmooth first-order algorithms is based on similar principles. In order to have a decreasing Lyapunov function, specific notions of regularity for nonsmooth functions can be considered and will be presented in Section 1.3.2. In Section 1.3.4, we will see that the ODE method and the ergodic approach can be adapted to nonsmooth algorithms by considering differential inclusions instead of differential equations. As to the Sard property, it can be satisfied by considering semialgebraic or tame functions, often encountered in practical cases, where points of nondifferentiability are organized into smooth manifolds (Section 1.3.3).

1.3.2 Regularity of nonsmooth functions

Long before the emergence of modern machine learning problems, such as those we highlighted in Section 1.1, nonsmooth and nonconvex problems had already piqued the interest of the optimization community. Regarding convex functions, it is known that the notion of gradient can be extended

to nonsmooth functions $f : \mathbb{R}^p \to \mathbb{R}$ by considering the set-valued map ∂f satisfying for all x, for all $v \in \partial f(x)$,

$$f(z) \geq f(x) + \langle v, z - x \rangle \text{ for all } z \in \mathbb{R}^p. \tag{1.12}$$

This definition was introduced by Rockafellar [135,136] and also by Moreau in [117], where he calls ∂f the subgradient of f. In the convex setting, an observation that can be made from the inequality (1.12) is that the mapping ∂f possesses an inherent variational interpretation, and accounts for the local variations of the function. It can also be seen as a one-sided first-order approximation.

Motivated by the minimization of maximum value functions, the notion of subgradient was then extended to locally Lipschitz nonsmooth functions by Clarke [55]. He defined it as the convex-valued graph closure of the gradient. Precisely, for a locally Lipschitz function $f : \mathbb{R}^p \to \mathbb{R}$, the Clarke subgradient is given for all $x \in \mathbb{R}^p$ by

$$\partial^c f(x) = \operatorname{conv}\{v \in \mathbb{R}^p \ : \ v = \lim_{k\to\infty} \nabla f(x_k), x_k \underset{k\to\infty}{\to} x, \{x_k\}_{k\in\mathbb{N}} \subset \operatorname{diff}_f\},$$

where diff_f is the differentiability set of f. Nonetheless, while this oracle is well-defined for any locally Lipschitz function, it generally lacks variational information.

Indeed, some Lipschitz functions have a maximal Clarke subgradient [45,135] equal to the unit ball everywhere, hence making it impossible to access descent directions. An example from [135] is as follows: let $A \subset \mathbb{R}$ be such that A and A^c are dense, and for all open interval I neither $A \cap I$ nor its complement in I have zero Lebesgue measure. Then, consider the function

$$f : x \to \int_0^x 2\mathbb{1}_A(s) - 1 \, \mathrm{d}s. \tag{1.13}$$

The Clarke derivative of f is equal to $[-1, 1]$ everywhere. In the absence of variational information in the general locally Lipschitz setting, a further regularity notion is needed in order to give a sense to nonsmooth first-order algorithms.

Semismooth functions. Mifflin [113] introduced a class of nonsmooth functions called *semismooth*, which exhibits favorable variational properties with respect to the Clarke subgradient. Semismooth functions $f : \mathbb{R}^p \to \mathbb{R}$, satisfy at each $x \in \mathbb{R}^p$, for y in a neighborhood of x,

$$f(y) = f(x) + \langle v, y - x \rangle + o(\|x - y\|), \quad \text{as } y \to x, \ \text{ for all } v \in \partial^c f(y).$$

Examples of semismooth functions are convex functions and pointwise maximum over a compact family of smooth functions. These functions are also stable under composition. Mifflin proposed an algorithm [112] for a semismooth objective that converges to stationary points of a constrained problem. Semismooth functions were also explored in the context of Newton's methods in [130].

Path differentiable functions. In line with a chain rule lemma for convex functions from Brézis [48, Lemma 3.3], Valadier [155] introduced the notion of "non-pathological functions" in the context of Moreau's sweeping process [118]. They are locally Lipschitz functions f satisfying a chain rule along absolutely continuous curve $\gamma : [0, 1] \to \mathbb{R}^p$: for almost all $t \in [0, 1]$,

$$\frac{\mathrm{d}(f \circ \gamma)}{\mathrm{d}t}(t) = \langle \partial^c f(\gamma(t)), \dot{\gamma}(t) \rangle. \tag{1.14}$$

This chain rule property seems favorable for optimization since it implies that the function f decreases along the curves of the subgradient flow $\dot{\gamma} \in -\partial^c f(\gamma)$. It has recently been rediscovered in nonsmooth optimization in several works, first [63] which calls it a *chain rule* property and then [37, 41] which employ the terminology *path differentiable* functions.

On the links between the two notions. While both notions were introduced in different contexts, several works have highlighted their connections. Borwein and Moors [44] showed that semismooth functions satisfy the property to be essentially smooth, hence satisfying a weaker chain rule. Ruszczyński [142] showed a chain rule along semismooth curves for semismooth functions. In this thesis, Chapter 3 Section 3.4, we establish that semismoothness implies path differentiability but the converse is false in general.

Simple functions encountered in machine learning such as semialgebraic ones actually satisfy both notions [33, 63] and pathological functions such as (1.13) are hardly encountered. A setting that excludes pathological cases can be considered rigorously with the *o-minimal structures*.

1.3.3 O-minimal structures: a favorable setting for optimization

Tame functions are semismooth and path differentiable. In a topological sense, functions exhibiting a unit ball Clarke subgradient, hence neither semismooth nor path differentiable, are predominant in the space of Lipschitz functions [45]. Yet, it is hard to conceive that simple functions met in machine learning, e.g. semialgebraic functions, can exhibit such a pathological aspect. Actually, the non-differentiability sets of these functions are well-structured. In the case of the ℓ_1 norm or ReLU networks, for instance, the non-differentiability sets are organized into affine subspaces.

One can hope for a similar phenomenon when considering compositions involving other common functions such as the exponential or the logarithm. In fact, semialgebraic functions can be generalized in broader dictionaries of elementary functions, including for instance the exponential, with the o-minimal structures [59, 156]. Considering functions belonging to such structures, and called *definable* or *tame*, allows to formally exclude pathological cases such as the function (1.13).

In the case of nonsmooth semialgebraic or definable functions, nondifferentiability sets are well-structured, and organized in smooth manifolds. This can be formulated with the *Whitney stratification* property [160]. This property leads to a projection formula [40] for the Clarke subgradient. Based on this result, [33] showed that definable and locally Lipschitz functions are semismooth, and later on in [63], that these functions are also path differentiable.

Definable Sard's theorem. An important consequence of the definable setting is that it excludes pathological situations where the Sard property doesn't hold. Indeed, definable functions satisfy the Sard property [39]. As we mentioned in Section 1.3.1, this condition plays a fundamental role in the development of a convergence theory but it often appears as an abstract assumption [21, 66, 72, 77, 142]. The definable setting allows to obtain it in a more reasonable way than the classical Sard's theorem which requires a degree of differentiability at least equal to the dimension of the parameter.

Outside the definable setting, many pathological behaviors can happen. A counterexample to Sard's theorem was first given by Whitney [161] with a fractal construction. Another one exhibiting pathological subgradient sequences was proposed in [133].

The o-minimal structures and their consequences in optimization will be presented in detail in Chapter 2. Examples from machine learning are also exposed in Section 2.2.3.

1.3.4 Differential inclusion method

The ODE method presented in Section 1.3.1, has been extended to set-valued dynamics in [21]. The authors of [21] adapt the notion of asymptotic pseudo-trajectories to set-valued recursions

$w_{k+1} \in w_k + \alpha_k(H(w_k) + \epsilon)$ where H is now a set-valued map with compact and convex values, and closed graph. Similarly to the smooth ODE method, the discrete recursion is seen as an approximation of the differential inclusion $\dot{w} \in H(w)$.

This setting encompasses nonsmooth first-order methods. For instance, it allows to study a stochastic subgradient method

$$w_{k+1} \in w_k - \alpha_k(\partial^c f(w_k) + \epsilon_k), \tag{1.15}$$

as a discrete stochastic approximation of the subgradient flow $\dot{\gamma} \in -\partial^c f(\gamma)$. It was for instance used to study the stochastic subgradient method (1.15) in [63,108], its heavy ball version [142] and an inertial method [51].

The closed-measure approach. The ergodic approach [19, 22] has been extended to the set-valued setting [29,75] and also explored in the case of the subgradient method in [41]. The authors of [29, 41] propose an algorithmic interpretation of the weak convergence in terms of *essential accumulation points*. As for the smooth setting, one of the main advantages of this approach is that it provides convergence results in a more general setting, e.g. without the Sard property, allowing for convergence results beyond the definable case. Additionally, the framework proposed in [29,41] includes a more precise study of the algorithm's behavior, allowing to capture oscillations that may be neglected in traditional convergence analyses.

1.3.5 An illustrative summary: how to analyze subgradient method?

Gathering the previous notions, we summarize the methodology to analyze the subgradient method. This approach is canonical and can be applied to other algorithms admitting a Lyapunov system. Some technical details are indeed omitted for this introduction and a comprehensive summary of the analysis will be presented in Chapter 4. Consider a function F to optimize, and a stochastic subgradient method

$$w_{k+1} \in w_k - \alpha_k(\partial^c F(w_k) + \epsilon_k)$$

with vanishing stepsizes α_k and where ϵ_k is a conditionally centered noise.

If we assume F to be semialgebraic or definable, F is path differentiable (Section 1.3.3). It means that F decreases along the curves of the subgradient flow,

$$\dot{w} \in -\partial^c F(w),$$

hence F is a Lyapunov function. In this case, we can rely on the differential inclusion method (Section 1.3.4) in order to study the stochastic subgradient method. The Sard property, obtained by the definable setting, allows to do a sharp Lyapunov analysis and to obtain the convergence of the objective function, as well as the criticality of all accumulation points w^*, that is $0 \in \partial^c F(w^*)$. Outside this condition, the closed-measure approach enables to show weaker convergence results.

Path differentiability was exploited in [37, 63] in order to prove the convergence of the subgradient method (1.15). An analysis of the deterministic subgradient method for path differentiable functions was carried out in [41] with a study of oscillations thanks to the closed-measure approach.

On the convergence analysis for semismooth functions. Semismoothness enables an asymptotic descent lemma in [72] leading to the same convergence results as the ODE method for the stochastic subgradient algorithm. The (nonsmooth) ODE method is often favored in the literature

due to its versatility. For example, various works that consider semismooth functions [84, 142, 143] employ the ODE approach through a weak form of path differentiability.

In Chapter 3 Section 3.4, we show that semismooth functions are path differentiable, allowing to apply the differential inclusion approach when considering semismooth functions.

1.3.6 Limits of the nonsmooth theory.

The theory presented above exhibits several limitations when considering practical machine learning settings.

A first gap: tame geometry and stochastic optimization. As highlighted at the beginning of this introduction, machine learning problems involve the minimization of a general expectation (1.1). While the integrand f can reasonably be taken semialgebraic or definable, this property is not preserved when applying the expectation. Some technical considerations were hence made in the works aforementioned. For instance, in order to obtain path differentiability of the risk, [63] considers a discrete and finite distribution. Sard property is also obtained with this setting in [37,63] thanks to the stability of semialgebraic and tame functions under finite sum.

A part of this thesis aims to go beyond these considerations. In particular, we prove the path differentiability of general risk functions thanks to an integral rule in Chapter 3 Section 3.2. This allows to apply the ODE method, in particular the closed measure approach in order to obtain convergence results outside the Sard property. For the latter, we propose to obtain it in the case of a wide range of absolutely continuous distributions by using a result on the integration of definable functions [56], which will be presented in Chapter 2 Section 2.2.5. This result was not introduced in the stochastic optimization literature until now but it allows to justify Sard property outside empirical risk minimization [37, 63].

A second gap: Clarke subdifferential and calculus. While nonsmooth optimization theory relies on the Clarke subdifferential, this oracle is not really used in machine learning. Machine learning models are complex, in particular in deep learning, hence their training highly relies on the application of calculus formulas in order to build first-order oracles.

As we saw in Section 1.2.3, backpropagation is the application of the chain rule of differentiable calculus on the composition of nonsmooth functions. Nonsmooth implicit differentiation is used for differentiating implicit layers or solutions to optimization problems. We also recall that first-order sampling, e.g. the stochastic gradient method, is justified in the smooth case by the interchanging of expectation and gradient $\mathbb{E}_{\xi\sim P}[\nabla_w f(\cdot,\xi)] = \nabla F$. In the nonsmooth case, first-order sampling is used with backpropagation assuming that a descent direction is approximated and that a nonsmooth version of the interchanging rule holds.

Unfortunately, in general, the application of these rules within the Clarke subdifferential theory remains heuristical, and the conventional calculus rules do not hold with Clarke subderivatives. For instance, the chain rule doesn't hold. As to implicit differentiation, a theorem exists for the regularity and existence of the implicit function in the case of a nonsmooth Lipschitz equation [54] but it doesn't come with a nonsmooth implicit differentiation formula.

Regarding first-order sampling, an interchanging rule for integral and Clarke subgradient doesn't exist. Several works [63, 108] consider the stochastic subgradient method written as (1.15) but it doesn't represent the practice. Besides the use of backpropagation, the stochastic subgradient method, which samples stochastic subgradients, writes

$$w_{k+1} \in w_k - \alpha_k \partial_w^c f(w_k, \xi_k)$$

where $\partial_w^c f$ is the Clarke subgradient with respect to the first variable. In general,

$$\partial^c F \subset \mathbb{E}_{\xi \sim P}[\partial_w^c f(w_k, \xi)],$$

but the equality is not true.

Justifying nonsmooth calculus is crucial to establish the convergence of algorithms as they are practically implemented. Moreover, in a domain such as deep learning where models are abundant, a versatile and user-friendly theory is desirable. In the following section, we will present two variational models [37, 72] that offer a straightforward justification for most calculus rules used in machine learning.

1.4 An operator-free view of nonsmooth calculus

Before the introduction of deep neural networks and automatic differentiation, the need for a nonsmooth calculus was already expressed in the convex optimization literature. Indeed, one may not always obtain the subgradient of the sum by summing subgradients, nor do we have a general counterpart to the chain rule for computing subderivatives in compositions. Consequently, conditions to ensure the validity of simple calculus rules such as the sum or composition were established [54, 136], requiring convexity, Clarke regularity, or qualification conditions in the case of constrained problems.

Regarding stochastic problems, the interchanging $\partial F = \mathbb{E}_{\xi \sim P}[\partial_w f(\cdot, \xi)]$ holds when f is convex in its first argument. This rule is fundamental in stochastic problems. For instance, it justifies subgradient sampling in order to design online first-order algorithms, statistical consistency of stochastic programming schemes, e.g. enabling a law of large numbers, with applications to risk-averse optimization and stochastic variational inequalities, see for instance [148].

As highlighted in the previous part, the Clarke subdifferential doesn't allow a calculus in the general nonsmooth nonconvex setting. Two variational models [37, 120] have been proposed in order to extend calculus rules to nonsmooth nonconvex functions. In these models, the derivatives of a function are set-valued maps, regarded as non-unique entities:

Norkin's semismooth derivatives. The notion of generalized semismooth gradients was proposed by Norkin [120]. They are closed graph and locally bounded set-valued maps D_f satisfying the semismooth property with respect to f: for all $x \in \mathbb{R}^p$,

$$f(y) = f(x) + \langle v, y - x \rangle + o(\|x - y\|) \text{ as } y \to x, \text{ for all } v \in D_f(y).$$

Conservative derivatives Bolte and Pauwels [37] proposed the notion of conservative gradients which are set-valued maps D_f satisfying the chain rule along absolutely continuous curves $\gamma : \mathbb{R}_+ \to \mathbb{R}^p$: for almost all $t \geq 0$,

$$\frac{\mathrm{d}(f \circ \gamma)}{\mathrm{d}t}(t) = \langle v, \dot{\gamma}(t) \rangle \text{ for all } v \in D_f(\gamma(t)).$$

As we may notice, these two models are based on semismoothness and path differentiability properties seen in Section 1.3.2. Formerly, the construction of a first-order oracle in the smooth setting or

in the case of the Clarke subdifferential came from a formula involving the function, resulting in a unique oracle for a given function. In these new models, which we may qualify as "operator-free", a first-order oracle is defined by its property to account for the variations of the function, hence many set-valued maps representing a derivative can exist for the same function.

These models justify most simple calculus rules. For instance, for two semialgebraic functions f and g, taking the sum of the subgradients $\partial^c f + \partial^c g$ may not be equal to the subgradient of $f + g$, but it is a conservative gradient for $f + g$. A similar mechanism is obtained when using the chain rule on Clarke Jacobians: applying the chain rule on a composition $F \circ G$ with the Clarke Jacobians $\mathrm{Jac}^c F$ and $\mathrm{Jac}^c G$ outputs the set-valued map $\mathrm{Jac}^c F(G)\,\mathrm{Jac}^c G$ which is a conservative Jacobian for $F \circ G$. This provides a variational model for nonsmooth automatic differentiation. Different versions of automatic differentiation used in practice, such as the forward and reverse modes [36, 37], can be modeled this way.

The same rules work for semismooth derivatives: for instance, taking the sum of two semismooth gradients yields a semismooth gradient for the sum. An integral rule holds for semismooth derivatives [114, 121], in particular if $D(\cdot, \xi)$ is a semismooth gradient for $f(\cdot, \xi)$ almost surely in $\xi \sim P$, then taking the expectation $\mathbb{E}_{\xi\sim P}[D(\cdot, \xi)]$ gives a semismooth gradient for $\mathbb{E}_{\xi\sim P}[f(\cdot, \xi)]$, hence justifying first-order sampling for semismooth gradients.

The advantage of these operator-free approaches is that they allow for a faithful and simple justification of the calculus rules. In order to justify automatic differentiation in alignment with the Clarke subdifferential model, a framework was proposed in [90]. However, this approach requires satisfying qualification conditions and implementation rules that may not reflect common practice.

In the precise case of semialgebraic or definable functions, it was shown that semismooth and conservative derivatives are equivalent [62]. In Chapter 3 Section 3.4, we investigate the differences between both notions in the general case and establish that semismooth gradients are conservative gradients but the converse is false.

In this thesis, we focus on the model of conservative derivatives that we present in Chapter 3 and propose two extensions. In Section 3.2, a first extension of the conservative calculus will be an integral rule, justifying first-order sampling in the path differentiable case. A second one, in Section 3.3, will be a nonsmooth implicit differentiation formula with applications to implicit models and bi-level programming in Section 3.3.3.

Conservative gradients find other applications, justifying for instance parametric optimality for value functions [127], path differentiability of ODE flows [110], differentiation of iterative algorithms [38] and parametric solution of monotone inclusions [42].

Convergence of nonsmooth algorithms as implemented in practice. Thanks to their definition, conservative gradients allow for a convergence theory of first-order algorithms as implemented in practice. For instance, suppose we are minimizing a composition $F := F_1 \circ \ldots \circ F_r$. Applying the backpropagation on F is the application of the chain rule to Clarke derivatives and yields a conservative gradient D_F. The subgradient method with backpropagation can then be written as a set-valued recursion

$$w_{k+1} \in w_k - \alpha_k D_F(w_k),$$

where D_F is a conservative gradient for F. Thanks to the differential inclusion method (Section 1.3.4) this recursion can be studied through the conservative gradient flow $\dot{w} \in -D_F(w)$. By

the definition of the conservative gradient, F decreases along the curves of this flow hence leading to the convergence to critical points $\{w \,:\, 0 \in D_F(w)\}$. In other words, the analysis from Section 1.3.4 can be repeated replacing $\partial^c F$ by D_F.

The stochastic subgradient method with backpropagation and finite distribution is considered in [37]. For an empirical risk $F := \frac{1}{n}\sum_{i=1}^n f_i$, the output of backpropagation applied to a function f_i falls in a conservative gradient D_i for f_i. If now we sample uniformly from the backpropagation outputs of the functions $(f_i)_{i=1,\dots,n}$, it will approximate an element from $D_F := \frac{1}{n}\sum_{i=1}^n D_i$ which is a conservative gradient for F. The stochastic subgradient method with backpropagation can then be written

$$w_{k+1} \in w_k - \alpha_k(D_F(w_k) + \epsilon_k),$$

with a noise ϵ_k that is centered conditionally to w_k, and it can be studied with the differential inclusion method.

In this thesis, we extend this analysis to a general distribution by using a conservative integral rule that we show in Chapter 3 Section 3.2. As a result, we propose a general setting to analyze stochastic first-order methods as implemented in practice (Section 4.2). We apply the differential inclusion method to study the stochastic subgradient method and its heavy ball version.

Toward calculus-free guarantees. Using such types of operator-free oracles leads to the convergence to general critical points $\{\bar{w} \,:\, 0 \in D_F(\bar{w})\}$. Such a convergence suggests that using calculus rules generates artifacts that may impact the convergence of the method. For instance, in the case of automatic differentiation, the authors of [36] highlighted that non-uniqueness of implementations could lead to artificial critical points having absurd locations. Consequently, one may seek to certify that the calculus rules, used to define a conservative gradient D_F, doesn't impact the convergence, which can be expressed by the convergence to Clarke critical points $\{w : 0 \in \partial^c F(w)\}$.

Both semismooth and conservative gradients have good variational properties as they are gradients almost everywhere, see [142, Appendix A] or [114] for semismooth derivatives and [37, Theorem 1] for conservative gradients. A simple solution proposed in [121] to retrieve the convergence to Clarke critical points is to add a uniform noise at each step.

Another axis of research, initiated by [28,36], aims to certify this convergence without modifying the method. It is shown in [36] that when the objective function is a finite sum and definable, randomizing initialization and avoiding a finite set of stepsizes is sufficient to avoid all artifacts. This question was also studied in [28] in the case of the stochastic subgradient method with a constant stepsize. The authors of [28] showed for samples from a general probability space, that the avoidance happens for a randomized initialization and whenever the stepsizes avoid a zero Lebesgue measure set.

In this thesis, Chapter 4 Section 4.3, we study this question in the case of the stochastic subgradient method and the heavy ball version. For the stochastic subgradient method, we consider the case of an absolutely continuous distribution and derive a sharper characterization of the initialization set than [28].

1.5 Thesis outline and contributions

In Chapter 2, we recall notions from set-valued analysis. Indeed, set-valued maps are recurrent in this work, and integration and measurability notions have to be defined for such objects. In the same chapter, we present the o-minimal structures, which are central to this thesis. Some results

are proved and may be of independent interest such as an integration result for definable set-valued maps and a Fubini-type lemma for dense definable sets.

In view of studying algorithms with ODE methods [21, 29] we are interested in the model of conservative gradients which we present in Chapter 3. We then propose two extensions of the conservative calculus: a nonsmooth implicit differentiation formula, and a nonsmooth integral rule. We apply our nonsmooth implicit differentiation formula in order to justify first-order optimization methods on bi-level problems, e.g. hyperparameter optimization, and implicit models. The motivation for the integral rule is twofold. First, it allows to obtain path differentiability for a general integral function, hence justifying a descent property for risk minimization outside the semialgebraic case. Secondly, it justifies nonsmooth first-order sampling in practical implementations which may use automatic differentiation.

At the end of Chapter 3, we study the relation between semismooth and conservative derivatives, and establish that semismooth derivatives are conservative derivatives in general. Comparative examples in one dimension are provided.

We use the calculus developed in Chapter 3 in Chapter 4 in order to propose a general nonsmooth stochastic setting that is compatible with calculus, e.g. automatic differentiation and nonsmooth implicit differentiation. A noticeable feature of our setting is to rely on an integration result of definable functions in order to obtain a Sard property for a wide family of absolutely continuous distributions. Based on path differentiability, our framework allows to use differential inclusion methods [21, 29] to obtain convergence results for the stochastic subgradient method and its heavy ball version.

At the end of Chapter 4, we study the question of artifacts avoidance. In the case of the stochastic subgradient method, we focus on the case of an absolutely continuous distribution. We study the question for the stochastic heavy ball method and complete previous analyses [29, 142] of this algorithm.

References. This thesis is based on the following articles:

- [35] Jérôme Bolte, Tam Le, Edouard Pauwels, Antonio Silveti-Falls, Nonsmooth implicit differentiation for machine learning and optimization, Neurips (2021).
- [34] Jérôme Bolte, Tam Le, Edouard Pauwels. Subgradient sampling for nonsmooth nonconvex minimization, SIAM Journal on Optimization (2023).
- [100] Tam Le, Nonsmooth nonconvex stochastic heavy ball, submitted (2023).

1.6 Notations

Throughout this thesis, we denote $\|\cdot\|$ the Euclidean norm. For a subset $A \subset \mathbb{R}^p$, $\operatorname{conv} A$ is its convex hull, $\overline{A}$ its closure, $\dim A$ is the Hausdorff dimension of A and we use the notation $\|A\| := \sup\{\|y\| \ : \ y \in A\}$ whenever it is defined. $B(x, r)$ denotes the open ball of center $x \in \mathbb{R}^p$ and radius $r \geq 0$. We denote $P_{\mathcal{C}}$ the projection onto a convex set $\mathcal{C}$. We denote $\operatorname{Graph} f$ the graph of a function f, and for a function f of two variables (w, s), we use the subset $\nabla_w f$ for the partial gradient of f with respect to w. The Jacobian of a function F is denoted $\operatorname{Jac} F$. The support of a measure μ is denoted $\operatorname{supp} \mu$, and the linear span of a subset A is Span. λ denotes the Lebesgue measure.

Chapter 2

Preliminaries on nonsmooth and definable optimization

In this chapter, we gather essential definitions for understanding set-valued maps and o-minimal structures. Some additional results are proved and can be of independent interest, such as the definability of set-valued integrals and a Fubini-type lemma for dense definable sets.

2.1 Digest of set-valued and variational analysis

2.1.1 Set-valued maps

Set-valued maps are recurrent in this thesis. We use the notation $D : A \rightrightarrows B$ to define a set-valued map D which maps an element from A to a subset of B. We summarize in this part the notions of measurability, integration, and regularity for such objects. These are taken from the references [5, 10, 11]. Throughout this subsection, $(X, \mathcal{F})$ is a measurable space.

While the measurability of set-valued maps is not as straightforward as that of single-valued ones (see, e.g., [5, Chapter 18]), in the specific case of nonempty compact-valued maps, measurability can be understood in relation to the Hausdorff topology.

Definition 2.1 (Measurable set-valued maps)**.** *Denote $\mathcal{K}_p$ the set of nonempty compact subsets of $\mathbb{R}^p$. It is a measurable space considering the Borel σ-algebra $\mathcal{B}_H(\mathcal{K}_p)$ induced by the topology of the Hausdorff distance. A nonempty compact valued map $F : X \rightrightarrows \mathbb{R}^p$ is called* measurable*, if it is measurable from $(X, \mathcal{F})$ to $(\mathcal{K}_p, \mathcal{B}_H(\mathcal{K}_p))$.*

In this case, for all closed subsets $A \subset \mathbb{R}^p$, the upper inverse $F^u(A) := \{x \in X \ : \ F(x) \subset A\}$ *is measurable in $(X, \mathcal{F})$.*

In view of manipulating integrals of set-valued map, a fundamental result gives the existence of measurable selections:

Proposition 2.2 (Measurable selections, Theorem 18.13 [5])**.** *Let $F : X \rightrightarrows \mathbb{R}^p$ be a measurable nonempty and compact valued map. Then there exists a measurable selection of F, that is a measurable function $v : X \to \mathbb{R}^p$ satisfying for all $x \in X$, $v(x) \in F(x)$.*

Corollary 2.3 (Castaing's Theorem)**.** *Let $F : X \rightrightarrows \mathbb{R}^p$ be nonempty compact valued. Then F is measurable if and only if there exists a sequence of measurable selections $(F_n)_{n\in\mathbb{N}}$ such that $\forall x \in X, F(x) = \overline{\{F_1(x), F_2(x), ...\}}$.*

Remark 2.4 (Measurability of set-valued composition)**.** Corollary 2.3 can justify measurability of composed set-valued functions. For instance, given $g : \mathbb{R}^p \to \mathbb{R}$ continuous and $F : X \rightrightarrows \mathbb{R}^p$ compact valued and measurable, then $g \circ F$ is measurable. Indeed, let $(F_k)_{k\in\mathbb{N}}$ be a sequence of measurable selections given by Castaing's Theorem, such that $\forall x \in X, F(s) = \overline{\{F_1(x), F_2(x), ...\}}$. Then by continuity of g, we have for all $x \in X$, $g(F(x)) = g(\overline{\{F_1(x), F_2(x), ...\}}) = \overline{\{g(F_1(x)), g(F_2(x)), ...\}}$. The functions $g \circ F_i$ are all measurable and $g \circ F$ is compact valued by continuity of g, whence by Castaing's Theorem, $g \circ F$ is measurable.

Finally, the integral of a set-valued map is defined as the set of integrals over integrable selections:

Definition 2.5 (Aumann integral)**.** Let $(X, \mathcal{F}, \mu)$ be a measure space and $F : X \rightrightarrows \mathbb{R}^p$ a set-valued map. Then the *integral* of F with respect to the measure μ is

$$\int_X F(x)\,\mathrm{d}\mu(x) = \left\{ \int_X v(x)\,\mathrm{d}\mu(x) \;:\; v \text{ is integrable and for all } x \in X, v(x) \in F(x) \right\}.$$

We also use the expectation notation $\mathbb{E}_{\xi\sim P}[F(\xi)] = \int_X F(x)\,\mathrm{d}P(x)$ whenever $(X, \mathcal{F}, P)$ is a probability space and ξ is a random variable with distribution P.

Graph-closed and locally bounded set-valued maps are recurrent objects in this thesis. As we will see in Chapter 4, they allow defining set-valued recursions with continuous-time counterparts. They are thus essential to study nonsmooth first-order algorithms. A graph-closed, nonempty, and compact valued map also satisfies the property to be upper semicontinuous, see e.g., [10].

Definition 2.6. Let $F : \mathbb{R}^m \rightrightarrows \mathbb{R}^p$ be a set-valued map.

1. *(graph-closedness) F is* graph-closed *if its graph defined by*

$$\operatorname{Graph} F := \{(x, y) \in \mathbb{R}^m \times \mathbb{R}^p \;:\; y \in F(x)\},$$

   is a closed subset of $\mathbb{R}^m \times \mathbb{R}^p$.

2. *(Local boundedness) F is* locally bounded *if for all $x \in \mathbb{R}^m$, there exist a neighborhood $\mathcal{U}$ of x and $M > 0$, such that for all $z \in U$ and $y \in F(z)$, $\|y\| < M$.*

3. *(Upper semicontinuity) F is* upper semicontinuous *at $x \in \mathbb{R}^m$, if for each open subset $\mathcal{V}$ containing $F(x)$, there exists a neighborhood $\mathcal{U}$ of x such that for all $z \in \mathcal{U}$, $F(z) \subset \mathcal{V}$.*

2.1.2 The Clarke subdifferential

The *Clarke subdifferential* [54] is a classical tool to define the derivative of nonsmooth functions. It is defined and nonempty valued for all locally Lipschitz function.

Definition 2.7 (Clarke subgradient and Jacobian)**.** *Let $f : \mathbb{R}^p \to \mathbb{R}$ be a locally Lipschitz function. In virtue of Rademacher's theorem, f is differentiable almost everywhere. If diff_f is the differentiability domain of f, the Clarke subgradient is defined for all $x \in \mathbb{R}^p$ as*

$$\partial^c f(x) = \operatorname{conv}\{\lim_{k\to\infty} \nabla f(x_k) \;:\; x_k \in \operatorname{diff}_f, x_k \underset{k\to\infty}{\to} x\}.$$

This definition extends to a multivariate map $F : \mathbb{R}^p \to \mathbb{R}^m$. The Clarke Jacobian of F is defined as

$$\operatorname{Jac}^c F(x) = \operatorname{conv}\{\lim_{k\to\infty} \operatorname{Jac} F(x_k) \;:\; x_k \in \operatorname{diff}_F, x_k \underset{k\to\infty}{\to} x\}.$$

For f and F as in the above definition, the Clarke subgradient and Jacobian satisfy the following:

- $\partial^c f$, $\mathrm{Jac}^c F$ are graph-closed, nonempty, convex and compact valued. In particular, they are upper semicontinuous.
- [159] For any full Lebesgue measure subset $\Delta \subset \mathbb{R}^p$ on which f is differentiable,

$$\partial^c f(x) = \mathrm{conv}\{\lim_{k\to\infty} \nabla f(x_k) \ : \ x_k \in \Delta, x_k \underset{k\to\infty}{\to} x\}.$$

  Alternatively, the definition of $\partial^c f$ doesn't depend on the choice of the set of differentiability. This also holds for the Clarke Jacobian.

2.2 Semialgebraic sets and o-minimal structures

Considering semialgebraic and definable sets in o-minimal structures aims to include a wide class of sets and functions used in practice, yet having a simple geometric structure, therefore, excluding pathological behaviors. This section serves as a condensed introduction to this theory, based on the more complete references [59, 156]. The main consequences of this theory in optimization will be emphasized such as stratification, path differentiability, and Sard's theorem.

2.2.1 Definition and elementary properties

Semialgebraic sets are defined as finite unions and intersections of polynomial inequalities and equalities:

Definition 2.8 (Semialgebraic sets)**.** A subset $A \subset \mathbb{R}^n$ is *semialgebraic* if there exist polynomial functions P_{ij} and Q_{ij} with $i = 1, \dots, l$ and $j = 1, \dots, k$ such that $A = \bigcup_{i=1}^{l} \bigcap_{j=1}^{k} \{x \in \mathbb{R}^n \ : \ P_{ij}(x) < 0, \ Q_{ij}(x) = 0\}$.

This definition naturally extends to functions by considering their graphs: a function $f : \mathbb{R}^p \to \mathbb{R}^q$ or a set-valued map $G : \mathbb{R}^p \rightrightarrows \mathbb{R}^q$ are called semialgebraic if their graphs, as sets of $\mathbb{R}^{p+q}$ are semialgebraic.

The class of semialgebraic functions includes most functions used in machine learning. The ReLU function $x \mapsto \max(x, 0)$, the MaxPooling, the square loss, and ℓ_1 norm are semialgebraic. Semialgebraic sets satisfy several stability properties. First, they are stable by finite intersection, union and complementation. Also, given a semi algebraic set $A \subset \mathbb{R}^m$, $A \times \mathbb{R}^p$ and $\mathbb{R}^p \times A$ are semialgebraic. Then, by Tarski-Seidenberg theorem, the projection of a semialgebraic set is semialgebraic. These stability properties are axiomatized to define o-minimal structures [156]:

Definition 2.9 (o-minimal structure)**.** Let $\mathcal{O} = (\mathcal{O}_p)_{p\in\mathbb{N}}$ be a collection of sets such that, for all $p \in \mathbb{N}$, $\mathcal{O}_p$ is a set of subsets of $\mathbb{R}^p$. $\mathcal{O}$ is an o-minimal structure on $(\mathbb{R}, +, \cdot)$ if it satisfies the following axioms, for all $p \in \mathbb{N}$:

1. $\mathcal{O}_p$ is stable by finite intersection, union, and complementation, and contains $\mathbb{R}^p$.
2. If $A \in \mathcal{O}_p$ then $A \times \mathbb{R}$ and $\mathbb{R} \times A$ belong to $\mathcal{O}_{p+1}$.
3. If $A \in \mathcal{O}_{p+1}$ then $\mathcal{P}_p(A) \in \mathcal{O}_p$, where $\mathcal{P}_p$ is the projection onto the p first coordinates,.
4. $\mathcal{O}_p$ contains all sets of the form $\{x \in \mathbb{R}^p : P(x) = 0\}$, where P is a polynomial.
5. The elements of $\mathcal{O}_1$ are exactly the finite unions of intervals.

Definition 2.10 (Definable set and function)**.** *A set $A \subset \mathbb{R}^m$ is called* definable *in an o-minimal structure $\mathcal{O} = (\mathcal{O}_p)_{p\in\mathbb{N}}$, if $A \in \mathcal{O}_m$. A function $f : \mathbb{R}^m \to \mathbb{R}^n$ is called definable in $\mathcal{O}$ if* Graph $f \in \mathcal{O}_{m+n}$.

> With an abuse of terminology, and when there is no confusion, we will often call a set or function definable, without specifying the underlying o-minimal structure which is assumed to be the same for all objects. The reader can also consider "definable" to be "semialgebraic" for simplicity.

In Definition 2.9, the property 3 corresponds to the Tarski-Seidenberg theorem, which allows to "eliminate" quantifiers $\exists$ and by $\forall$. For instance, suppose a $A \subset \mathbb{R}^p \times \mathbb{R}^q$ is definable in an o-minimal structure $\mathcal{O}$, then the projection of A onto the p first components, written $\{x \in \mathbb{R}^p : \exists y \in \mathbb{R}^q, (x, y) \in A\}$, is also definable in $\mathcal{O}$.

As a result, these axioms imply that many properties, defined with quantifiers and logical operators are definable. First, one can easily verify that definable functions are stable by composition, finite sum, and product.

Then, the interior and the closure of a definable set remain definable. For a definable function $f : \mathbb{R}^p \to \mathbb{R}$, the differentiability domain of f is definable, and its gradient ∇f is also definable. Furthermore, the Clarke subgradient of a locally Lipschitz definable function is definable in the same structure. By extension, the Hessian of a definable function is also definable.

We may recall the following property which is the definable counterpart of the measurable selection theorem.

Proposition 2.11 (Definable choice [59])**.** *Let $A \subset \mathbb{R}^p \times \mathbb{R}^q$ be a definable set. Denote $\mathcal{P}_p$ the projection on the p first coordinates. Then there exists a definable function $h : \mathcal{P}_p A \to \mathbb{R}^q$ such that for all $x \in \mathcal{P}_p A$, $(x, h(x)) \in A$.*

O-minimal structures are however not closed under some operations. For instance, given a sequence of semialgebraic functions uniformly converging to some function f, f may not be definable, even in a larger structure than semialgebraic functions. In particular, a parameterized integral over a definable function may not be definable without any assumption on the integrating measure.

2.2.2 Main o-minimal structures

The class of semialgebraic sets is the smallest o-minimal structure and may not contain some usual functions such as the sinus and cosinus, the exponential, or the logarithm. In order to include these functions, larger o-minimal structures may be considered. We present here two main o-minimal structures, called $\mathbb{R}_{\text{an}}$ or the globally subanalytic sets, and $\mathbb{R}_{\text{an,exp}}$.

As for the semialgebraic sets, all the definitions related to sets extend naturally to the graphs of the functions. For instance, globally subanalytic functions, are functions such that their graphs are globally subanalytic. In order to lighten the writing, the term "function" may refer to the graph of the function when employed in the context of sets. For instance, we may say that semialgebraic sets contain polynomial functions, as they contain the graphs of polynomial functions.

O-minimal structure $\mathbb{R}_{\text{an}}$. The structure of *globally subanalytic* sets, denoted $\mathbb{R}_{\text{an}}$, contains a large class of analytic functions.

In order to define this structure, we may first extend semialgebraicity to real analytic functions.

Definition 2.12 (Semianalytic sets [156])**.** *A subset A of $\mathbb{R}^n$ is semianalytic if for any point $x \in \mathbb{R}^d$, there exists a neighborhood U of x such that $U \cap A$ has the form $\bigcup_{i=1}^{l} \bigcap_{j=1}^{k} \{x \in \mathbb{R}^n : g_{ij}(x) < 0,\ h_{ij}(x) = 0$, where the g_{ij} and h_{ij} are real analytic.*

The definition slightly differs from semialgebraic sets but appears to be well-posed. For instance, it ensures semianalytic sets to be stable by closure. The function $x \mapsto \exp(-1/x^2)$ is for instance not semianalytic, but could be expressed in terms of analytic inequalities and equalities.

Semianalytic Tarski-Seidenberg theorem doesn't hold. The subanalytic sets extend the class of semianalytic sets to ensure stability by projection.

Definition 2.13 (Subanalytic sets)**.** *A subset A of $\mathbb{R}^n$ is* subanalytic *if there exists $m \in \mathbb{N}$ such that A is the projection of a semianalytic set $M \subset \mathbb{R}^{n+m}$ on $\mathbb{R}^n$.*

One can finally define globally subanalytic sets:

Definition 2.14 (Globally subanalytic sets)**.** *$A \subset \mathbb{R}^n$ is globally subanalytic if $\tau_n(A)$ is subanalytic, where $\tau_n : x \mapsto \left(\frac{x_1}{\sqrt{1+x_1^2}}, \dots, \frac{x_n}{\sqrt{1+x_n^2}} \right)$.*

This definition, where τ_n is an analytic diffeomorphism, aims to exclude non-analytic behavior which could happen at infinity and which could break stability by composition. For instance, the sinus function or the exponential function are not allowed in this structure, since their composition with $x \mapsto -1/x$ is not subanalytic.

Examples of globally subanalytic functions, besides semialgebraic functions, are power functions with rational exponents or analytic functions restricted to semialgebraic compact sets.

O-minimal structure $\mathbb{R}_{\text{an,exp}}$. The exponential function, yet widely used in machine learning and statistics, is not globally subanalytic. It is however definable in an o-minimal structure called $\mathbb{R}_{\text{an,exp}}$ which also contains $\mathbb{R}_{\text{an}}$, see [67]. Remark that this structure contains the logarithm function and also all power functions.

2.2.3 Some examples in machine learning

The o-minimal structures presented above contain most functions used in machine learning. As an illustration, we provide a non-exhaustive list of definable functions used in machine learning.

First, the ℓ_p norms where $p \geq 1$ is rational are semialgebraic. The maximum of a vector's components, which can be defined in terms of affine inequalities and affine functions, is semialgebraic. This implies for instance that the ReLU function and the ℓ_∞ norm are semialgebraic.

The arctan function is globally subanalytic. As to include other analytic functions in general, any restriction of an analytic function to a compact semialgebraic set is globally subanalytic. For instance, sinus and cosinus restricted to compact intervals are globally subanalytic.

When using the exponential function or the logarithm on their full domain, one may consider the structure $\mathbb{R}_{\text{an,exp}}$. This applies to several activation functions in deep learning such as

- Sigmoid, $x \mapsto \frac{1}{1+\exp(-x)}$,
- Softplus, $x \mapsto \log(1 + \exp(x))$,
- Softmax, $x \in \mathbb{R}^I \mapsto \frac{\exp(x)}{\sum_{i=1}^I \exp(-x_i)}$,

where the operations on $x \in \mathbb{R}^I$ are defined component-wise.

Several usual density functions are definable. For instance, the Gaussian, Laplace, and Gamma distributions are definable in $\mathbb{R}_{\text{an,exp}}$ but not globally subanalytic due to the presence of the exponential function. However, their truncated version, for instance the truncated Gaussian distribution, is globally subanalytic. The Student's t density, with rational coefficient ν, $\phi(s) \propto (1 + \frac{s^{-2}}{\nu})^{-\frac{\nu+1}{2}}$, is globally subanalytic.

2.2.4 The stratification property

The rigidity and the simple geometry of definable sets can be viewed in terms of the stratification into C^r manifolds.

Definition 2.15 (Stratification [156])**.** *Let $A \subset \mathbb{R}^p$ be definable. There exists a partition $(M_k)_{k=1,\dots,q}$ of A such that for $k = 1, \dots, q$, M_k is a C^r-manifold and for all couple i, j, $\overline{M_i} \cap M_j \neq \emptyset \Rightarrow M_j \subset \overline{M_i}$. Such a partition is called C^r-stratification.*

A stratification can be understood as a partition ordered by dimension, and such that lower dimension manifolds are included in the boundaries of higher dimension manifolds. For instance, a stratification of a closed cube is given by its interior cube, the faces, the edges, and vertices. For a definable function $F : \mathbb{R}^p \to \mathbb{R}$ and $r \geq 0$ one may consider a stratification $(M_i')_{i=1,\dots n'}$ of $\mathbb{R}^p$ for which for all $i = 1, \dots, n'$, the restriction of F to M_i' is C^r. Taking the union of the manifolds M_i' of dimension p gives a dense open subset in $\mathbb{R}^p$ on which F is C^r. Take for instance the graph of the absolute value $|\cdot|$, which is a semialgebraic set. It admits as a stratification $\{M_1, M_2, M_3\}$ where $M_1 = \{(x, -x) \ : \ x \in]-\infty, 0[\}$, $M_2 = \{(x, x) \ : \ x \in]0, +\infty[\}$, and $M_3 = \{(0, 0)\}$.

Definable sets actually admit a stronger type of stratification called *Whitney stratification*. For a differentiable manifold M, we denote $T_x M$ the tangent space of M at x.

Definition 2.16 (Whitney stratification)**.** *Let $A \subset \mathbb{R}^p$ be definable. A admits a C^1-stratification $(M_k)_{k=1,\dots,q}$ which satisfies the following:*

For $i \neq j$, for any $x \in \overline{M_i} \cap M_j$ and for all sequences $(x_k)_{k\in\mathbb{N}}$ included in M_i, we have

$$\left.\begin{array}{c} \lim_{k\to\infty} x_k = x \\ \lim_{k\to\infty} T_{x_k} M_i = \mathcal{T} \end{array}\right\} \Rightarrow T_x M_j \subset \mathcal{T}$$

where the limit is understood in the Grassmannian sense. Such a stratification is called Whitney stratification*.*

In a variational sense, this property states that nondifferentiability sets of definable functions are structured into smooth manifolds whose connections are well-behaved. In particular, this property leads to a projection formula for the Clarke subgradient [40] with important consequences in optimization such as path differentiability and a definable Sard's theorem. These aspects will be exposed in Chapter 3 in the context of conservative gradients.

2.2.5 Definability and integration

Definability of parameterized integrals. Definable functions are stable under usual algebraic operations such as composition, addition, or product. As mentioned earlier, this concept does not easily encompass functions expressed as integrals. Yet, minimizing an expectation is prevalent in machine learning but there is no evident way to ensure a proper setting leading to a convergence theory, in particular a Sard property.

When the expectation is taken with respect to a finitely discrete distribution, we may assume the integrand $f(w, \xi)$ to be definable with respect to the parameter w, in which case the definability and then the Sard property is obtained by the stability of definable functions by sum.

For general distribution, the question of stability is open. However, in the specific case of absolutely continuous distributions, [56] gives the stability of *constructible* functions under integration. Constructible functions are products of globally subanalytic functions and logarithms of globally subanalytic functions. In particular, these functions are definable in the structure $\mathbb{R}_{\text{an,exp}}$. We may formulate this result in terms of the structures $\mathbb{R}_{\text{an}}$ and $\mathbb{R}_{\text{an,exp}}$.

Theorem 2.17 (from [56, Theorem 1.3]). *Let $g : \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}^p$ be globally subanalytic. Assume for all $w \in \mathbb{R}^p$, $g(w, \cdot)$ is integrable with respect to Lebesgue. Then $F : w \in \mathbb{R}^p \mapsto \int_{\mathbb{R}^m} g(w, s)\,\mathrm{d}s$ is definable in $\mathbb{R}_{\mathrm{an,exp}}$.*

This theorem can also be generalized to set-valued maps thanks to the representation of convex sets into support functions:

Theorem 2.18 (Definable set-valued integrals). *Let $\phi : \mathbb{R}^m \to \mathbb{R}$ be a globally subanalytic density function, $D : \mathbb{R}^p \times \mathbb{R}^m \rightrightarrows \mathbb{R}^p$ globally subanalytic, graph-closed, locally bounded and convex-valued. Assume $D_F : w \mapsto \int_{\mathbb{R}^m} D(w, s)\phi(s)\,\mathrm{d}s$ is well defined, then D_F is definable in $\mathbb{R}_{\mathrm{an,exp}}$.*

Proof: Let D be as in the theorem. We will apply Theorem 2.17 to prove the definability of D_F in $\mathbb{R}_{\mathrm{an,exp}}$. D is definable in $\mathbb{R}_{\mathrm{an}}$ hence the set-valued map $(w, q, s) \mapsto \langle D(w, s), q\rangle$ is definable in $\mathbb{R}_{\mathrm{an}}$ as well. Let G be its graph. The function $H : (w, q, s) \mapsto \max_{v \in D(w,s)} \langle v, q\rangle$ is clearly definable in $\mathbb{R}_{\mathrm{an}}$ as its graph writes

$$\operatorname{Graph} H = \left\{(w, q, s, y) \in G \ : \ \forall (w', q', s', y') \in G, (w', q', s') = (w, q, s) \implies y \geq y'\right\}.$$

By definition of the Aumann integral, we have for all $w \in \mathbb{R}^p$

$$D_F(w) = \left\{ \int_{\mathbb{R}^m} g(s)\phi(s)\,\mathrm{d}s \ : \ g \text{ is a measurable selection of } D(w, \cdot) \right\},$$

For $C \subset \mathbb{R}^p$ compact convex, define the support function $h_C : q \mapsto \max_{v \in C} \langle v, q\rangle$. Then by linearity of the integral, for $(w, q) \in \mathbb{R}^p \times \mathbb{R}^p$, it holds that

$$\begin{aligned} h_{D_F(w)}(q) &= \max_{v \in D_F(w)} \langle v, q\rangle \\ &= \max \left\{ \int_{\mathbb{R}^m} \langle g(s), q\rangle \phi(s)\,\mathrm{d}s \ : \ g \text{ is a measurable selection of } D(w, \cdot) \right\}. \end{aligned}$$

By [5, Theorem 18.19], there exists a measurable selection $\tilde{g} : \mathbb{R}^m \to \mathbb{R}^p$ of $D(w, \cdot)$ such that $\forall s \in \mathbb{R}^m, \langle \tilde{g}(s), q\rangle = \max_{v \in D(w,s)} \langle v, q\rangle = H(w, q, s)$, and thus, $\tilde{g}$ achieves the maximum $h_{D_F(w)}(q)$, i.e., $h_{D_F(w)}(q) = \int_{\mathbb{R}^m} H(w, q, s)\phi(s)\,\mathrm{d}s$. Furthermore, by duality, for all convex compact set $C \subset \mathbb{R}^p$ it holds that $C = \{z \in \mathbb{R}^p \ : \ \sup_{v \in \mathbb{R}^p} \langle v, z\rangle - h_C(v) = 0\}$. Applying this property with $C = D_F(w)$ for each $w \in \mathbb{R}^p$ gives

$$\begin{aligned} \operatorname{Graph} D_F &= \{(w, z) \in \mathbb{R}^p \times \mathbb{R}^p \ : \ z \in D_F(w)\} \\ &= \left\{ (w, z) \in \mathbb{R}^p \times \mathbb{R}^p \ : \ \sup_{q \in \mathbb{R}^p} \langle q, z\rangle - h_{D_F(w)}(q) = 0 \right\} \\ &= \left\{ (w, z) \in \mathbb{R}^p \times \mathbb{R}^p \ : \ \sup_{q \in \mathbb{R}^p} \langle q, z\rangle - \int_{\mathbb{R}^m} H(w, q, s)\phi(s)\,\mathrm{d}s = 0 \right\}. \end{aligned} \tag{2.1}$$

By Theorem 2.17, $(w, q) \in \mathbb{R}^p \times \mathbb{R}^p \mapsto \int_{\mathbb{R}^m} H(w, q, s)\phi(s)\,\mathrm{d}s$ is definable in $\mathbb{R}_{\mathrm{an,exp}}$, hence by equality (2.1) D_F is also definable in $\mathbb{R}_{\mathrm{an,exp}}$. □

Definable dense sets. The stratification property excludes many pathological sets. In particular, fractal and "dust" phenomena can't happen. A definable subset of $\mathbb{R}^p$ with full Lebesgue measure, i.e. which complement has zero measure, has a dense interior. A definable zero Lebesgue measure subset from $\mathbb{R}$ is necessarily finite.

For measurable sets, $L \subset \mathbb{R}^r \times \mathbb{R}^q$, Fubini's theorem implies that L has full Lebesgue measure if and only if the slices $L_w = \{s \in \mathbb{R}^q \ : \ (w, s) \in L\}$ have full Lebesgue measure for w chosen in some set of full Lebesgue measure. This can be adapted for dense definable sets:

Lemma 2.19 (Definable Fubini's type lemma)**.** *Let $(r,q) \in \mathbb{N}^* \times \mathbb{N}^*$, and $L \subset \mathbb{R}^r \times \mathbb{R}^q$ be a definable set. Then L is dense if and only if there is a dense definable set $Z \subset \mathbb{R}^r$ such that for all $w \in Z$, $L_w := \{s \in \mathbb{R}^q \ : \ (w,s) \in L\}$ is dense definable in $\mathbb{R}^q$.*

Proof: Let us start with the direct implication. Set $Z = \{w \in \mathbb{R}^r \ : \ \forall z \in \mathbb{R}^q, \forall \epsilon > 0, \exists s \in \mathbb{R}^q, (w,s) \in L, \|s-z\| < \epsilon\}$. This set is definable and is precisely the set of w such that L_w is dense in $\mathbb{R}^q$. Assume that Z^c has a nonempty interior. This means that there is a nonempty open set $U \subset \mathbb{R}^r$, such that for all $w \in U$

$$\exists z \in \mathbb{R}^q, \exists \epsilon > 0, \forall s \in \mathbb{R}^q, (w,s) \in L \Rightarrow \|s-z\| \geq \epsilon.$$

By definable choice, see Proposition 2.11, there are definable functions $z \colon U \to \mathbb{R}^q$ and $\epsilon \colon U \to \mathbb{R}_+^*$, such that for all $w \in U$, we have $\{(w,v) \in U \times \mathbb{R}^q \ : \ \|v - z(w)\| < \epsilon(w)\} \subset L^c$. By stratification, see Definition 2.15, reducing U if needed, z and ϵ can be chosen continuous hence L^c has nonempty interior which contradicts the density of L.

As for the reverse implication, fix any $(\bar{w}, \bar{s}) \in \mathbb{R}^r \times \mathbb{R}^q$ and $\epsilon > 0$. Since Z is dense there is $w \in Z$ such that $\|w - \bar{w}\| < \epsilon/\sqrt{2}$. Since L_w is dense, there is $s \in L_w$ such that $\|s - \bar{s}\| < \epsilon/\sqrt{2}$. Overall, we have $(w,s) \in L$ such that $\|(w,s) - (\bar{w},\bar{s})\| < \epsilon$ which shows that L is dense as $(\bar{w},\bar{s}) \in \mathbb{R}^r \times \mathbb{R}^q$ and $\epsilon > 0$ were arbitrary. $\square$

Chapter 3

Nonsmooth calculus with conservative derivatives

In order to obtain the Jacobian of the composition of two differentiable functions, one may use the chain rule formula. As to the sum of two differentiable functions, the sum of the gradients provides the gradient of the sum. These simple calculus rules don't hold with Clarke derivatives. In the case of locally Lipschitz functions, one has the following one-sided inclusions [54]:

- (Composition rule, or chain rule) For $F : \mathbb{R}^q \to \mathbb{R}^r$ and $G : \mathbb{R}^p \to \mathbb{R}^q$ locally Lipschitz, $\mathrm{Jac}^c(F \circ G) \subset \mathrm{conv}\, \mathrm{Jac}^c F(G)\, \mathrm{Jac}^c G$,
- (Sum rule) For $f : \mathbb{R}^p \to \mathbb{R}$ and $g : \mathbb{R}^p \to \mathbb{R}$ locally Lipschitz, $\partial^c(f+g) \subset \partial^c f + \partial^c g$.

In general, the inclusions are strict, and the usual conditions to obtain equalities are convexity or Clarke regularity [54] which are often too restrictive for machine learning applications. The equality doesn't hold even in simple cases. Consider for instance $f = g = \mathrm{relu}$, then the difference $f - g$ is the zero function, with derivative equal to 0 everywhere. However, the difference of the Clarke subderivatives at 0 is $\partial^c f(0) - \partial^c g(0) = [0,1] - [0,1] = [-1,1] \neq 0$.

In this chapter, we focus on a variational model called *conservative gradients and Jacobians* introduced in [37]. It allows extending calculus rules, such as the composition and the sum, to a general nonsmooth nonconvex setting. The main advantage of this theory is that it doesn't require verifying qualification conditions, hence justifying faithfully the formal use of the Clarke subdifferential as done in practice. Another advantage is that its definition is based on a chain rule along curves, which aligns with the differential inclusion method to be presented in Chapter 4. This provides a very flexible framework for the analysis of nonsmooth first-order methods. Conservative derivatives theory finds many other applications mentioned in the introduction.

We propose two extensions of the conservative calculus: an integral rule, motivated by the use of first-order sampling in machine learning, and a nonsmooth implicit differentiation formula with some applications to bi-level problems and implicit models. Some connections with Norkin's semismooth derivatives [120] are established at the end of this chapter.

3.1 Conservative gradients and Jacobians

3.1.1 Definition and first calculus properties

Conservative derivatives are set-valued maps satisfying a chain rule along absolutely continuous curves. This can be seen as an axiomatization of non-pathological functions [155], but also as that

of the zero circulation property, see [37].

Definition 3.1 (Conservative gradient [37])**.** *Let $f : \mathbb{R}^p \to \mathbb{R}$ be a locally Lipschitz function. A locally bounded and graph-closed set-valued map $D : \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ that is nonempty valued is called* a conservative gradient *for f, if it satisfies one of the following equivalent properties*

- *(Chain rule along curves) For all absolutely continuous curve $\gamma : [0,1] \to \mathbb{R}^p$, f admits a chain rule with respect to D along γ,*

$$\frac{\mathrm{d}(f \circ \gamma)}{\mathrm{d}t}(t) = \langle v, \dot{\gamma}(t) \rangle, \text{ for all } v \in D(\gamma(t))$$

  for almost all $t \in [0,1]$.

- *(Integral form) For $\gamma : [0,1] \to \mathbb{R}^p$ absolutely continuous and for all measurable selection v of D,*

$$f(\gamma(1)) - f(\gamma(0)) = \int_0^1 \langle v(\gamma(t)), \dot{\gamma}(t) \rangle \, \mathrm{d}t.$$

Conservative gradients may be taken convex-valued: if D is a conservative gradient, $\operatorname{conv} D$ is also a conservative gradient. The chain rule along curves may be extended to multivariate maps in order to define conservative Jacobians:

Definition 3.2 (Conservative Jacobian [37])**.** *Let $F : \mathbb{R}^p \to \mathbb{R}^m$ be locally Lipschitz. A locally bounded and graph-closed set-valued map $J : \mathbb{R}^p \rightrightarrows \mathbb{R}^{m \times p}$ that is nonempty valued is called* a conservative Jacobian *for F, if for all absolutely continuous curve $\gamma : [0,1] \to \mathbb{R}^p$,*

$$\frac{\mathrm{d}(F \circ \gamma)}{\mathrm{d}t}(t) = V\dot{\gamma}(t) \text{ for all } V \in J(\gamma(t))$$

for almost all $t \in [0,1]$.

This model is centered around the oracle but yields a regularity notion, which is path differentiability, or chain rule along curves in [63], or non-pathological function [155].

Definition 3.3 (Path differentiable function)**.** *A locally Lipschitz function $f : \mathbb{R}^p \to \mathbb{R}$ is called* path differentiable *if it satisfies the following equivalent properties:*

- *there exists a conservative gradient for f,*
- *$\partial^c f$ is a conservative gradient.*

The definition extends to multivariate functions by considering conservative Jacobians.

Elementary path differentiable functions: Path differentiability holds on simple functions:

- [48] A convex or concave function is path differentiable.
- [62] A semialgebraic or definable function is path differentiable.

For path differentiable functions, the Clarke subderivatives are conservative derivatives. Finally, most machine learning functions are path differentiable. For instance, in deep learning, ReLU and MaxPooling are path differentiable. We may then compose conservative derivatives with the following rules:

Proposition 3.4 (Composition rule [37, Lemma 5]). *For $F : \mathbb{R}^q \to \mathbb{R}^r$ and $G : \mathbb{R}^p \to \mathbb{R}^q$ path differentiable with conservative Jacobians J_F and J_G respectively, $J_F(G)J_G$ is a conservative Jacobian for $F \circ G$.*

Corollary 3.5 (Sum rule [37, Corollary 4]). *For $f : \mathbb{R}^p \to \mathbb{R}$ and $g : \mathbb{R}^p \to \mathbb{R}$ path differentiable with conservative gradients D_f and D_g respectively, $D_f + D_g$ is a conservative gradient for $f + g$.*

The composition rule justifies automatic differentiation, which is the application of the chain rule formula on Clarke subderivatives. For instance, applying automatic differentiation on a neural network written as the composition of ReLU and affine functions may not output an element of the Clarke Jacobian but a selection in a conservative Jacobian.

The sum rule justifies first-order sampling for finite distribution. Consider an empirical risk $F = \frac{1}{n}\sum_{i=1}^n f_i$ where the $f_i : \mathbb{R}^p \to \mathbb{R}$ are locally Lipschitz and definable, with conservative gradients D_i containing the automatic differentiation oracle. Then sampling D_{i_k} where i_k is uniformly sampled from $\{1, \dots, n\}$ is in average $\frac{1}{n}\sum_{i=1}^n D_i$, which is a conservative gradient for F.

3.1.2 Variational structure of conservative derivatives.

By construction, conservative gradients have a rich variational structure. Through the following results from [37], we highlight a primary property of conservative gradients which is to be gradient almost everywhere. In particular, the Clarke subgradient is a minimal convex-valued conservative gradient. Note that the following properties all derive from the chain rule property and don't require any geometric assumption such as semialgebraicity or definability.

Theorem 3.6 (Conservative gradients are gradient a.e. [37, Theorem 1]). *Let D be a conservative gradient for $f : \mathbb{R}^p \to \mathbb{R}$ locally Lipschitz, then $D = \{\nabla f\}$ Lebesgue almost everywhere.*

As mentioned earlier, in the case of path differentiable functions, the Clarke subgradient is a conservative gradient. Actually, it is the smallest convex-valued one:

Proposition 3.7 ([37, Corollary 1]). *Let $f : \mathcal{U} \to \mathbb{R}$ be locally Lipschitz and path differentiable where $\mathcal{U}$ is a nonempty open subset from $\mathbb{R}^p$. Then $\partial^c f$ is a conservative gradient and for any convex-valued conservative gradient D of f, $\partial^c f \subset D$.*

This may be extended to Clarke Jacobians as well:

Proposition 3.8. *Given a nonempty open subset $\mathcal{U}$ of $\mathbb{R}^n$ and $F : \mathcal{U} \subset \mathbb{R}^n \to \mathbb{R}^m$ locally Lipschitz, let J_F be a convex-valued conservative Jacobian for F. Then for almost all $x \in \mathcal{U}$, $J_F(x) = \{\operatorname{Jac} F\}$ and, for all $x \in \mathcal{U}$, $\operatorname{Jac}^c F(x) \subset J_F(x)$.*

Proof: Using [37, Lemma 4] for $i \in \{1, \dots, m\}$, $[J_F]_i$ is a conservative map for F_i on $\mathcal{U}$ and it is equal to ∇F_i on a set of full measure $S_i \subset \mathcal{U}$. Hence for all $x \in S := \bigcap_{i=1}^m S_i$, which is of full measure in $\mathcal{U}$, $J_F(x) = \operatorname{Jac} F(x)$. Since S has full measure within $\mathcal{U}$, [159] gives the representation

$$\operatorname{Jac}^c F(x) = \operatorname{conv}\left\{ \lim_{k\to+\infty} \operatorname{Jac} F(x_k) : x_k \in S, x_k \underset{k\to+\infty}{\to} x \right\}, \text{ for any } x \in \mathcal{U}.$$

But since J_F coincides with $\operatorname{Jac} F$ throughout S, we have

$$\operatorname{Jac}^c F(x) = \operatorname{conv}\left\{ \lim_{k\to+\infty} J_F(x_k) : x_k \in S, x_k \underset{k\to+\infty}{\to} x \right\}$$

for each $x \in \mathcal{U}$. Finally, by graph-closedness and convexity of J_F, we get for each $x \in \mathcal{U}$,

$$\operatorname{Jac}^c F(x) \subset \operatorname{conv}\left\{ J_F\left(\lim_{k\to+\infty} x_k\right) : x_k \in S, x_k \underset{k\to+\infty}{\to} x \right\} = J_F(x).$$

□

For a locally Lipschitz path differentiable function $f : \mathbb{R}^p \to \mathbb{R}$, note that a graph-closed compact valued map $D : \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ almost everywhere equal to ∇f is not necessarily a conservative gradient for f. Such a counterexample D may be provided by taking D equal to $\partial^c f$ everywhere outside the unit sphere, and D equal to $\partial^c f \cup \overline{B(0,1)}$ on the unit sphere. In this case, the chain rule property does not hold for absolutely continuous curves lying in the unit sphere.

In dimension $p = 1$ however, we have the following equivalence:

Proposition 3.9 (A characterization in dimension 1)**.** *Let $f : \mathbb{R} \to \mathbb{R}$ be locally Lipschitz. Then a graph-closed and locally bounded set-valued map $D : \mathbb{R} \rightrightarrows \mathbb{R}$ is a conservative gradient for f if and only if $D = f'$ Lebesgue almost everywhere.*

Proof: The direct implication is a property of conservative derivatives, Theorem 3.6.

For the converse implication, we recall a substitution formula for absolutely continuous functions [147, Corollary 7]: If $g\colon \mathbb{R} \to \mathbb{R}$ is absolutely continuous and $v\colon \mathbb{R} \to \mathbb{R}$ is measurable and bounded, then for all α, β, $\int_{g(\alpha)}^{g(\beta)} v(t)dt = \int_\alpha^\beta v(g(s))\dot{g}(s)ds$. Suppose $D = f'$ almost everywhere and let $g : [0,1] \to \mathbb{R}$ be absolutely continuous, such that $g(0) = x$ and $g(1) = y$ where $(x, y) \in \mathbb{R} \times \mathbb{R}$. The restriction of f to any compact set is absolutely continuous, hence $f(y) - f(x) = \int_x^y f'(t)\,\mathrm{d}t$. Since D is locally bounded, the substitution formula applies for any measurable selection v of D to give

$$f(y) - f(x) = \int_x^y f'(t)\,\mathrm{d}t = \int_x^y v(t)\,\mathrm{d}t = \int_0^1 v(g(s))g'(s)\,\mathrm{d}s,$$

hence f is path differentiable. □

This gives a criterion to verify the path differentiability of a function from $\mathbb{R}$ to $\mathbb{R}$. A function $f : \mathbb{R} \to \mathbb{R}$ is path differentiable if and only if $\partial^c f$ is a singleton Lebesgue almost everywhere. This is also equivalent to the continuity of the derivative almost everywhere as for essentially smooth functions in [44].

On definable conservative gradients. More precise characterizations of conservative gradient and path differentiable functions are made in the definable setting. They all fundamentally derive from the Whitney stratification property of definable functions, leading to a projection formula [37, 40].

For instance, in [104], the authors show that for a definable function, a conservative gradient is included in the sum of the Clarke subgradient and a normal operator. Furthermore, the conservativity notion coincides with Norkin's semismooth gradient [62]. Some other consequences of the definable setting are Kurdyka-Lojasiewicz inequality and the following important result which is a Sard's theorem:

Theorem 3.10 (Morse-Sard theorem for conservative gradients [37, Theorem 5])**.** *Let D be a conservative gradient for $f : \mathbb{R}^p \to \mathbb{R}$ locally Lischitz. Assume D and f are definable in the same o-minimal structure. Then $f(\operatorname{crit} D)$ is finite, where* $\operatorname{crit} D := \{w \in \mathbb{R}^p \ : \ 0 \in D(w)\}$.

In particular, if f is definable locally Lipschitz, then $f(\operatorname{crit} \partial^c f)$ is finite.

3.2 A conservative integral rule

In this part $(S, \mathcal{A}, \mu)$ is a complete [1] measure space. We consider a function $f : \mathbb{R}^p \times S \to \mathbb{R}$ such that for almost all $s \in S$, $f(\cdot, s)$ is path-differentiable with conservative gradient $D(\cdot, s)$.

In general, one may not expect to have the Clarke subgradient of the integral equal to the integral of the Clarke subgradient. For instance, if $f\colon (w, s) \mapsto s|w|$ and $P = \frac{1}{2}\delta_{-1} + \frac{1}{2}\delta_1$ or P is the uniform density on $[-1, 1]$, then for $F(w) := \int_{[-1,1]} f(w, s)\,\mathrm{d}P(s)$, one has $\partial^c F(w) = 0$ for all $w \in \mathbb{R}$, but $\mathbb{E}_{\xi\sim P}[\partial^c_w f(0, \xi)] = [-1, 1]$. Our goal is thus to use conservative derivatives in order to establish an integral rule (Theorem 3.13), i.e., to show $\int_S D(\cdot, s)\,\mathrm{d}\mu(s)$ is a conservative gradient for $\int_S f(\cdot, s)\,\mathrm{d}\mu(s)$.

We are motivated by risk minimization problems in machine learning where the function to minimize is defined as an expectation with respect to an unknown distribution. Such a result is fundamental in order to justify first-order sampling schemes. Another motivation for such a result is to obtain a chain rule property for an abstract integral function. Until now, the most general way in the nonconvex setting to obtain it was to assume semialgebraicity or definability of the function [62]. These geometric assumptions can't hold reasonably on integral functions with respect to general measures.

First, we provide a result of derivation under the integral sign when the integrand is absolutely continuous in its first variable. We shall use the following lemma:

Lemma 3.11 (Measurability of partial derivatives)**.** *Let $U \subset \mathbb{R}$ be open and $f : U \times S \to \mathbb{R}$ a $(\mathcal{B}(\mathbb{R}) \times \mathcal{A})$-measurable function. We suppose that there exists $M \subset S$ of full measure such that for all $s \in M$, $f(\cdot, s)$ is absolutely continuous. Then $\frac{\partial f}{\partial x}$ is jointly measurable and is defined almost everywhere in $U \times S$. Also, for almost all $x \in U$, $\frac{\partial f}{\partial x}(x, s)$ is defined for almost all $s \in S$.*

Proposition 3.12 (Differentiation of absolutely continuous integrals)**.** *Let $U \subset \mathbb{R}$ be open and $f : U \times S \to \mathbb{R}$ such that:*

1. *For all $x \in U$, $f(x, \cdot)$ is integrable.*
2. *For almost all $s \in S$, $f(\cdot, s)$ is absolutely continuous.*
3. *$\frac{\partial f}{\partial x}$ is locally integrable, jointly in x and s: for any compact interval $[a, b] \subset U$,*

$$\int_S \int_a^b \left|\frac{\partial f}{\partial x}(x, s)\right| \mathrm{d}x\,\mathrm{d}\mu(s) < \infty.$$

Then, the function $g : x \mapsto \int_S f(x, s)\,\mathrm{d}\mu(s)$, is absolutely continuous, differentiable at almost all $x \in U$ with $g'(x) = \int_S \frac{\partial f}{\partial x}(x, s)\,\mathrm{d}\mu(s)$.

Proofs of Lemma 3.11 and Proposition 3.12 can be found in appendix Section 3.4.1.

The integral rule states as follows:

Theorem 3.13 (Integral rule of conservative gradients)**.** *Let $D : \mathbb{R}^p \times S \rightrightarrows \mathbb{R}^p$ and $f : \mathbb{R}^p \times S \to \mathbb{R}$ be such that:*

1. *For all $x \in \mathbb{R}^p$, $f(x, \cdot)$ is integrable.*

[1] This is a technical assumption meant to apply Fubini's theorem.

2. *For almost all $s \in S$, $f(\cdot, s)$ is locally Lipschitz continuous and $D(\cdot, s)$ is conservative for $f(\cdot, s)$.*

3. *$D : \mathbb{R}^p \times S \rightrightarrows \mathbb{R}^p$ is jointly measurable in $\mathcal{B}(\mathbb{R}^p) \times \mathcal{A}$.*

4. *For all compact subset $C \subset \mathbb{R}^p$, there exists an integrable function $\kappa : S \to \mathbb{R}_+$ such that for all $(x, s) \in C \times S$, $\|D(x, s)\| \leq \kappa(s)$, where for $(x, s) \in \mathbb{R}^p \times S$, $\|D(x, s)\| := \sup_{y \in D(x,s)} \|y\|$.*

Then $\int_S f(\cdot, s)\, \mathrm{d}\mu(s)$ is path-differentiable and $\int_S D(\cdot, s)\, \mathrm{d}\mu(s)$ is a conservative gradient for the integral $\int_S f(\cdot, s)\, \mathrm{d}\mu(s)$.

Proof: With the chain rule definition of conservativity, Definition 3.1, we will show that the problem reduces to the differentiation of an absolutely continuous integral, and then, we shall use Proposition 3.12 to conclude. Let $f : \mathbb{R}^p \times S \to \mathbb{R}$ and $D : \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ verifying the assumptions 1 to 4 displayed above.

Following the conservative gradient definition, Definition 3.1, we verify that $\int_S D(\cdot, s)\, \mathrm{d}\mu(s)$ is graph-closed, nonempty valued, and locally bounded. By the measurable selection theorem, see Proposition 2.2, $\int_S D(\cdot, s)\, \mathrm{d}\mu(s)$ is nonempty valued. It is locally bounded by item 4. For almost all $s \in S$, $D(\cdot, s)$ is graph-closed and locally bounded, hence it is upper semicontinuous, see [10, Corollary 1 in Chapter 1, Section 1]. By Aumann's integral properties, see [149, Theorem 2], and since $D(\cdot, s)$ is upper semicontinuous, compact valued for all s, then $\int_S D(\cdot, s)\, \mathrm{d}\mu(s)$ is graph-closed.

Now, we have to verify the chain rule property. Let $\gamma : [0, 1] \to \mathbb{R}^p$ be any absolutely continuous curve. By hypothesis, there exists a set of full measure $M \subset S$ such that for all $s \in M$, $f(\cdot, s)$ has conservative gradient $D(\cdot, s)$. We have $\forall s \in M$, $f(\gamma(\cdot), s)$ is absolutely continuous because f is locally Lipschitz in $x \in C$, and γ is absolutely continuous. Thus, $\forall s \in M$ $f(\gamma(\cdot), s)$ is differentiable a.e. and the chain rule property (3.1) holds for almost all $t \in [0, 1]$, i.e.,

$$\forall v \in D(\gamma(t), s),\ \frac{\mathrm{d}}{\mathrm{d}t} f(\gamma(t), s) = \langle v, \dot{\gamma}(t) \rangle. \tag{3.1}$$

Let $E \subset [0, 1] \times S$ be the domain of existence of $\frac{\mathrm{d}}{\mathrm{d}t} f(\gamma(t), s)$. E is measurable and of full measure according to Lemma 3.11. We want to verify the measurability of the domain of validity of eq. (3.1), which is $E \cap \{(t, s) \in [0, 1] \times S \ :\ \varphi(t, s) = 0\}$, where $\varphi(t, s) = \frac{\mathrm{d}}{\mathrm{d}t} f(\gamma(t), s) - \langle D(\gamma(t), s), \dot{\gamma}(t) \rangle$ for all $(t, s) \in E$ and $\varphi(t, s) = 1$ elsewhere. By Castaing's Theorem (see Remark 2.4) φ is measurable. The set $\{(t, s) \in [0, 1] \times S \ :\ \varphi(t, s) = 0\}$ is the upper inverse of $\{0\}$ by φ, hence it is jointly measurable. Similarly as in the proof of Lemma 3.11, by Fubini's Theorem, $\varphi^u(\{0\})$ is of full measure and there exists $I_1 \subset [0, 1]$ of full measure such that for all $t \in I$ eq. (3.1) holds for almost all $s \in S$.

Let $t \in I_1$. From eq. (3.1), we can say that, for any measurable selection $v : s \to \mathbb{R}^p$ of $D(\gamma(t), \cdot)$, we have for almost all $s \in S$

$$\frac{\mathrm{d}}{\mathrm{d}t} f(\gamma(t), s) = \langle v(s), \dot{\gamma}(t) \rangle. \tag{3.2}$$

Integrating (3.2) over $s \in S$ we have for any a in the Aumann integral $\int_S D(\gamma(t), s)\, \mathrm{d}s$ and measurable selection v such that $a = \int_S v(s)\, \mathrm{d}s$,

$$\int_S \frac{\mathrm{d}}{\mathrm{d}t} f(\gamma(t), s)\, \mathrm{d}\mu(s) = \int_S \langle v(s), \dot{\gamma}(t) \rangle\, \mathrm{d}\mu(s) = \langle a, \dot{\gamma}(t) \rangle. \tag{3.3}$$

On the other hand, $\gamma([0, 1])$ is compact by continuity of γ. Let κ be given by assumption 4 for the compact set $C = \gamma([0, 1])$. The Cauchy-Schwarz inequality gives for all $(t, s) \in [0, 1] \times S$,

$|\langle v(s), \dot\gamma(t)\rangle| \leq \|D(\gamma(t), s)\|\|\dot\gamma(t)\| \leq \kappa(s)\|\dot\gamma(t)\|$. Since γ is absolutely continuous, $\dot\gamma$ is integrable on $[0,1]$ hence the function $(t,s) \mapsto \kappa(s)\|\dot\gamma(t)\|$ is locally integrable jointly in (t,s), and so is $(t,s) \mapsto \frac{\mathrm{d}}{\mathrm{d}t} f(\gamma(t), s) = \langle v(s), \dot\gamma(t)\rangle$. Proposition 3.12 now applies to $(t,s) \mapsto f(\gamma(t), s)$, hence there exists I_2 of full measure such that

$$\forall t \in I_2,\ \int_S \frac{\mathrm{d}}{\mathrm{d}t} f(\gamma(t), s)\,\mathrm{d}\mu(s) = \frac{\mathrm{d}}{\mathrm{d}t}\int_S f(\gamma(t), s)\,\mathrm{d}\mu(s). \tag{3.4}$$

Combining eq. (3.3) which holds on I_1 and eq. (3.4) which holds on I_2 we have

$$\forall t \in I_1 \cap I_2,\ \forall a \in \int_S D(\gamma(t), s)\,\mathrm{d}\mu(s),\ \frac{\mathrm{d}}{\mathrm{d}t}\int_S f(\gamma(t), s)\,\mathrm{d}\mu(s) = \langle a, \dot\gamma(t)\rangle$$

and $I_1 \cap I_2$ is of full measure. Finally, we have shown that $\int_S D(\cdot, s)\,\mathrm{d}\mu(s)$ is nonempty, compact, valued graph-closed, and verifies the chain rule property, hence it is a conservative gradient for $\int_S f(\cdot, s)\,\mathrm{d}\mu(s)$. □

3.3 A conservative implicit differentiation formula

We are motivated by a recent trend in implicit models, or bi-level optimization problems, consisting in using implicit differentiation in order to differentiate fixed point solutions or the solutions of optimization problems. In these situations, the fixed point equations and optimality conditions are often nonsmooth. In practice, implicit differentiation is applied with the formal use of the automatic differentiation (backpropagation) oracle. In the nonsmooth setting, a Lipschitz version of the implicit function theorem exists [54] but doesn't come with a calculus: applying the implicit differentiation with Clarke Jacobians may not output a Clarke Jacobian of the implicit function, see Section 3.3.2.

Observing this gap, we extend the implicit differentiation formula to locally Lipschitz path differentiable equations. Our results apply to most practical (definable) situations. We present some applications to bi-level conic problems, hyperparameter optimization for lasso models, and deep equilibrium networks.

3.3.1 Conservative implicit differentiation

Our main nonsmooth implicit differentiation formula states as follows:

Theorem 3.14 (Conservative implicit differentiation)**.** *Let $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ be path differentiable on $\mathcal{U} \times \mathcal{V} \subset \mathbb{R}^n \times \mathbb{R}^m$ an open set and $G : \mathcal{U} \to \mathcal{V}$ a locally Lipschitz function such that, for each $x \in \mathcal{U}$,*

$$F(x, G(x)) = 0. \tag{3.5}$$

Furthermore, assume that for each $x \in \mathcal{U}$, for each $[A\ B] \in J_F(x, G(x))$, the matrix B is invertible where J_F is a conservative Jacobian for F. Then, $G : \mathcal{U} \to \mathcal{V}$ is path differentiable with conservative Jacobian given, for each $x \in \mathcal{U}$, by

$$J_G \colon x \rightrightarrows \left\{-B^{-1}A : [A\ B] \in J_F(x, G(x))\right\}.$$

Proof: Let $\gamma : [0,1] \to \mathcal{U}$ be absolutely continuous, then the composition $G \circ \gamma$ is also absolutely continuous since G is locally Lipschitz. By (3.5) we have, for all $t \in [0,1]$,

$$F(\gamma(t), G(t))) = 0$$

which we can differentiate almost everywhere; for almost every $t \in [0,1]$ and for any $[A\ B] \in J_F(\gamma(t), G(\gamma(t)))$,

$$[A\ B]\begin{bmatrix}\dot{\gamma}(t)\\ \frac{\mathrm{d}}{\mathrm{d}t}G(\gamma(t))\end{bmatrix} = 0 \implies -A\dot{\gamma}(t) = B\frac{\mathrm{d}}{\mathrm{d}t}G(\gamma(t)).$$

Since B is assumed to be invertible, we have, for almost every $t \in [0,1]$,

$$-B^{-1}A\dot{\gamma}(t) = \frac{\mathrm{d}}{\mathrm{d}t}G(\gamma(t)).$$

The set-valued mapping $J_G\colon x \rightrightarrows \left\{-B^{-1}A : [A\ B] \in J_F(x, G(x))\right\}$ is nonempty, locally bounded, and has a closed graph for each $x \in \mathcal{U}$ since $J_F(x, G(x))$ is a conservative Jacobian and B is invertible. We conclude that G is path differentiable on $\mathcal{U}$ with conservative Jacobian J_G. □

Under an invertibility condition, one has the existence of an implicit function and the formula applies:

Corollary 3.15 (Path differentiable implicit function theorem)**.** *Let $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ be path differentiable with conservative Jacobian J_F. Let $(\hat{x}, \hat{y}) \in \mathbb{R}^n \times \mathbb{R}^m$ be such that $F(\hat{x}, \hat{y}) = 0$. Assume that $J_F(\hat{x}, \hat{y})$ is convex and that, for each $[A\ B] \in J_F(\hat{x}, \hat{y})$, the matrix B is invertible. Then, there exists an open neighborhood $\mathcal{U} \times \mathcal{V} \subset \mathbb{R}^n \times \mathbb{R}^m$ of $(\hat{x}, \hat{y})$ and a path differentiable function $G : \mathcal{U} \to \mathcal{V}$ such that the conclusion of Theorem 3.14 holds.*

Proof: Since $J_F(\hat{x}, \hat{y})$ is convex, it follows from Theorem 3.8 that $\mathrm{Jac}^c F(\hat{x}, \hat{y}) \subset J_F(\hat{x}, \hat{y})$ and thus, for any $[A\ B] \in \mathrm{Jac}^c F(\hat{x}, \hat{y})$, B is invertible, i.e., the conditions to apply [54, 7.1 Corollary] to F are satisfied. Therefore there exists an open neighborhood $\mathcal{U}_1 \times \mathcal{V}_1 \subset \mathbb{R}^n \times \mathbb{R}^m$ of $(\hat{x}, \hat{y})$ and a locally Lipschitz function $G : \mathcal{U}_1 \to \mathcal{V}_1$ such that, for all $x \in \mathcal{U}_1$,

$$F(x, G(x)) = 0.$$

By the continuity of the determinant and the fact that J_F has a closed graph, there exists an open neighborhood $\mathcal{U}_2 \times \mathcal{V}_2 \subset \mathbb{R}^n \times \mathbb{R}^m$ of $(\hat{x}, \hat{y})$ such that, for all $(x, y) \in \mathcal{U}_2 \times \mathcal{V}_2$, for all $[A\ B] \in J_F(x, y)$, the matrix B is invertible. Let $\mathcal{U} \times \mathcal{V} := (\mathcal{U}_1 \cap \mathcal{U}_2) \times (\mathcal{V}_1 \cap \mathcal{V}_2)$, which is an open neighborhood of $(\hat{x}, \hat{y})$. Then the requirements of Theorem 3.14 are met for F, J_F, and G on $\mathcal{U} \times \mathcal{V}$ and the desired claims follow. □

A natural consequence is a nonsmooth inverse differentiation formula:

Corollary 3.16 (Path differentiable inverse function theorem)**.** *Let $\mathcal{U}$ and $\mathcal{V}$ be open neighborhoods of 0 in $\mathbb{R}^n$ and $\Phi : \mathcal{U} \to \mathcal{V}$ path differentiable with $\Phi(0) = 0$. Assume that Φ has a conservative Jacobian J_Φ such that $J_\Phi(0)$ contains only invertible matrices. Then, locally, Φ has a path differentiable inverse Ψ with a conservative Jacobian given by*

$$J_\Psi(y) = \left\{A^{-1} : A \in J_\Phi(\Psi(y))\right\}.$$

Proof: Consider the function $F(x, y) = x - \Phi(y)$ and observe that it satisfies the assumptions of Corollary 3.15, so that we obtain a function G which is exactly the desired inverse. □

3.3.2 Counterexample to a potential Clarke implicit differentiation formula

One may not expect $J_G \colon x \rightrightarrows \left\{-B^{-1}A : [A\ B] \in \operatorname{Jac}^c F(x, G(x))\right\}$ to be equal to $\operatorname{Jac}^c G$ in general.

In order to provide a counterexample, we follow the example given by Clarke [54, Remark 7.1.2]. Consider the mapping $\Phi : \mathbb{R}^2 \to \mathbb{R}^2$ given by

$$\Phi(x, y) = (|x| + y, 2x + |y|)\,.$$

It is locally Lipschitz and semialgebraic and thus path differentiable with its Clarke Jacobian a conservative Jacobian. We have the following explicit piecewise linear representation

$$\Phi(x, y) = \begin{cases} (x + y, 2x + y) & \text{if } \ x \geq 0 \text{ and } y \geq 0, \\ (x + y, 2x - y) & \text{if } \ x \geq 0 \text{ and } y \leq 0, \\ (-x + y, 2x - y) & \text{if } \ x \leq 0 \text{ and } y \leq 0, \\ (-x + y, 2x + y) & \text{if } \ x \leq 0 \text{ and } y \geq 0 \end{cases}$$

from which we deduce that the Clarke Jacobian of Φ has the following structure

$$\operatorname{Jac}^c \Phi(0) = \operatorname{conv} \left\{ \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix} \right\}$$

where the matrices correspond to linear maps in the explicit definition of Φ. Therefore $\operatorname{Jac}^c \Phi(0)$ is an affine set whose dimension is 2. In addition, it contains only invertible matrices [54, Remark 7.1.2]. We will use the following explicit matrix inverses:

$$\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}^{-1} = \frac{1}{3} \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}, \quad \begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix}^{-1} = \frac{1}{3} \begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix}.$$

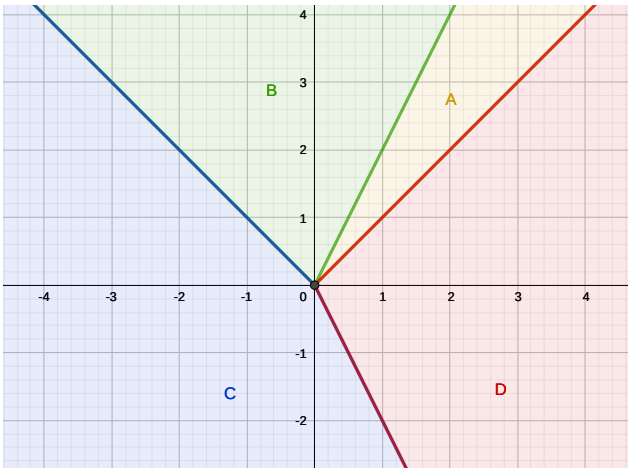


Figure 3.1: Illustration of the four different sets in the explicit piecewise affine representation of $\Psi = \Phi^{-1}$.

Using the above, one can verify that Φ is a homeomorphism whose inverse is also piecewise

linear. We set $\Psi = \Phi^{-1}$; it is given by

$$\begin{aligned}
\Psi(u,v) &= (v-u, 2u-v) && \text{for } (u,v) \in A,\\
\Psi(u,v) &= \frac{1}{3}(u+v, 2u-v) && \text{for } (u,v) \in B,\\
\Psi(u,v) &= (u+v, 2u+v) && \text{for } (u,v) \in C,\\
\Psi(u,v) &= \frac{1}{3}(v-u, 2u+v) && \text{for } (u,v) \in D,
\end{aligned}$$

where the subsets A, B, C, D form a "partition"[2] of $\mathbb{R}^2$

$$\begin{aligned}
A &= \left\{(u,v) \in \mathbb{R}^2 : v-u \geq 0, 2u-v \geq 0\right\} && \text{(corresponding to } x \geq 0, y \geq 0),\\
B &= \left\{(u,v) \in \mathbb{R}^2 : u+v \geq 0, 2u-v \leq 0\right\} && \text{(corresponding to } x \geq 0, y \leq 0),\\
C &= \left\{(u,v) \in \mathbb{R}^2 : u+v \leq 0, 2u+v \leq 0\right\} && \text{(corresponding to } x \leq 0, y \leq 0),\\
D &= \left\{(u,v) \in \mathbb{R}^2 : v-u \leq 0, 2u+v \geq 0\right\} && \text{(corresponding to } x \leq 0, y \geq 0).
\end{aligned}$$

A graphical representation of these sets is given in Figure 3.1.

From this explicit piecewise linear representation of Ψ, we deduce that its Clarke Jacobian at 0 is the following

$$\operatorname{Jac}^c \Psi(0) = \operatorname{conv}\left\{ \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix}, \frac{1}{3}\begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix}, \frac{1}{3}\begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \right\}.$$

For a given subset of linear space we denote by $\operatorname{aff} F$ the affine span of F. It is easy to see that $\dim \operatorname{aff}[\operatorname{Jac}^c \Phi(0)] = 2$ while $\dim \operatorname{aff}[\operatorname{Jac}^c \Psi(0)] = 3$. More concretely, vectorize the set $\operatorname{Jac}^c \Psi(0)$ at $M = \frac{1}{3}\begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}$ by considering the matrices given by

$$\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} - M, \quad \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix} - M, \quad \frac{1}{3}\begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix} - M$$

that is

$$\frac{1}{3}\begin{bmatrix} 2 & 2 \\ 4 & 4 \end{bmatrix}, \quad \frac{1}{3}\begin{bmatrix} -4 & 2 \\ 4 & -2 \end{bmatrix}, \quad \frac{1}{3}\begin{bmatrix} -2 & 0 \\ 0 & 2 \end{bmatrix}.$$

These matrices are independent so that $\operatorname{Jac}^c \Psi(0)$ is an affine set whose dimension is 3.

Matrix inversion is a semialgebraic diffeomorphism (when restricted to invertible matrices) so it preserves dimension. For this reason the set $[\operatorname{Jac}^c \Psi(0)]^{-1} = \{M^{-1}, M \in \operatorname{Jac}^c \Psi(0)\}$ is a semialgebraic set of dimension 3, and we have

$$[\operatorname{Jac}^c \Psi(0)]^{-1} \not\subset [\operatorname{Jac}^c \Phi(0)]. \tag{3.6}$$

However, we have shown that $z \mapsto [\operatorname{Jac}^c \Psi(\Phi(z))]^{-1}$ is a conservative Jacobian. This example excludes the possibility of a simple inverse (implicit) function theorem with a "Clarke Jacobian calculus" and illustrates the requirement for a more flexible notion (conservativity) when using calculus rules in an implicit function (or inverse function) context.

[2]Each piece having two half-lines in common with other pieces.

3.3.3 Applications of nonsmooth implicit differentiation in Machine Learning

We present some applications of our nonsmooth implicit differentiation formula in machine learning. The proofs are postponed in appendix Section 3.4.2.

Monotone deep equilibrium networks. Deep Equilibrium Networks (DEQs) [12] are specific neural network architectures including layers whose input-output relation is implicitly defined through a fixed point equation of the form

$$z = f(z, x) \tag{3.7}$$

where $x \in \mathbb{R}^p$ is a given input and $z \in \mathbb{R}^m$ is the corresponding output. We may consider that the variable x represents both the input layer and layer parameters. Assuming that, for each $x \in \mathbb{R}^p$, there is a unique $z \in \mathbb{R}^m$ satisfying the relation (3.7), this defines an input-output relation $z\colon \mathbb{R}^p \to \mathbb{R}^m$. Furthermore, if f is path differentiable with convex-valued conservative Jacobian $J_f\colon \mathbb{R}^m \times \mathbb{R}^p \rightrightarrows \mathbb{R}^{m\times(m+p)}$ whose projection on the first m columns are all invertible, then the function z itself admits a conservative Jacobian which can be computed from Theorem 3.14.

We now focus on monotone operator implicit layers [162] for which assumptions are easily stated. Our method applies to other similar architectures, e.g., DEQs [12] or implicit graph neural networks [81]. Let $\sigma\colon \mathbb{R}^m \to \mathbb{R}^m$ be the proximal operator of a convex function and assume σ is path differentiable with conservative Jacobian $J_\sigma\colon \mathbb{R}^m \rightrightarrows \mathbb{R}^{m\times m}$, assumed to be convex-valued. This encompasses the majority of activation functions used in practice [57]. Let $W \in \mathbb{R}^{m\times m}$ be a matrix such that $W + W^\mathsf{T} \succeq 2\theta I$ with $\theta > 0$. Under these assumptions the implicit equation

$$z = \sigma(Wz + b) \tag{3.8}$$

has a unique output $z(W, b)$ [162, Theorem 2]. The transformation $(W, b) \mapsto z(W, b)$ is a *monotone implicit layer.*

The set-valued mapping obtained from Theorem 3.14 provides a conservative Jacobian for $(W, z) \mapsto z(W, z)$. A similar expression was described in [162, Theorem 2], without using conservativity and using the Clarke Jacobian formally as a classical Jacobian. The proposition below provides a full justification of this heuristic and ensures the convergence of algorithmic differentiation-based training.

Proposition 3.17 (Path differentiation through monotone layers)**.** *Assume that J_σ is convex-valued and that, for all $J \in J_\sigma(Wz(W, b)+b)$, the matrix $(\mathrm{Id}_m - JW)$ is invertible. Consider a loss-like function $\ell\colon \mathbb{R}^m \to \mathbb{R}$ with conservative gradient $D_\ell\colon \mathbb{R}^m \rightrightarrows \mathbb{R}^m$, then $g : (W, z) \mapsto \ell(z(W, b))$ is path differentiable and has a conservative gradient D_g defined through*

$$D_g\colon (W, b) \rightrightarrows \left\{ J^\mathsf{T}(\mathrm{Id}_m - JW)^{-T} v z^\mathsf{T}, J^\mathsf{T}(\mathrm{Id}_m - JW)^{-T} v) :\ J \in J_\sigma(Wz + b),\, v \in D_\ell(z) \right\}.$$

Remark 3.18. Convexity and invertibility assumptions are satisfied when J_σ is the Clarke Jacobian [162].

Optimization layers: the conic program case. Optimization layers in deep learning may take many forms; we consider here those based on conic programming [2, 3, 6, 50]. We follow [3], simplifying the analysis by ignoring infeasibility certificates, which correspond to the absence of a primal-dual solution [50], in line with the implementation described in [2, Appendix B]. Consider a conic problem (P) and its dual (D):

$$\begin{array}{llll} \text{(P)} & \inf & c^\mathsf{T} x & \\ & \text{subject to} & Ax + s = b & \\ & & s \in \mathcal{K} & \end{array} \qquad \begin{array}{llll} \text{(D)} & \inf & b^\mathsf{T} y & \\ & \text{subject to} & A^\mathsf{T} y + c = 0 & \\ & & y \in \mathcal{K}^*, & \end{array} \tag{3.9}$$

with primal variable $x \in \mathbb{R}^n$, dual variable $y \in \mathbb{R}^m$, and primal slack variable $s \in \mathbb{R}^m$. The set $\mathcal{K} \subset \mathbb{R}^m$ is a nonempty closed convex cone and $\mathcal{K}^* \subset \mathbb{R}^m$ is its dual cone. The problem parameters are the matrix $A \in \mathbb{R}^{m \times n}$ and the vectors $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$; the cone $\mathcal{K}$ is fixed. Under the assumption that there is a unique primal-dual solution (x, y, s), we study the path differentiability of the solution mapping as a function of its parameters:

$$(A, b, c) \mapsto \mathrm{sol}(A, b, c) = (x, y, s).$$

For this, let us interpret the solution mapping as a composition mapping involving equation-like implicit formulations. Set $N = n + m$, given $A, b, c \in \mathbb{R}^{m \times n} \times \mathbb{R}^m \times \mathbb{R}^n$, define

$$Q(A, b, c) = \begin{bmatrix} 0 & A^\mathsf{T} \\ -A & 0 \end{bmatrix} \in \mathbb{R}^{N \times N} \qquad V(b, c) = \begin{bmatrix} c \\ b \end{bmatrix} \in \mathbb{R}^N.$$

Consider a vector $z = (u, v) \in \mathbb{R}^n \times \mathbb{R}^m$, denote by proj the projection onto $\mathbb{R}^n \times \mathcal{K}^*$ and define the *residual map* $\mathcal{N} : \mathbb{R}^N \times \mathbb{R}^{m \times n} \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^N$ as

$$\mathcal{N}(z, A, b, c) = (Q(A, b, c) - \mathrm{Id}_N) \operatorname{proj} z + V(b, c) + z.$$

The mapping $\mathcal{N}$ is a synthetic form of optimality measure for (P) and (D), capturing KKT conditions. To simplify the presentation, we ignore the extreme cases of infeasibility and unboundedness which correspond to an absence of solution in [50].

Define the function $\phi : \mathbb{R}^N \to \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$ through $\phi(u, v) := (u, P_{\mathcal{K}^*}(v), P_{\mathcal{K}^*}(v) - v)$. We can show $\phi(u, v)$ provides a primal-dual KKT solution of problems (P) and (D) if and only if $\mathcal{N}(z, A, b, c) = 0$, see appendix Section 3.4.2. When we assume that, for fixed A, b, and c, there is a unique $z \in \mathbb{R}^N$ such that $\mathcal{N}(z, A, b, c) = 0$, we have an implicitly defined a function $z = \nu(A, b, c)$, such that

$$\mathrm{sol}(A, b, c) = [\phi \circ \nu](A, b, c). \tag{3.10}$$

We can now apply our conservative implicit differentiation formula. The following result extends the discussion in [3, 50], limited to situations where proj is differentiable at the proposed solution z, to a fully nonsmooth setting.

Proposition 3.19 (Path differentiation through cone programming layers)**.** *Assume that $P_{\mathcal{K}^*}$, $\mathcal{N}$ are path differentiable, denote respectively by $J_{P_{\mathcal{K}^*}}$, $J_{\mathcal{N}}$ corresponding convex-valued conservative Jacobians. Assume that, for all $A, b, c \in \mathbb{R}^{m \times n} \times \mathbb{R}^m \times \mathbb{R}^n$, $z = \nu(A, b, c) \in \mathbb{R}^n \times \mathbb{R}^m$ is the unique solution to $\mathcal{N}(z, A, b, c) = 0$ and that all matrices formed from the N first columns of $J_{\mathcal{N}}(z, A, b, c)$ are invertible. Then, ϕ, ν, and* sol *are path differentiable functions with conservative Jacobians:*

$$\begin{aligned} J_\nu(A, b, c) &:= \left\{ -U^{-1} V : [U \ V] \in J_{\mathcal{N}}(\nu(A, b, c), A, b, c) \right\}, \\ J_\phi(z) &:= \begin{bmatrix} \mathrm{Id}_n & 0 \\ 0 & J_{P_{\mathcal{K}^*}}(v) \\ 0 & (J_{P_{\mathcal{K}^*}}(v) - \mathrm{Id}_m) \end{bmatrix}, \\ J_{\mathrm{sol}}(A, b, c) &:= J_\phi(\nu(A, b, c)) J_\nu(A, b, c). \end{aligned}$$

In practice, the path differentiability of conic projections is pervasive since they are generally semialgebraic (orthant, second-order cone, PSD cone). See [85, 94, 109, 109] for the computations of the corresponding Clarke Jacobians (which are conservative). Note that a conservative Jacobian for $\mathcal{N}$ may be obtained from $J_{P_{\mathcal{K}^*}}$ using Proposition 3.4.

Hyperparameter selection for Lasso type problems. Implicit differentiation can be used to tune hyperparameters via first-order methods optimizing some measure of task performance, see [26] and references therein. In a nonsmooth context, we recall the formulation in [25] of the general hyperparameter optimization problem as a bi-level optimization problem:

$$\min_{\lambda\in\mathbb{R}^m} C(\hat{\beta}(\lambda)) \quad \text{such that} \quad \hat{\beta}(\lambda) \in \operatorname*{argmin}_{\beta\in\mathbb{R}^p} \psi(\beta,\lambda)$$

where $C:\mathbb{R}^p\to\mathbb{R}$ is continuously differentiable (e.g., test loss) and $\psi\colon \mathbb{R}^p\times\mathbb{R}^m\to\mathbb{R}$ is a possibly nonsmooth training loss, convex in β, with hyperparameter $\lambda\in\mathbb{R}^m$. We seek a subgradient-type method for this problem with convergence guarantees; our nonsmooth implicit differentiation results can be used for this purpose. We demonstrate this approach on the Lasso problem [152]

$$\hat{\beta}\left(\lambda\right) \in \operatorname{argmin}\left\{\frac{1}{2}\left\|y-X\beta\right\|_2^2 + e^{\lambda}\left\|\beta\right\|_1 : \beta\in\mathbb{R}^p\right\} \tag{3.11}$$

where $y\in\mathbb{R}^n$ is the vector of observations, $X=[X_1,\ldots,X_p]\in\mathbb{R}^{n\times p}$ is the design matrix with columns $X_j\in\mathbb{R}^n$, $j\in\{1,\ldots p\}$, and $\lambda\in\mathbb{R}$ is the hyperparameter. We may recall the definition of the proximal operator: given a convex proper lower semicontinuous function $f:\mathbb{R}^n\to\mathbb{R}\cup\{\infty\}$, we define its proximal operator through $x\in\mathbb{R}^n$, $\operatorname{prox}_f\left(x\right) := \operatorname*{argmin}_{u\in\mathbb{R}^n}\{f\left(u\right)+\frac{1}{2}\left\|u-x\right\|^2\}$.

Define $F:\mathbb{R}\times\mathbb{R}^p\to\mathbb{R}^p$ to be

$$F\left(\lambda,\beta\right) := \beta - \operatorname{prox}_{e^{\lambda}\|\cdot\|_1}\left(\beta - X^{\mathsf{T}}\left(X\beta-y\right)\right)$$

and recall that, for each $i\in\{1,\ldots,p\}$, $[\operatorname{prox}_{e^{\lambda}\|\cdot\|_1}(\beta)]_i = \operatorname{sign}(\beta_i)\max\{|\beta_i|-e^{\lambda},0\}$. The function $F(\lambda,\beta)$ is thus nonsmooth but locally Lipschitz on $\mathbb{R}\times\mathbb{R}^p$. An optimal $\hat{\beta}(\lambda)$ for (3.11) must satisfy $F(\lambda,\hat{\beta}(\lambda))=0$ [58, Prop. 3.1]. For a given solution $\hat{\beta}(\lambda)$, we introduce the equicorrelation set by $\mathcal{E} := \{j\in\{1,\ldots,p\} : |X_j^{\mathsf{T}}(y-X\hat{\beta}(\lambda))| = e^{\lambda}\}$ which contains the support set $\operatorname{supp}\hat{\beta} := \{i\in\{1,\ldots,p\} : \hat{\beta}_i\neq 0\}$. In fact, $\mathcal{E}$ does not depend on the choice of the solution $\hat{\beta}$, see [153, Lemma 1].

Proposition 3.20 (Conservative Jacobian for the solution mapping)**.** *For all $\lambda\in\mathbb{R}$, assume $X_{\mathcal{E}}^{\mathsf{T}}X_{\mathcal{E}}$ is invertible where $X_{\mathcal{E}}$ is the submatrix of X formed by taking the columns indexed by $\mathcal{E}$. Then $\hat{\beta}(\lambda)$ is single-valued, path differentiable with conservative Jacobian, $J_{\hat{\beta}}\left(\lambda\right)$, given for all λ as*

$$\left\{\left[-e^{\lambda}\left(\mathrm{Id}_p - \operatorname{diag}\left(q\right)\left(\mathrm{Id}_p - X^{\mathsf{T}}X\right)\right)^{-1}\operatorname{diag}\left(q\right)\operatorname{sign}\left(\hat{\beta}-X^{\mathsf{T}}\left(X\hat{\beta}-y\right)\right)\right] \;:\; q\in\mathcal{M}(\lambda)\right\}$$

where $\mathcal{M}(\lambda)\subset\mathbb{R}^p$ is the set of vectors q such that $q_i=1$ if $i\in\operatorname{supp}\hat{\beta}$, $q_i=0$ if $i\notin\mathcal{E}$ and $q_i\in[0,1]$ if $i\in\mathcal{E}\setminus\operatorname{supp}\hat{\beta}$.

Taking, in Proposition 3.20, $q_i=1$ for all $i\in\mathcal{E}$ corresponds to the directional derivative given by LARS algorithm [71], see also [107]. Alternatively, taking $q_i=0$ for $i\notin\operatorname{supp}\hat{\beta}$ gives the weak derivative described by [25]. Both are particular selections in $J_{\hat{\beta}}$, which is the underlying conservative field.

3.3.4 Some pathological examples beyond the invertibility condition

The invertibility condition of Theorem 3.14 is an important point to satisfy in practical applications. In this part, we highlight through several toy examples that using implicit differentiation outside the invertibility condition can result in absurd training dynamics. Details on the experiments are found in appendix Section 3.4.3.

A cyclic gradient dynamics via fixed-point/optimization layer. Consider the bi-level problem:

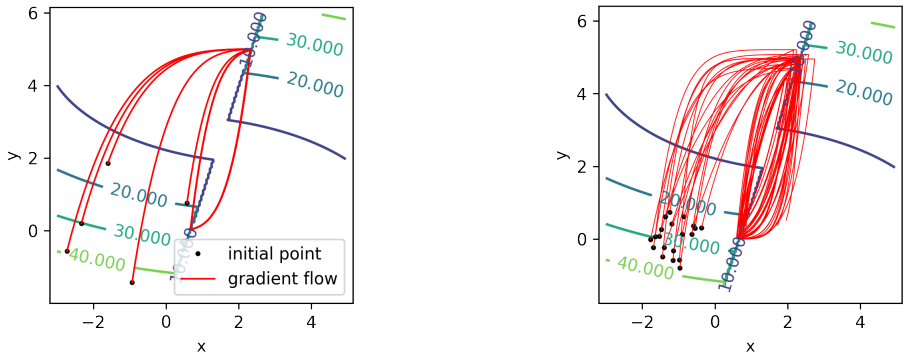
$$\begin{aligned} \min_{x,y,s} \quad & \ell(x,y,s) := (x-s_1)^2 + 4(y-s_2)^2 \\ \text{s.t.} \quad & s \in s(x,y) := \arg\max \{(a+b)(-3x+y+2) : \ a \in [0,3], b \in [0,5]\}. \end{aligned} \tag{3.12}$$

Problem (3.12) has an equivalent fixed-point formulation using projected gradient descent on the inner problem, see Section 3.4.3. Backpropagation applied to (3.12) associates to (x, y) the following:

$$\nabla_{(x,y)} \ell(x,y,s(x)) + \tilde{J}_s(x,y)^\mathsf{T} \nabla_s \ell(x,y,s(x)) \tag{3.13}$$

where $\tilde{J}_s$ is piecewise derivative. We implement gradient descent for (3.12), evaluating (3.13) either using `cvxpylayers` [2] or the `JAX` tutorial [165] for fixed-point layers. In both cases, the invertibility condition in Theorem 3.14 fails when $-3x + y + 2 = 0$, resulting in discontinuity of s, affecting the dynamics globally: the gradient trajectory converges to a limit cycle of non-critical points (Figure 3.2a); see Section 3.4.3 for details.

Persistence under small perturbations: For different initial points the gradient flow converges to the same limit cycle (Figure 3.2a). The cycle persists even if we perturb the coefficients in the problem (3.12) (Figure 3.2b, see appendix Section 3.4.3 for details on the perturbed problems).



(a) (b)

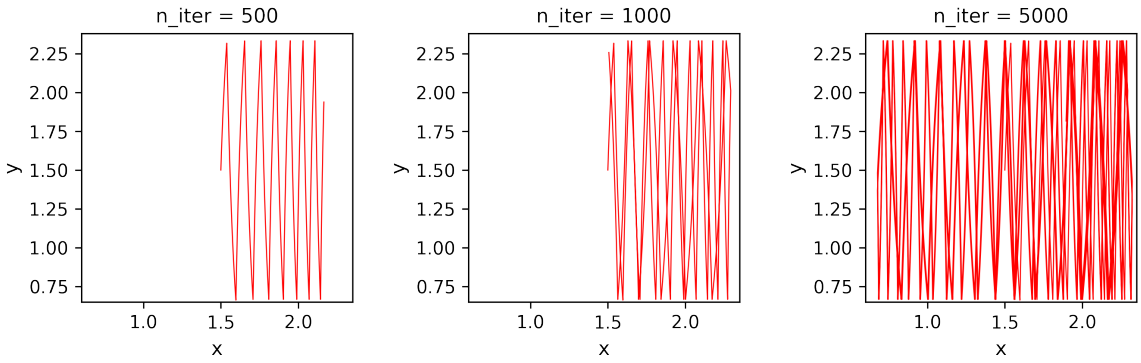
Figure 3.2: (a) Gradient flow for several initializations. (b) Gradient flow for 20 perturbed experiments with $\sigma^2 = 0.4$.

A chaotic dynamics in $\mathbb{R}^4$

We combine two cycles of the previous example into gradient dynamics in $\mathbb{R}^4$. To perform this, we consider a block-separable sum of the same function where we add a scaling parameter $\eta > 0$:

$$g : (x, y, z, w) \mapsto f(x, y) + \eta f(z, w).$$

This will combine the two cycles but the parameter η will make one cycle "faster" than the other. Projecting the path of the gradient descent on the variables (y, z) we obtain chaotic dynamics filling the space as the number of iterations increases.



(a) (b) (c)

Figure 3.3: Gradient path after (a) 500, (b) 1000, and (c) 5000 iterations.

Lorenz-like dynamics. The Lorenz Ordinary Differential Equation (ODE) writes:

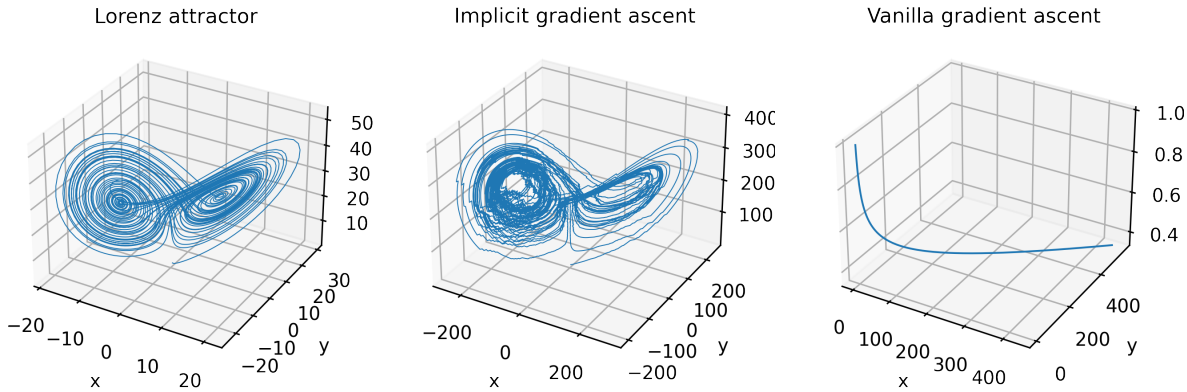
$$\dot{x} = \sigma(y - x), \qquad \dot{y} \quad = x(\rho - z) - y, \quad \text{and} \quad \dot{z} = xy - \beta z. \tag{3.14}$$

It is well-known that taking $(\sigma, \rho, \beta) = (10, 28, 8/3)$, and $(x(0), y(0), z(0)) = (0, 1, 1.05)$ gives a chaotic trajectory, displayed in Figure 3.4a. Denoting $F : (x, y, z) \mapsto (\sigma(y-x), x(\rho-z)-y, xy-\beta z)$ the vector field of the Lorenz system (3.14), consider the optimization problem

$$\max_{u \in \mathbb{R}^3} \quad u^\mathsf{T} z \qquad \text{s.t.} \qquad z \in \arg\min_{s \in \mathbb{R}^3} \|s - F(u)\|^4, \tag{3.15}$$

where the power 4 aims to break the invertibility condition of Proposition 3.19. This problem is obviously equivalent to

$$\max_{u \in \mathbb{R}^3} \quad u^\mathsf{T} F(u). \tag{3.16}$$



(a) (b) (c)

Figure 3.4: Implicit gradient ascent (b) outputs a pathological curve with some qualitative aspects of the Lorenz dynamics (a) and really different from a classical gradient (c).

The function $g : u \mapsto u^\mathsf{T} F(u)$ is a nondegenerate quadratic function, see Section 3.4.3. The function g has a unique critical point $(0, 0, 0)$ which is a strict saddle point. We perform gradient ascent with implicit differentiation using `cvxpylayers` on (3.15), and the classical gradient ascent on the equivalent problem (3.16). The path obtained by implicit differentiation (Figure 3.4b) resembles the Lorenz attractor (Figure 3.4a), in stark contrast to the conventional method (Figure 3.4c). The chaotic dynamics are a consequence of the lack of invertibility, due to the power 4 in (3.15), and various numerical approximations related to optimization and implicit differentiation.

3.4 A comparison of conservativity and semismoothness

We compare the conservative derivatives notion to its semismooth counterpart due to Norkin [120]. The main result we establish is that in general, semismooth gradients are conservative gradients. Formerly, the equivalence was shown in [62] in the semialgebraic case and a chain rule along semismooth curves was shown for semismooth gradients [142].

Throughout this section, $f\colon \mathbb{R}^p \to \mathbb{R}$ is Lipschitz continuous and $D\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ is locally bounded nonempty convex-valued and upper semicontinuous. Convex values are indeed required by Norkin in [119–121].

Definition 3.21 (Semismooth generalized gradients)**.** The set-valued mapping D is a *semismooth generalized gradient of* f if for all $x \in \mathbb{R}^p$, we have

$$\limsup_{y \to x,\, g \in D(y)} \frac{f(y) - f(x) - \langle g, y - x\rangle}{\|y - x\|} = 0.$$

The lim sup property in the definition is referred to as the *semismoothness property* of the generalized gradients. On the other hand, conservative gradients are defined in Definition 3.1. In both cases, the corresponding set-valued gradient map is a singleton almost everywhere and contains the Clarke subgradient of f everywhere [37, 119]. Functions with generalized gradient are called *differentiable in the generalized sense*, and those with conservative gradients are called path-differentiable.

The following strengthens the chain rule along semismooth curves given in [142, Theorem 1] and generalizes [62] beyond the definable setting.

Proposition 3.22. *If D is a Norkin's semismooth gradient of f, then it is a conservative gradient of f.*

Proof: We shall use the chain rule characterization of conservative gradients, Definition 3.1. Let D be a generalized gradient of f as in Definition 3.21, and $\gamma\colon [0, 1] \to \mathbb{R}^p$ be an absolutely continuous path. Then both γ and $f \circ \gamma$ are absolutely continuous, hence differentiable almost everywhere. Therefore, there exists a full measure subset $R \subset [0, 1]$ such that both are differentiable at every point on R.

Suppose, toward a contradiction, that the chain rule is not valid along γ, that is, there exists a non-zero measure set $E_1 \subset R$ such that for all $t \in E_1$, there is $g \in D(\gamma(t))$ such that $\frac{\mathrm{d}}{\mathrm{d}t}(f \circ \gamma)(t) \neq \langle g, \dot\gamma(t)\rangle$. Note that this implies that $\dot\gamma(t) \neq 0$ for all $t \in E_1$, since if $\dot\gamma(t) = 0$ then $0 = \frac{\mathrm{d}}{\mathrm{d}t}(f \circ \gamma)(t) = \langle g, \dot\gamma(t)\rangle$. Reducing E_1 and changing sign if necessary, we may assume without loss of generality that for all $t \in E_1$, there is $g \in D(\gamma(t))$ such that $\frac{\mathrm{d}}{\mathrm{d}t}(f \circ \gamma)(t) < \langle g, \dot\gamma(t)\rangle$.

Consider the measurable function (measurability is justified in [37]), $g\colon [0, 1] \to \mathbb{R}^p$, defined for all $t \in R$ by $g(t) = \arg\max_{v \in D(\gamma(t))} \langle \dot\gamma(t), v\rangle$ and $g(t) = 0$ otherwise.

We have for all $t \in E_1$, $0 < \langle \dot\gamma(t), g(t)\rangle - \frac{\mathrm{d}}{\mathrm{d}t}(f\circ\gamma)(t)$. This means that there is $\epsilon > 0$ and a nonzero set $E_2 \subset E_1$ such that $\epsilon \leq \langle \dot\gamma(t), g(t)\rangle - \frac{\mathrm{d}}{\mathrm{d}t} f\circ\gamma(t)$ for all $t \in E_2$ (otherwise, one would have $\langle \dot\gamma, g\rangle - \frac{\mathrm{d}}{\mathrm{d}t}(f\circ\gamma) = 0$ almost everywhere on E_1).

Let us apply Lusin's theorem (see, e.g., [139, Section 3.3]) and fix an arbitrary $\alpha > 0$, such that $\lambda(E_2) > \alpha$. There is a closed subset $E_3 \subset E_2$ such that $\lambda(E_2 \setminus E_3) < \alpha$ and g restricted to E_3 is continuous. The set E_3 has positive measure since $\lambda(E_3) = \lambda(E_2) - \lambda(E_2 \setminus E_3) > \alpha - \alpha = 0$. Let us summarize, $E_3 \subset [0,1]$ is closed with positive measure and we have the following on E_3:

- Both $f\circ\gamma$ and γ have derivatives and $\dot\gamma \neq 0$.
- $\frac{\mathrm{d}}{\mathrm{d}t}(f\circ\gamma) + \epsilon \leq \langle \dot\gamma, g\rangle$.
- g restricted to E_3 is continuous.

Lebesgue density theorem (see, e.g., [73, Theorem 1.35]) ensures that almost all $t \in E_3$ have density 1, that is, $\lambda([t-\delta, t+\delta]\cap E_3)/\lambda([t-\delta,t+\delta]) \to 1$ as $\delta \to 0$. Since E_3 has a positive measure, there exists $\bar t \in E_3$, a point of density 1 in E_3. We have for all $t \neq \bar t$, such that $\gamma(t) \neq \gamma(\bar t)$,

$$\begin{aligned}
\frac{f(\gamma(t)) - f(\gamma(\bar t))}{(t-\bar t)} &= \frac{\|\gamma(t)-\gamma(\bar t)\|}{t-\bar t}\frac{f(\gamma(t)) - f(\gamma(\bar t))}{\|\gamma(t)-\gamma(\bar t)\|} \\
&= \frac{\|\gamma(t)-\gamma(\bar t)\|}{t-\bar t}\left(\frac{f(\gamma(t)) - f(\gamma(\bar t)) - \langle g(t), \gamma(t)-\gamma(\bar t)\rangle}{\|\gamma(t)-\gamma(\bar t)\|} + \frac{\langle g(t), \gamma(t)-\gamma(\bar t)\rangle}{\|\gamma(t)-\gamma(\bar t)\|}\right) \\
&= \frac{\|\gamma(t)-\gamma(\bar t)\|}{(t-\bar t)}\left(\frac{f(\gamma(t)) - f(\gamma(\bar t)) - \langle g(t), \gamma(t)-\gamma(\bar t)\rangle}{\|\gamma(t)-\gamma(\bar t)\|}\right) + \frac{\langle g(t), \gamma(t)-\gamma(\bar t)\rangle}{t-\bar t}.
\end{aligned}$$

Letting $t \to \bar t$ with $t \in E_3$, $t \neq \bar t$ and $\gamma(t) \neq \gamma(\bar t)$, which is possible because $\bar t$ has density 1 in E_3 and $\dot\gamma(\bar t) \neq 0$, we have

$$\begin{aligned}
&\frac{f(\gamma(t)) - f(\gamma(\bar t))}{(t-\bar t)} \to \frac{\mathrm{d}}{\mathrm{d}t}(f\circ\gamma)(\bar t), && \frac{\|\gamma(t)-\gamma(\bar t)\|}{(t-\bar t)} \to \|\dot\gamma(\bar t)\| \\
&\frac{f(\gamma(t)) - f(\gamma(\bar t)) - \langle g(t), \gamma(t)-\gamma(\bar t)\rangle}{\|\gamma(t)-\gamma(\bar t)\|} \to 0, && \frac{\langle g(t), \gamma(t)-\gamma(\bar t)\rangle}{t-\bar t} \to \langle g(\bar t), \dot\gamma(\bar t)\rangle,
\end{aligned}$$

where the two identities on the first line follow from the differentiability of $f\circ\gamma$ and γ at $\bar t \in E_3$, the third one stems from the semismooth property of generalized gradients (Definition 3.21) while the last one is by differentiability of γ and continuity of g restricted to E_3 at $\bar t$. We obtain that $\frac{\mathrm{d}}{\mathrm{d}t}(f\circ\gamma)(\bar t) = \langle g(\bar t), \dot\gamma(\bar t)\rangle \geq \frac{\mathrm{d}}{\mathrm{d}t}(f\circ\gamma)(\bar t) + \epsilon$, where the equality follows by the previous limit and the inequality is because $\bar t \in E_3$. This is contradictory since $\epsilon > 0$, which concludes the proof. □

Functions differentiable in the generalized sense are path-differentiable. In the semialgebraic case, both notions coincide [62], but the inclusion is strict in general.

Proposition 3.23. *Consider the closed set $C \subset [-1,1]$ defined through $C = \{1/k \ : \ k \in \mathbb{Z}, k \neq 0\} \cup \{0\}$. Then the distance function F to C is path-differentiable but not differentiable in the generalized sense.*

Proof:

It is clear that $\partial^c F$ is locally constant ($+1$ or -1) out of a closed countable set (the set C and its cut locus), hence by the characterization from Proposition 3.9, F is path differentiable.

On the other hand, $F(0) = 0$ and for all $k \in \mathbb{N}^*$, $F(1/k) = 0$ and $\partial^c F(1/k) = [-1,1]$ so that $-1 \in \partial^c F(1/k)$. The equality $\frac{F(1/k)-F(0)-\langle -1, 1/k-0\rangle}{\|1/k-0\|} = 1$ contradicts the semismoothness property for the Clarke subgradient of F. Since F is differentiable in the generalized sense if and only if its Clarke subgradient is a generalized gradient, we conclude that F is not differentiable in the generalized sense at 0. □

On the size and cardinality of the nondifferentiability set. Both semismooth and conservative gradients have the property to be gradient almost everywhere with respect to Lebesgue (see [142, Theorem A.2] or [114] for semismooth gradients).

Some differences can be noticed in dimension one. For a semismooth derivative D_F of some function $F : \mathbb{R} \to \mathbb{R}$, the set where D_F is not a singleton, which is equal to the set $\{w \in \mathbb{R} : D_F(w) \neq F'(w)\}$, is countable and in particular, it has Hausdorff dimension zero [142, Theorem A.2].

A conservative derivative can exhibit more pathological behavior. Not only this set can be uncountable, but it can also have Hausdorff dimension one:

Proposition 3.24. *There exists a locally Lipschitz and path differentiable function $f : \mathbb{R} \to \mathbb{R}$ such that $\partial^c f$ is not a singleton on an uncountable set which has furthermore Hausdorff dimension 1.*

Proof: Given a subset Z with zero Lebesgue measure, one may find a 1-Lipschitz function $h : \mathbb{R} \to \mathbb{R}$ such that h is differentiable on Z^c, and not on Z. Such a function is given by a construction from Zahorski [164], see also [129, Theorem 1.13]: let $(U_k)_{k\in\mathbb{N}}$ be a sequence of open subsets from $\mathbb{R}$ decreasing to Z for the inclusion, with $\lambda(U_k \cap [a, b]) \leq \frac{1}{2^k}(b - a)$ for all segment $[a, b] \subset \mathbb{R}$. The desired function h is then given for all $x \in \mathbb{R}$ by

$$h(x) := \sum_{k=0}^{\infty} (-1)^k \lambda(U_k \cap (-\infty, x]).$$

One can verify that h is well-defined, only differentiable outside Z with derivative equal to -1 or 1, and $\partial^c h$ is equal to $[-1, 1]$ on Z.

If now we take Z to be the countable union of modified Cantor sets whose dimensions grow to 1, see e.g [111, Section 8.2.1], Z is closed by Baire's theorem, has zero Lebesgue measure and has Hausdorff dimension 1. For x in Z^c, which is open, $\partial^c h(x)$ is a singleton equal to -1 or 1, hence h is path differentiable by the Proposition 3.9. □

Appendix

3.4.1 Conservative integral rule: missing proofs

Lemma 3.11 (Measurability of partial derivatives)**.** *Let $U \subset \mathbb{R}$ be open and $f : U \times S \to \mathbb{R}$ a $(\mathcal{B}(\mathbb{R}) \times \mathcal{A})$-measurable function. We suppose that there exists $M \subset S$ of full measure such that for all $s \in M$, $f(\cdot, s)$ is absolutely continuous. Then $\frac{\partial f}{\partial x}$ is jointly measurable and is defined almost everywhere in $U \times S$. Also, for almost all $x \in U$, $\frac{\partial f}{\partial x}(x, s)$ is defined for almost all $s \in S$.*

Proof: Define the following quantities for all $x \in U$ and $s \in M$:

$$f'_u(x,s) = \limsup_{h \to 0} \frac{f(x+h,s) - f(x,s)}{h} \text{ and } f'_l(x,s) = \liminf_{h \to 0} \frac{f(x+h,s) - f(x,s)}{h}.$$

By continuity of f both limit operators may operate only in $\mathbb{Q}$ without changing the value of f'_u and f'_l. Whence f'_l and f'_u are measurable and so is $\frac{\partial f}{\partial x}$. Furthermore, the domain E of $\frac{\partial f}{\partial x}$ is

$$\left\{(x,s) \in U \times S \ : \ f'_l(x,s) = f'_u(x,s),\ -\infty < f_u(x,s) < +\infty\right\}$$

which is measurable. Applying Fubini's Theorem yields

$$\int_{U\times S} \mathbb{1}_{E^c}(x,s)\, \mathrm{d}(\lambda \times \mu)\,(x,s) = \int_S \int_U \mathbb{1}_{E^c}(x,s)\, \mathrm{d}x\, \mathrm{d}\mu(s) = \int_U \int_S \mathbb{1}_{E^c}(x,s)\, \mathrm{d}\mu(s)\, \mathrm{d}x.$$

Since $f(\cdot, s)$ is absolutely continuous for $s \in M$, it is differentiable almost everywhere, hence we have $\forall s \in M, \int_U \mathbb{1}_{E^c}(x,s)\, \mathrm{d}x = 0$ and the second integral is zero. The third integral vanishes, so for almost all $x \in U$, $\int_S \mathbb{1}_{E^c}(x,s)\, \mathrm{d}\mu(s) = 0$, i.e., $\frac{\partial f}{\partial x}(x,s)$ is defined for almost all s, which concludes the proof. □

Proposition 3.12 (Differentiation of absolutely continuous integrals)**.** *Let $U \subset \mathbb{R}$ be open and $f : U \times S \to \mathbb{R}$ such that:*

1. *For all $x \in U$, $f(x, \cdot)$ is integrable.*
2. *For almost all $s \in S$, $f(\cdot, s)$ is absolutely continuous.*
3. *$\frac{\partial f}{\partial x}$ is locally integrable, jointly in x and s: for any compact interval $[a, b] \subset U$,*

$$\int_S \int_a^b \left| \frac{\partial f}{\partial x}(x,s) \right| \mathrm{d}x\, \mathrm{d}\mu(s) < \infty.$$

Then, the function $g : x \mapsto \int_S f(x,s)\, \mathrm{d}\mu(s)$, is absolutely continuous, differentiable at almost all $x \in U$ with $g'(x) = \int_S \frac{\partial f}{\partial x}(x,s)\, \mathrm{d}\mu(s)$.

Proof: Let $f : U \times S \to \mathbb{R}$ satisfying all the assumptions. We consider the function $g : x \in U \mapsto \int_S f(x,s)\, \mathrm{d}\mu(s)$ and $a < b$ in U. From Lemma 3.11, $\frac{\partial f}{\partial x}(x,s)$ and exists almost everywhere in $(x,s) \in U \times S$ and admits a measurable extension. The function $\frac{\partial f}{\partial x}$ defined almost everywhere is identified with some measurable extension. Since for almost all $s \in S$ $f(\cdot, s)$ is absolutely continuous, the fundamental theorem of calculus for Lebesgue integration (see Theorem 14 in Section 4, Chapter 5 of [139]) implies that

$$g(b) - g(a) = \int_S [f(b,s) - f(a,s)]\, \mathrm{d}\mu(s) = \int_S \int_a^b \frac{\partial f}{\partial t}(t,s)\, \mathrm{d}t\, \mathrm{d}\mu(s).$$

Under Assumption 3, Fubini-Lebesgue's Theorem applies and $g(b) - g(a) = \int_a^b \int_S \frac{\partial f}{\partial t}(t,s)\,\mathrm{d}\mu(s)\,\mathrm{d}t$. The function $x \mapsto \int_S \frac{\partial f}{\partial x}(x,s)\,\mathrm{d}\mu(s)$ is integrable on $[a,b]$ so g is absolutely continuous. By the fundamental theorem of calculus, $g'(x)$ is defined for almost all $x \in U$ with $g'(x) = \int_S \frac{\partial f}{\partial x}(x,s)\,\mathrm{d}\mu(s)$. □

3.4.2 Conservative implicit differentiation: proofs of machine learning applications

Monotone layers.

Proposition 3.17 (Path differentiation through monotone layers)**.** *Assume that J_σ is convex-valued and that, for all $J \in J_\sigma(Wz(W,b)+b)$, the matrix $(\mathrm{Id}_m - JW)$ is invertible. Consider a loss-like function $\ell\colon \mathbb{R}^m \to \mathbb{R}$ with conservative gradient $D_\ell\colon \mathbb{R}^m \rightrightarrows \mathbb{R}^m$, then $g : (W,z) \mapsto \ell(z(W,b))$ is path differentiable and has a conservative gradient D_g defined through*

$$D_g\colon (W,b) \rightrightarrows \left\{ J^\mathsf{T}(\mathrm{Id}_m - JW)^{-T} v z^\mathsf{T}, J^\mathsf{T}(\mathrm{Id}_m - JW)^{-T} v) :\ J \in J_\sigma(Wz+b),\, v \in D_\ell(z) \right\}.$$

Proof: The quantity $z(W,b)$ is defined implicitly by the relation

$$z(W,b) - \sigma(Wz(W,b) + b) = 0. \tag{3.17}$$

We set $M = m + m + m \times m$ and represent the pair $(W,b) \in \mathbb{R}^{m\times m} \times \mathbb{R}^m$ as $(w_1, \dots, w_m, b) \in \mathbb{R}^{M-m}$ where $w_i \in \mathbb{R}^m$ is the i-th row of W for $i \in \{1, \dots, m\}$. We denote by $\mathcal{B}\colon \mathbb{R}^M \to \mathbb{R}^m$ the bilinear map defined as

$$\mathcal{B}(w_1, \dots, w_m, b, z) := Wz + b$$

so that $\mathcal{B}$ is infinitely differentiable. Equation (3.17) is then equivalent to

$$z - (\sigma \circ \mathcal{B})(w_1, \dots, w_m, b, z) = 0.$$

We denote by F the mapping

$$F\colon (w_1, \dots, w_m, b, z) \mapsto z - (\sigma \circ \mathcal{B})(w_1, \dots, w_m, b, z).$$

For $i \in \{1 \dots m\}$, denote by $Z_i \in \mathbb{R}^{m\times m}$ the matrix whose i-th row is z, and remaining rows are null. The Jacobian of $\mathcal{B}$, $\operatorname{Jac}\mathcal{B}\colon \mathbb{R}^M \to \mathbb{R}^{m\times M}$ is as follows:

$$\operatorname{Jac}\mathcal{B}(w_1, \dots, w_m, b, z) = [Z_1 \quad \dots \quad Z_m \quad \mathrm{Id}_m \quad W]$$

where $[A\,B]$ is used to denote the columnwise concatenation of matrices A and B. By hypothesis, we have a conservative Jacobian for σ, J_σ. Conservative Jacobians may be composed as usual Jacobians [37, Lemma 5]. As $\mathcal{B}$ is continuously differentiable, $\operatorname{Jac}\mathcal{B}$ is also a conservative Jacobian for $\mathcal{B}$. Therefore, we have the following conservative Jacobian for F,

$$J_F(w_1, \dots, w_m, b, z) \rightrightarrows \left\{ [-JZ_1 \quad \dots \quad -JZ_m \quad -J \quad \mathrm{Id}_m - JW],\ J \in J_\sigma(Wz+b) \right\}.$$

Finally, by hypothesis, for any W, b, and z such that $F(W,b,z) = 0$ and any $J \in J_\sigma(Wz+b)$, the matrix $\mathrm{Id}_m - JW$ is invertible. Therefore, Theorem 3.14 applies and, setting $\tilde{M} = m \times m + m = M - m$, the set-valued mapping

$$\begin{aligned} J_z\colon \mathbb{R}^{\tilde{M}} &\rightrightarrows \mathbb{R}^{m\times\tilde{M}} \\ (w_1, \dots, w_m, b) &\rightrightarrows \left\{ (\mathrm{Id}_m - JW)^{-1} J\,[Z_1 \quad \dots \quad Z_m \quad \mathrm{Id}_m],\ J \in J_\sigma(Wz+b) \right\} \end{aligned}$$

is conservative for $(W, b) \mapsto z(W, b)$ as defined in (3.17). We denote by $Z \in \mathbb{R}^{m \times \tilde{M}}$ the matrix $[Z_1 \dots Z_m \ \mathrm{Id}_m]$ appearing in the definition of J_z. Given the loss function ℓ, the mapping $J_\ell \colon z \mapsto \{v^\mathsf{T},\, v \in D_\ell(z)\}$ is a conservative Jacobian for ℓ [37, Lemma 3] and therefore, the set-valued mapping

$$\begin{aligned} J_g \colon \mathbb{R}^{\tilde{M}} &\rightrightarrows \mathbb{R}^{1 \times \tilde{M}} \\ (w_1, \dots, w_m, b) &\rightrightarrows \left\{ v^\mathsf{T} (\mathrm{Id}_m - JW)^{-1} JZ,\ J \in J_\sigma(Wz + b),\ v \in D_\ell(z(W, b)) \right\} \end{aligned}$$

is a conservative Jacobian for $g \colon (W, b) \mapsto \ell(z(W, b))$. Using [37, Lemma 4], we obtain a conservative gradient field for g by a simple transposition as follows

$$D_g \colon (w_1, \dots, w_m, b) \rightrightarrows \left\{ Z^\mathsf{T} J^\mathsf{T} (\mathrm{Id}_m - JW)^{-T} v,\ J \in J_\sigma(Wz + b),\ v \in D_\ell(z(W, b)) \right\}.$$

We now identify the terms by block computation; recall that $Z = [Z_1 \ \dots \ Z_m \ \mathrm{Id}_m]$ and that $Z_i \in \mathbb{R}^{m \times m}$ is the matrix whose i-th row is z with remaining rows null for each $i \in \{1, \dots, m\}$. The term associated to b corresponds to the last $m \times m$ block in Z, it is indeed of the form $J^\mathsf{T}(\mathrm{Id}_m - JW)^{-T} v$. Similarly, for each $i \in \{1, \dots, m\}$, the term associated to w_i is of the form $Z_i^\mathsf{T} J^\mathsf{T} (\mathrm{Id}_m - JW)^{-T} v$. For any $a \in \mathbb{R}^m$ and $i \in \{1, \dots, m\}$, we have $Z_i^\mathsf{T} a = a_i z$ where a_i is the i-th coordinate of a and z corresponds to the i-th row of Z_i^T. So the component associated to w_i in D_g is of the form $[J^\mathsf{T}(\mathrm{Id}_m - JW)^{-T} v]_i z$, where $[\cdot]_i$ denotes the i-th coordinate. Since w_i denotes the i-th row of W, rearranging this expression in matrix format provides a term of the form $J^\mathsf{T}(\mathrm{Id}_m - JW)^{-T} v z^\mathsf{T}$ for the W component. This concludes the proof. □

Cone programming layer. We show $\phi(u, v)$ provides a primal-dual KKT solution of problems (P) and (D) if and only if $\mathcal{N}(z, A, b, c) = 0$. Let us first expand on the link between the zeros of the residual map and KKT solutions. We provide a simplified view of [3, 50], ignoring cases of infeasibility and unboundedness. Note that this corresponds to enforcing $w = 1$ as done in [2, 3].

The following is due to Moreau [116]. Recall that the polar of a closed convex cone $\mathcal{K} \subset \mathbb{R}^m$ is given by $\mathcal{K}^\circ = \left\{ x \in \mathbb{R}^m,\ y^T x \leq 0,\ \forall y \in \mathcal{K} \right\}$, in which case $(\mathcal{K}^\circ)^\circ = \mathcal{K}$ and the dual cone satisfies $\mathcal{K}^* = -\mathcal{K}^\circ$.

Proposition 3.25. *Let $s, y, v \in \mathbb{R}^m$; the following are equivalent*

- $v = s + y,\ s \in \mathcal{K},\ y \in \mathcal{K}^\circ,\ s^T y = 0.$
- $s = P_{\mathcal{K}}(v),\ y = P_{\mathcal{K}^\circ}(v).$

We may reformulate this equivalence as follows, using changes of signs on y and v, noticing that $-P_{\mathcal{K}^\circ}(-\cdot) = P_{\mathcal{K}^*}(\cdot)$ since $\mathcal{K}^* = -\mathcal{K}^\circ$,

(i) $v = y - s, \quad s \in \mathcal{K}, \quad y \in \mathcal{K}^*, \quad s^T y = 0.$

(ii) $s = P_{\mathcal{K}^*}(v) - v, \quad y = P_{\mathcal{K}^*}(v).$

Now the KKT system in (x, y, s) for the problem (P) and (D) can be written as follows (see, for example, [50]),

$$\begin{aligned} A^T y + c &= 0, \quad y \in \mathcal{K}^* \\ -Ax + b &= s, \quad s \in \mathcal{K} \\ s^T y &= 0 \end{aligned}$$

which is equivalent, by setting $v = y - s$ and $u = x$, to

$$\begin{aligned} A^T P_{\mathcal{K}^*}(v) + c &= 0 \\ -Au + b &= P_{\mathcal{K}^*}(v) - v \end{aligned} \tag{3.18}$$

The system (3.18) is equivalent to $\mathcal{N}(z, A, b, c) = 0$ with $z = (u, v)$. We have shown that (x, y, s) is a KKT solution to the system if and only if $(x, y, s) = (u, P_{\mathcal{K}^*}(v), P_{\mathcal{K}^*}(v) - v) = \phi(z)$ for $z = (x, y - s)$ such that $\mathcal{N}(z, A, b, c) = 0$.

Proposition 3.19 (Path differentiation through cone programming layers)**.** *Assume that $P_{\mathcal{K}^*}$, $\mathcal{N}$ are path differentiable, denote respectively by $J_{P_{\mathcal{K}^*}}$, $J_{\mathcal{N}}$ corresponding convex-valued conservative Jacobians. Assume that, for all $A, b, c \in \mathbb{R}^{m\times n} \times \mathbb{R}^m \times \mathbb{R}^n$, $z = \nu(A, b, c) \in \mathbb{R}^n \times \mathbb{R}^m$ is the unique solution to $\mathcal{N}(z, A, b, c) = 0$ and that all matrices formed from the N first columns of $J_{\mathcal{N}}(z, A, b, c)$ are invertible. Then, ϕ, ν, and* sol *are path differentiable functions with conservative Jacobians:*

$$\begin{aligned} J_\nu(A, b, c) &:= \left\{ -U^{-1}V : [U\ V] \in J_{\mathcal{N}}(\nu(A, b, c), A, b, c) \right\}, \\ J_\phi(z) &:= \begin{bmatrix} \mathrm{Id}_n & 0 \\ 0 & J_{P_{\mathcal{K}^*}}(v) \\ 0 & (J_{P_{\mathcal{K}^*}}(v) - \mathrm{Id}_m) \end{bmatrix}, \\ J_{\mathrm{sol}}(A, b, c) &:= J_\phi(\nu(A, b, c)) J_\nu(A, b, c). \end{aligned}$$

Proof: First, the assumptions clearly ensure that ν and sol are single-valued and can be interpreted as functions such that $\mathrm{sol} = \phi \circ \nu$. By assumption, ϕ is differentiable. We will first use Corollary 3.15 to obtain a conservative Jacobian for ν and then justify the expression for ϕ. The composition obtained for J_{sol} results from Proposition 3.4.

Let $A, b, c \in \mathbb{R}^{m\times n} \times \mathbb{R}^m \times \mathbb{R}^n$, $z := (u, v) \in \mathbb{R}^n \times \mathbb{R}^m$ such that $\mathcal{N}(z, A, b, c) = 0$. By assumption, the submatrices formed from the first N columns of $J_{\mathcal{N}}(z, A, b, c)$ are invertible. Then applying Corollary 3.15, there exist open neighborhoods $\mathcal{U} \subset \mathbb{R}^{m\times n} \times \mathbb{R}^m \times \mathbb{R}^n$ and $\mathcal{V} \subset \mathbb{R}^N$ and a locally Lipschitz function $G : \mathcal{U} \to \mathcal{V}$ satisfying, for all $s \in \mathcal{U}$ $\mathcal{N}(G(s), s) = 0$ with G is path differentiable. Since, by assumption, the solution $\nu(A, b, c)$ to $\mathcal{N}(\nu(A, b, c), A, b, c) = 0$ is unique, ν coincides with G on $\mathcal{U}$. Thus, ν is path differentiable and a conservative Jacobian for ν is given by:

$$J_\nu(A, b, c) = \left\{ -U^{-1}V : [U\ V] \in J_{\mathcal{N}}(\nu(A, b, c), A, b, c) \right\}$$

Let us now turn to ϕ. Since $P_{\mathcal{K}^*}$ has for conservative Jacobian $J_{P_{\mathcal{K}^*}}$, we may construct a conservative Jacobian for the function ϕ as follows using [37, Lemmas 3, 4, and 5]:

$$J_\phi(z) = \begin{bmatrix} \mathrm{Id}_n & 0 \\ 0 & J_{P_{\mathcal{K}^*}}(v) \\ 0 & (J_{P_{\mathcal{K}^*}}(v) - \mathrm{Id}_m) \end{bmatrix}.$$

It follows from Proposition 3.4 that the composition $\mathrm{sol} = \phi \circ \nu$ is also path differentiable with conservative Jacobian

$$J_{\mathrm{sol}}(A, b, c) = J_\phi(\nu(A, b, c)) J_\nu(A, b, c).$$

□

Hyperparameter selection of Lasso type problems.

Proposition 3.20 (Conservative Jacobian for the solution mapping)**.** *For all $\lambda \in \mathbb{R}$, assume $X_{\mathcal{E}}^{\mathsf{T}} X_{\mathcal{E}}$ is invertible where $X_{\mathcal{E}}$ is the submatrix of X formed by taking the columns indexed by $\mathcal{E}$. Then $\hat{\beta}(\lambda)$ is single-valued, path differentiable with conservative Jacobian, $J_{\hat{\beta}}(\lambda)$, given for all λ as*

$$\left\{ \left[-e^{\lambda} \left(\mathrm{Id}_p - \mathrm{diag}\,(q) \left(\mathrm{Id}_p - X^{\mathsf{T}} X \right) \right)^{-1} \mathrm{diag}\,(q)\, \mathrm{sign}\left(\hat{\beta} - X^{\mathsf{T}} \left(X\hat{\beta} - y \right) \right) \right] \ : q \in \mathcal{M}(\lambda) \right\}$$

where $\mathcal{M}(\lambda) \subset \mathbb{R}^p$ is the set of vectors q such that $q_i = 1$ if $i \in \operatorname{supp} \hat{\beta}$, $q_i = 0$ if $i \notin \mathcal{E}$ and $q_i \in [0,1]$ if $i \in \mathcal{E} \setminus \operatorname{supp} \hat{\beta}$.

Proof: Our goal is to apply Corollary 3.15 to the path differentiable "optimality gap" function $F : \mathbb{R} \times \mathbb{R}^p \to \mathbb{R}^p$ defined in (3.3.3). For each $\lambda \in \mathbb{R}$, the invertibility of $X_{\mathcal{E}}^{\mathsf{T}} X_{\mathcal{E}}$ guarantees the uniqueness of $\hat{\beta}(\lambda)$ (see [125], [107, Lemma 1]), i.e., $\hat{\beta} : \mathbb{R} \to \mathbb{R}^p$ is a function. Because $\|\cdot\|_1$ is separable, the components of the prox can be written, for any $(\lambda, u) \in \mathbb{R} \times \mathbb{R}^p$, for all $i \in \{1, \ldots, p\}$, as

$$[\mathrm{prox}_{e^{\lambda}\|\cdot\|_1}(u)]_i = \mathrm{prox}_{e^{\lambda}|\cdot|}(u_i)$$

which have Clarke subdifferentials

$$\partial^c \mathrm{prox}_{e^{\lambda}|\cdot|} : u_i \rightrightarrows \mathbf{1}_{u_i, e^{\lambda}} \quad \times \begin{bmatrix} 1 \\ -\mathrm{sign}(u_i) \end{bmatrix} \quad \text{where} \quad \mathbf{1}_{e^{\lambda}}(u_i) := \begin{cases} 0 & |u_i| < e^{\lambda} \\ [0,1] & |u_i| = e^{\lambda} \\ 1 & |u_i| > e^{\lambda} \end{cases}.$$

Thus a conservative Jacobian for F at (λ, β) is given by

$$J_F : (\lambda, \beta) \rightrightarrows \{ [\underbrace{e^{\lambda} \mathrm{diag}(q)\, \mathrm{sign}(\beta - X^{\mathsf{T}}(X\beta - y))}_{A} \quad \underbrace{\mathrm{Id}_p - \mathrm{diag}(q) \left(\mathrm{Id}_p - X^{\mathsf{T}} X \right)}_{B}] : q \in \mathcal{C} \} \tag{3.19}$$

with $\mathcal{C} := \{ q : q_i \in \mathbf{1}_{e^{\lambda}} \left(\beta_i - X_i^{\mathsf{T}} (X\beta - y) \right) \}$. Let us estimate the factors q_i above in terms of the equicorrelation set $\mathcal{E}$. Recall the KKT conditions [153] for the Lasso problem; a solution $\hat{\beta}$ must satisfy

$$X^{\mathsf{T}} \left(y - X\hat{\beta} \right) = e^{\lambda} \delta \qquad \text{where} \qquad \delta_i \in \begin{cases} \left\{ \mathrm{sign}\left(\hat{\beta}_i \right) \right\} & i \in \operatorname{supp} \hat{\beta} \\ [-1, 1] & i \notin \operatorname{supp} \hat{\beta} \end{cases}. \tag{3.20}$$

For $i \in \operatorname{supp} \hat{\beta}$, (3.20) gives

$$\begin{aligned} X_i^{\mathsf{T}} \left(y - X\hat{\beta} \right) = e^{\lambda} \mathrm{sign}\left(\hat{\beta}_i \right) &\implies \mathrm{sign}\left(X_i^{\mathsf{T}} \left(y - X\hat{\beta} \right) \right) = \mathrm{sign}\left(\hat{\beta}_i \right) \\ &\implies \mathrm{sign}\left(\hat{\beta}_i \right) = \mathrm{sign}\left(\hat{\beta}_i - X_i^{\mathsf{T}} \left(X\hat{\beta} - y \right) \right) \\ &\qquad = \mathrm{sign}\left(X_i^{\mathsf{T}} \left(y - X\hat{\beta} \right) \right). \end{aligned}$$

Noting that $\left|\hat{\beta}_i\right| > 0$ and $\left|X_i^\mathsf{T}\left(y - X\hat{\beta}\right)\right| = e^\lambda$ since $i \in \operatorname{supp}\hat{\beta} \subset \mathcal{E}$,

$$\begin{aligned}\left|\hat{\beta}_i - X_i^\mathsf{T}\left(X\hat{\beta} - y\right)\right| &= \operatorname{sign}\left(\hat{\beta}_i - X_i^\mathsf{T}\left(X\hat{\beta} - y\right)\right)\left(\hat{\beta}_i - X_i^\mathsf{T}\left(X\hat{\beta} - y\right)\right) \\ &= \operatorname{sign}\left(\hat{\beta}_i\right)\hat{\beta}_i + \operatorname{sign}\left(X_i^\mathsf{T}\left(y - X\hat{\beta}\right)\right)X_i^\mathsf{T}\left(y - X\hat{\beta}\right) \\ &= \underbrace{\left|\hat{\beta}_i\right|}_{>0} + \underbrace{\left|X_i^\mathsf{T}\left(y - X\hat{\beta}\right)\right|}_{=e^\lambda} \\ &\implies q_i = 1.\end{aligned}$$

For $i \notin \mathcal{E}$, $\hat{\beta}_i = 0$ since $\operatorname{supp}\hat{\beta} \subset \mathcal{E}$. By (3.20), we have $\left|X_i^\mathsf{T}\left(y - X\hat{\beta}\right)\right| \leq e^\lambda$. However, since $i \notin \mathcal{E}$, the inequality is strict

$$\left|X_i^\mathsf{T}\left(y - X\hat{\beta}\right)\right| < e^\lambda$$

and can be used to solve for q_i

$$\left|\hat{\beta}_i - X_i^\mathsf{T}\left(X\hat{\beta} - y\right)\right| = \left|X_i^\mathsf{T}\left(y - X\hat{\beta}\right)\right| < e^\lambda \implies q_i = 0.$$

Finally, for $i \in \mathcal{E} \setminus \operatorname{supp}\hat{\beta}$, $\hat{\beta}_i = 0$ and $\left|X_i^\mathsf{T}\left(X\hat{\beta} - y\right)\right| = e^\lambda$ which gives

$$\left|\hat{\beta}_i - X_i^\mathsf{T}\left(X\hat{\beta} - y\right)\right| = \left|X_i^\mathsf{T}\left(X\hat{\beta} - y\right)\right| = e^\lambda$$

and thus $q_i \in [0, 1]$. Putting everything together we get an expression for q_i in terms of $\mathcal{E}$ and $\operatorname{supp}\hat{\beta}$

$$q_i \in \begin{cases}\{1\} & i \in \operatorname{supp}\hat{\beta} \\ [0,1] & i \in \mathcal{E} \setminus \operatorname{supp}\hat{\beta} \\ \{0\} & i \notin \mathcal{E}\end{cases}, \tag{3.21}$$

i.e., $q \in \mathcal{M}$. We proceed to show that B is invertible for all $\lambda \in \mathbb{R}$. Denote $Q := \operatorname{diag}(q)$ for brevity; using the same argument of [162, Theorem 2] involving similarity transformations and continuity, the matrix B is invertible if and only if

$$\tilde{B} := \operatorname{Id}_p - Q^{1/2}\left(\operatorname{Id}_p - X^\mathsf{T}X\right)Q^{1/2} = \operatorname{Id}_p - Q + Q^{1/2}X^\mathsf{T}XQ^{1/2}$$

is invertible. Since $\tilde{B} \succeq \operatorname{Id}_p - Q$, it follows that $\ker\left(\tilde{B}\right) \subset \ker\left(\operatorname{Id}_p - Q\right)$, however $\ker\left(\operatorname{Id}_p - Q\right)$ is a subspace of $W_\mathcal{E} := \operatorname{Span}\{e_j : j \in \mathcal{E}\}$ corresponding to $q_j = 1$. Since $q_j = 1 \implies j \in \mathcal{E}$ by (3.21), the restriction of $\tilde{B}$ to $\ker\left(\operatorname{Id}_p - Q\right)$ is a principal submatrix of (possibly equal to) $X_\mathcal{E}^\mathsf{T}X_\mathcal{E}$ which is invertible by assumption. Thus B is invertible and applying Corollary 3.15 then yields the final result. □

3.4.3 Pathological examples: details on the experiments

Here, we present some details regarding the toy experiments.

Cyclic gradient descent.

Fixed-point formulation.

Consider the optimization problem

$$(s_1, s_2) \in \underset{(a,b)\in[0,3]\times[0,5]}{\arg\max} (a+b)(-3x+y+2). \tag{3.22}$$

The optimality condition for this problem can be expressed using the fixed-point equation of the projected gradient descent algorithm. Denote for $x, y \in \mathbb{R}^2$, $q_{x,y} : (a, b) \mapsto (a+b)(-3x+y+2)$; we can verify (s_1, s_2) is solution to (3.12) if and only if it satisfies the equality

$$\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = P_{\mathcal{U}} \left(\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \nabla q_{x,y}(s_1, s_2) \right) = P_{\mathcal{U}} \left(\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} -3x+y+2 \\ -3x+y+2 \end{bmatrix} \right).$$

Where $P_{\mathcal{U}}$ is the projection on the set $\mathcal{U} := [0, 3]\times[0, 5]$ which can be implemented as a difference of relu functions

$$P_{\mathcal{U}}(x, y) = \mathrm{relu}(x, y) - \mathrm{relu}(x - 3, y - 5).$$

Let $h : \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^2$ be the function

$$h : (s, x, y) \mapsto P_{\mathcal{U}} \left(\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} -3x+y+2 \\ -3x+y+2 \end{bmatrix} \right).$$

Then the original problem (3.22) is equivalent to the fixed point equation $s = h(x, y, s)$. Indeed, we can easily verify the solutions $s : \mathbb{R}^2 \to \mathbb{R}^2$ to (3.22) are

$$s(x, y) = \begin{cases} \{(0, 0)\} & \text{if } -3x+y+2 < 0 \\ \{(3, 5)\} & \text{if } -3x+y+2 > 0 \\ [0, 3] \times [0, 5] & \text{if } -3x+y+2 = 0 \end{cases}$$

which creates a discontinuity for the function $\ell(\cdot, s(\cdot))$, now expressed as

$$\ell(x, y, s(x, y)) = \begin{cases} x^2 + 4y^2 & \text{if } -3x+y+2 < 0 \\ (x-3)^2 + 4(y-5)^2 & \text{if } -3x+y+2 > 0 \end{cases}.$$

On the perturbed experiments. Perturbed experiments are done on the following perturbed loss function

$$\ell_\varepsilon(x, y, s) = \left(\frac{1}{4} + \varepsilon_1\right)(x - s_1)^2 + (1 + \varepsilon_2)(y - s_2)^2$$

$$s \in s_\varepsilon(x, y) := \arg\max \{(a+b)(-(3+\varepsilon_3)x + y + 2 + \varepsilon_4) : a \in [0, 3-\varepsilon_5], b \in [0, 5-\varepsilon_6]\}$$

with $\varepsilon_1, \ldots, \varepsilon_6$ the perturbations. In Figure 3.2b, we consider several realizations of independent Gaussian variables $\varepsilon_1, \ldots, \varepsilon_6 \sim \mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 0.05$; despite this added noise, the unwanted dynamics persist.

Conic canonicalization. Let $c \in \mathbb{R}^2$ be a parameter vector and consider the problem

$$\max_{x\in[0,3]\times[0,5]} \quad c^\mathsf{T} x.$$

It can be formulated as a cone program (P) and its dual (D):

$$\begin{array}{lll} \text{(P)} & \inf & c^\mathsf{T} x \\ & \text{subject to} & Ax + s = b \\ & & s \in \mathcal{K} \end{array} \qquad \begin{array}{lll} \text{(D)} & \inf & b^\mathsf{T} y \\ & \text{subject to} & A^\mathsf{T} y + c = 0 \\ & & y \in \mathcal{K}^*, \end{array} \tag{3.23}$$

where

$$A = \begin{bmatrix} \mathrm{Id}_2 \\ -\mathrm{Id}_2 \end{bmatrix} \text{ and } b = \begin{bmatrix} 3 \\ 5 \\ 0 \\ 0 \end{bmatrix}.$$

Let (x, y, s) be a solution to the cone program (3.23) where x is the primal variable, y is the dual variable, and s the primal slack variable. Then it follows from (3.10) that a solution z to $\mathcal{N}(z, c) = 0$ is obtained by $z = (x, y - s)$. For $c = (0, 0)$, the solutions are $x \in [0, 3] \times [0, 5]$, $s = b - Ax$, and $y = (0, 0, 0, 0)$, hence the uniqueness assumption for Proposition 3.19 is not satisfied.

Quadratic form for the Lorenz-like attractor.

Set $u = (x, y, z)$, then

$$\begin{aligned} u^\mathsf{T} F(u) &= \sigma x(y - x) + xy(\rho - z) - y^2 + xyz - \beta z^2 \\ &= -\sigma x^2 - y^2 - \beta z^2 + (\sigma + \rho)xy \\ &= \frac{1}{2} u^\mathsf{T} H u \end{aligned}$$

where $H = \begin{bmatrix} -2\sigma & \sigma + \rho & 0 \\ \sigma + \rho & -2 & 0 \\ 0 & 0 & -2\beta \end{bmatrix}$.

For $(\sigma, \rho, \beta) = (10, 28, \frac{8}{3})$, g has for unique critical point $(0, 0, 0)$ which is a strict saddle-point.

Chapter 4

Analysis of nonsmooth nonconvex stochastic first-order methods

In order to study nonsmooth algorithms, we focus on the differential inclusion approach introduced in [21, 29]. This analysis technique conceptualizes algorithms as discrete-time approximations of limiting dynamical systems that are driven by a set-valued vector field, the so-called differential inclusions. In this framework, a nonsmooth algorithm is then a discrete-time recursion that asymptotically inherits from the behavior of a limiting differential inclusion.

We gather the main outcomes of this method under a condensed form, which is Theorem 4.8. The general approach [21] allows convergence results under a Sard property, which we obtain by definable integration presented in Chapter 2. Outside this case, the closed-measure approach [29] allows for a weaker convergence result.

After exposing our general nonsmooth stochastic setting, we apply the differential inclusion framework to the stochastic subgradient method and its heavy ball version. We recover and complete the results from former works, [63, 72] for the subgradient method, [29, 142] for the heavy ball method, in particular justifying first-order sampling and stating results outside the Sard property.

Our setting is built upon the conservative calculus developed in chapter 3 in order to justify first-order sampling and practical implementation, e.g. nonsmooth backpropagation and implicit differentiation.

We finally study the question of the artifacts avoidance for the subgradient method in the definable case and for the heavy ball method.

4.1 The differential inclusion method

4.1.1 Existence theory of differential inclusions

We present the main theory of the existence of solutions for differential inclusions. In this part, $H : \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ is a set-valued map that is graph-closed, locally bounded, and convex-valued. We consider the differential inclusion

$$\dot{x}(t) \in H(x(t)) \quad \text{for almost all } t \geq 0. \tag{DI}$$

For an interval $U \subset \mathbb{R}_+$, we call solution of the differential inclusion (DI), an absolutely continuous map $x : U \to \mathbb{R}^p$ satisfying the inclusion (DI) for almost all $t \geq 0$. x is called a global solution if $U = \mathbb{R}_+$.

A first result gives the existence of a local solution for (DI)

Theorem 4.1 (Local existence [9, Theorem 1]). *Let $x_0 \in \mathbb{R}^p$, assume $H : \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ is graph-closed, locally bounded and convex-valued. Then there exists $\delta > 0$ such that* (DI) *has a solution x on $[0, \delta]$ with $x(0) = x_0$.*

The following theorem allows to continue a local solution as long as the explosion in finite time doesn't happen:

Theorem 4.2 (Extension of a local solution [9, Theorem 2]). *Let $T > 0$ and $r > 0$. Assume $x : [0, \delta] \to \mathbb{R}^p$ is a solution of* (DI) *for some $\delta < T$ such that $x([0, \delta]) \subset \overline{B(0, r)}$. Then x can be extended to a solution on $[0, t]$ for some $t > 0$ such that $t = T$ or $\|x(t)\| = r$.*

We then define the *set-valued flow* Φ_t for all $w_0 \in \mathbb{R}^p$ and $t \in \mathbb{R}_+$ as

$$\Phi_t(w_0) := \{w(t) \ : \ w \text{ is a solution of (DI) with } w(0) = w_0\}.$$

Given a subset $A \subset \mathbb{R}^p$, we use the notation $\Phi(A)$ to denote the set of global solutions starting from A.

4.1.2 Stochastic algorithms as perturbed solutions

We summarize the main elements from the differential inclusion approaches [21, 29] leading to convergence results for first-order algorithms. Let $H : \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ be graph-closed, locally bounded, and convex-valued such that solutions to (DI) exist from any point of $\mathbb{R}^p$. We consider stochastic iterative algorithms of the form

$$x_{k+1} \in x_k + \alpha_k(H^{\delta_k}(x_k) + u_k) \quad \text{for } k \in \mathbb{N}. \tag{4.1}$$

$(u_k)_{k\in\mathbb{N}}$ is a sequence of random variables adapted to some filtration $(\mathcal{F}_k)_{k\in\mathbb{N}}$ on a probability space $(S, \mathcal{A}, P)$, and $\mathbb{E}[u_{k+1}|\mathcal{F}_k] = 0$ for all $k \in \mathbb{N}$. Note that the term "almost surely" will refer to the randomness of the whole sequence $(u_k)_{k\in\mathbb{N}}$.

For $\delta > 0$, $H^\delta : \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ is the fattened map

$$H^\delta(x) := \left\{z \in \mathbb{R}^p \ : \ \exists (x', z') \in \mathbb{R}^p, \|x - x'\| \le \delta, \|z - z'\| \le \delta, z' \in H(x')\right\}.$$

Interpolated process. The discrete sequence $(x_k)_{k\in\mathbb{N}}$ can be studied as a continuous time process through its piecewise affine interpolation $\bar{x} : \mathbb{R}_+ \to \mathbb{R}^p$. We recall its construction:

Let $\tau_0 = 0$ and for all $k \in \mathbb{N}^*$, $\tau_k := \sum_{i=0}^{k-1} \alpha_i$. Then define the continuous function $\bar{x}$ so that for each $k \in \mathbb{N}$, $\bar{x}(\tau_k) = x_k$ and $\bar{x}$ is affine on $[\tau_k, \tau_{k+1}]$.

Note that the limit sets of the discrete and continuous-time versions are equal. The differential inclusion method aims to study this interpolated process as a *perturbed solution* of (DI):

Definition 4.3 (Perturbed solution). *An absolutely continuous function x is called perturbed solution of* (DI) *if there exist a locally integrable function $U : \mathbb{R}_+ \to \mathbb{R}^p$ satisfying*

$$\lim_{t\to\infty} \sup_{v\in[0,T]} \left\| \int_t^{t+v} U(s)\,\mathrm{d}s \right\| = 0,$$

and a positive function $\delta : \mathbb{R}_+ \to \mathbb{R}_+$ satisfying $\delta(t) \to 0$ as $t \to \infty$, such that for almost all $t > 0$, $\dot{x}(t) \in H^{\delta(t)}(x(t)) + U(t)$.

Note that this definition doesn't involve randomness and applies to the realizations of the stochastic process. In [21], the relation of the discrete dynamic to the continuous time one is actually made more precise for bounded sequences through the notion of *asymptotic pseudo trajectory* (APT):

Definition 4.4 (Asymptotic pseudo trajectory)**.** *An absolutely continuous function $x : \mathbb{R}_+ \to \mathbb{R}^p$ is called an asymptotic trajectory (APT) of* (DI) *if for all $T > 0$,*

$$\lim_{t\to\infty} \inf_{z\in\Phi(\mathbb{R}^p)} \sup_{s\in[0,T]} \|x(t+s) - z(s)\|.$$

Due to the discontinuity of the set-valued flow, this definition of APT slightly differs from its smooth version from [19, 20]. Indeed, the paths $z \in S$ approximating the APT x don't necessarily start from a point of the path $x(\mathbb{R}_+)$.

The limit sets of APTs are then characterized through the notion of chain-transitive set for which the points are connected by a chain of solutions:

Definition 4.5 (Internally chain-transitive set)**.** *A subset $A \subset \mathbb{R}^p$ is called internally chain transitive for* (DI) *if it is compact, and for all $(x, y) \in A \times A$, it holds that for all $\epsilon > 0$ and $T > 0$, there exist $n \in \mathbb{N}$, solutions $x_1, \dots, x_n$ to* (DI) *and times $t_1, \dots, t_n > T$ satisfying*

- *for all $i = 1, \dots, n$, $x_i([0, t_i]) \subset A$,*
- *for all $i = 1, \dots, n-1$, $\|x_i(t_i) - x_{i+1}(0)\| \leq \epsilon$,*
- *$\|x_1(0) - x\| \leq \epsilon$ and $\|x_n(t_n) - y\| \leq \epsilon$.*

In this case, A is invariant.

On the closed-measure approach. The interpolated process $\bar{x}$ can also be studied as under its occupation measure form,

$$\mu(t) = \frac{1}{t} \int_0^t \delta_{\bar{x}(s)} \, \mathrm{d}s,$$

where δ_x is the Dirac mass at $x \in \mathbb{R}^p$. This leads to a weak notion of APT [22]. It was proved in [22] that the occupation measures converge to the invariant measure of the flow. Finally, the Poincaré recurrence theorem states that the invariance measure is supported on the *Birkhoff center*, which is the closure of recurrent points, hence relating the occupation measures limit to the stable points of the flow. In the case of a Lyapunov system, e.g. a subgradient flow, this leads to a form of weak convergence result since the recurrent points can be shown to be in the critical set.

This approach was generalized for set-valued dynamics in [29, 75]. In [29, 41], the authors give a more algorithmic interpretation of this convergence through the notion of *essential accumulation points*:

Definition 4.6 (Essential accumulation point)**.** *An accumulation point $w^* \in \mathbb{R}^p$ is called essential if for every neighborhood U of w^* one has*

$$\limsup_{k\to\infty} \frac{\sum_{i=0}^k \alpha_i \mathbb{1}_{\{w_i \in U\}}}{\sum_{i=0}^k \alpha_i} > 0 \text{ almost surely.}$$

Intuitively, non-essential accumulation points are hardly ever seen.

Convergence in the Lyapunov case. We also recall the definition of a Lyapunov function for a set-valued flow from [21]:

Definition 4.7 (Lyapunov function)**.** *A continuous function E is a Lyapunov function for a set $S \subset \mathbb{R}^p$ and for the dynamical system* (DI) *if*

$$\forall x \in \mathbb{R}^p \setminus S\ , \forall t > 0, \forall y \in \Phi_t(x),\ E(y) < E(x),$$
$$\forall x \in S,\ \forall t \geq 0, \forall y \in \Phi_t(x),\ E(y) \leq E(x).$$

We make the following assumptions:

Assumption 1.

1. $\sup_{k\in\mathbb{N}} \|x_k\| < \infty$ *almost surely,*
2. *For all $k \in \mathbb{N}$, $\alpha_k > 0$, $\alpha_k \to 0$ as $k \to \infty$ and $\sum_{k=0}^{\infty} \alpha_k = \infty$,*
3. *(Noise extinction) Almost surely, $\sup_{k\in\mathbb{N}} \mathbb{E}[\|u_{k+1}\|^2 | \mathcal{F}_k] < \infty$ and*

$$\forall T > 0, \qquad \lim_{k\to\infty} \sup_{m\geq k} \left\{ \left\| \sum_{i=k}^{m} \alpha_i u_i \right\|, \quad \text{s.t.} \sum_{i=k}^{m} \alpha_i \leq T \right\} = 0.$$

We gather the key results from [21] and [29] into the following theorem which provides convergence results for first-order methods having a Lyapunov function.

Theorem 4.8. *Let $(x_k)_{k\in\mathbb{N}}$ be defined as* (4.1) *and $S \subset \mathbb{R}^p$ such that E is a Lyapunov function for S and* (DI)*. Under Assumption 1, almost surely,*

1. *The accumulation points of $(x_k)_{k\in\mathbb{N}}$ is a connected set,*
2. *Any accumulation point x^* of $(x_k)_{k\in\mathbb{N}}$ such that $E(x^*) = \liminf_{k\to\infty} E(x_k)$ belongs to S.*
3. *Any essential accumulation point of $(x_k)_{k\in\mathbb{N}}$ belongs to S.*
4. *If $E(S)$ has empty interior, then all accumulation point belongs to S and $E(x_k)$ converges as $k \to \infty$.*

Proof: *Statement 1.* As H is locally bounded, and $(x_k)_{k\in\mathbb{N}}$ is almost surely bounded, $H^{\delta_k}(x_k)$ is bounded almost surely. As $\alpha_k \to 0$ by Assumption 1, it follows that $\alpha_k \|H^{\delta_k}(x_k)\|$, goes to 0 as $k \to \infty$, where we recall $\|H^{\delta_k}(x_k)\| = \sup_{y\in H^{\delta_k}(x_k)} \|y\|$. By Assumption 1.3, $\|\alpha_k u_k\|$ goes to 0 as $k \to \infty$, hence since $\|x_{k+1} - x_k\| \leq \alpha_k \|H^{\delta_k}(x_k)\| + \|\alpha_k u_k\|$, it holds that $\|x_{k+1} - x_k\|$ goes to 0. This shows that the accumulation points of $(x_k)_{k\in\mathbb{N}}$ is a connected set.

Statement 2. By [21, Theorem 4.2], $(x_k)_{k\in\mathbb{N}}$ taken as its interpolation is an APT of (DI). By [21, Theorem 4.3] the accumulation points of $(x_k)_{k\in\mathbb{N}}$ is internally chain transitive and hence invariant. We deduce the result by the definition of the Lyapunov function E and the invariance of the accumulation set. See for instance the proof of [21, Proposition 3.27].

Statement 3. Under Assumption 1, we are in the setting of [29]. The statement is merely an application of [29, Corollary 4.9].

Statement 4. By [21, Theorem 4.3] the accumulation points of $(x_k)_{k\in\mathbb{N}}$ is internally chain transitive, hence we may obtain the statement 4 by [21, Proposition 3.27]. □

4.2 Application to stochastic subgradient method and heavy ball momentum

4.2.1 A general nonsmooth stochastic setting

Let $(S, \mathcal{A}, P)$ be a probability space. We consider a minimization problem

$$\min_{w \in \mathbb{R}^p} F(w) := \mathbb{E}_{\xi \sim P}[f(w, \xi)]$$

where f is jointly measurable and integrable with respect to the second argument for all $w \in \mathbb{R}^p$. We assume $F^* := \inf_{w \in \mathbb{R}^p} F(w) > -\infty$.

We assume definability with respect to the first variable, and access to a conservative gradient with respect to the parameter w:

Assumption 2 (Stochastic conservative gradient)**.**

1. *For almost all $s \in S$, $f(\cdot, s)$ is locally Lipschitz, and there exists $D : \mathbb{R}^p \times S \rightrightarrows \mathbb{R}^p$ measurable, such that $D(\cdot, s)$ is a convex-valued conservative gradient for $f(\cdot, s)$,*
2. *$f(\cdot, s)$ and $D(\cdot, s)$ are globally subanalytic for almost all $s \in S$,*
3. *There exists a square integrable function $\kappa : S \to \mathbb{R}_+$ and $\psi : \mathbb{R}_+ \to \mathbb{R}_+$ nondecreasing and locally bounded such that for almost all $s \in S$, $\|D(w, s)\| \leq \kappa(s)\psi(\|w\|)$ for all $w \in \mathbb{R}^p$.*

Although D can be taken as the Clarke subgradient of f with respect to w, $\partial^c_w f$, this assumption is more general and allows us to capture the nonsmooth calculus presented in Chapter 3 thus making our results compatible with practical implementations of first-order methods. For instance, for almost all s, $D(\cdot, s)$ can be the product of several Clarke Jacobians in order to contain the output of backpropagation applied to $f(\cdot, s)$. This setting is also compatible with first-order oracles employing nonsmooth implicit differentiation, which we proved in Chapter 3 to output conservative Jacobians, as in hyperparameter optimization and the training of implicit neural networks.

Assumption 2.3 allows to capture polynomial growth which appears for instance in deep neural networks. Former works [21, 29] assumed linear growth which is insufficient in machine learning settings.

We define the expected conservative gradient D_F and the critical set of F relative to D_F.

$$D_F := \mathbb{E}_{\xi \sim P}[D(\cdot, \xi)] \quad \text{and} \quad \operatorname{crit} D_F := \{w \in \mathbb{R}^p \ : \ 0 \in D_F(w)\}$$

where the expectation is taken in the sense of Aumann, Definition 2.5. Note that under Assumption 2.3, this set-valued map is well defined.

Assumption 3 (Globally subanalytic distribution)**.** *f and D are jointly globally subanalytic. P has a globally subanalytic density with respect to Lebesgue.*

The main reason for this assumption is to ensure a strong form of Sard's theorem for the risk function F by applying integration results for definable functions presented in Chapter 2. Globally subanalytic densities are extremely flexible: they approximate all continuous densities on compact sets. Although Assumption 3 relates to the unknown distribution P, this is a reasonable proxy for a large class of distributions. A few examples of globally subanalytic densities can be found in Section 2.2.3.

Lemma 4.9 (Path differentiability of risk function)**.** *Under Assumption 2, F admits a chain rule with respect to $\partial^c F$, i.e., for any absolutely continuous curve $\gamma : [0,1] \to \mathbb{R}^p$,*

$$\frac{\mathrm{d}}{\mathrm{d}t}(F \circ \gamma)(t) = \langle a, \dot{\gamma}(t) \rangle \text{ for all } a \in \partial^c F(\gamma(t)) \text{ for almost all } t \in [0,1].$$

F also admits a chain rule with respect to D_F which is graph-closed and locally bounded.

Proof: This is a consequence of the general Theorem 3.13 which holds under Assumption 2. Note in particular that square integrability of κ implies integrability since P is a probability measure. □

4.2.2 Stochastic subgradient method

We consider v a measurable selection of D and a nonsmooth stochastic subgradient method:

$$\begin{aligned} &w_0 \in \mathbb{R}^p, \\ &w_{k+1} = w_k - \alpha_k v(w_k, \xi_k) \quad \text{for } k \in \mathbb{N}, \end{aligned} \tag{4.2}$$

where the $(\xi_k)_{k\in\mathbb{N}}$ are i.i.d. and follow the distribution P. We assume the following:

Assumption 4.

1. $\sup_{k\in\mathbb{N}} \|w_k\| < \infty$ *almost surely,*
2. *For all $k \in \mathbb{N}$, $\alpha_k > 0$, $\sum_{k=0}^{\infty} \alpha_k = \infty$, $\alpha_k \to 0$.*

Some solutions have been proposed in the literature in order to ensure sequence boundedness. For instance, [72] uses a projection on a compact set. The authors of [43, 123] use a restart mechanism where the sequence is reset, either at the initialization in [123] or on a sphere containing a level set in [43]. Such mechanisms are however hard to apply in our case since they require computing $F(w_k)$ at each iteration. The vanishing stepsize assumption is common in stochastic optimization [19, 21, 43, 134] but can raise the question as to the behavior of the constant stepsize regime. This case has been studied in [28] where the authors provide weak convergence results.

Our convergence results for this method state as follows:

Theorem 4.10 (Convergence of the stochastic subgradient method)**.** *Let $(w_k)_{k\in\mathbb{N}}$ be defined by (4.2). Under Assumption 2, Assumption 4 and if $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$, the following hold almost surely:*

1. *The accumulation points of $(w_k)_{k\in\mathbb{N}}$ is a connected set,*
2. *Any accumulation point w^* of $(w_k)_{k\in\mathbb{N}}$ such that $F(w^*) = \liminf_{k\to\infty} F(w_k)$ satisfies $0 \in D_F(w^*)$,*
3. *Any essential accumulation point w^* of $(w_k)_{k\in\mathbb{N}}$ satisfies $0 \in D_F(w^*)$,*
4. *Under Assumption 3, any accumulation point w^* satisfies $0 \in D_F(w^*)$ and $F(w_k)$ converges as $k \to \infty$.*

For compactly supported distributions, we may have a slower stepsize decrease.

Theorem 4.11 (Convergence under slow stepsizes)**.** *Let Assumption 2 and Assumption 4 hold. Then if $\alpha_k = o(1/\log k)$ and P has compact support, statement 1 to 4 from Theorem 4.10 hold.*

We apply the differential inclusion approach to prove our convergence results. The main points to address are the existence of a limit differential inclusion with a Lyapunov function, and the noise extinction property, Assumption 1.3.

We rely on the conservative gradient flow:

$$\dot{w} \in -D_F(w). \tag{4.3}$$

Existence of solutions and Lyapunov function.

Lemma 4.12 (Existence of a solution and Lyapunov function)**.** *Under Assumption 2, the differential inclusion* (4.10) *admits a global solution from any point and F is a Lyapunov function for* (4.3) *and* $\operatorname{crit} D_F = \{w \in \mathbb{R}^p \ : \ 0 \in D_F(w)\}$.

Proof: *Existence of a solution.* We recall $F^* \in \mathbb{R}$ is the strict lower bound to F. Fix $T > 0$ and $w_0 \in \mathbb{R}^p$. Let $\mathcal{D} = [-T, T] \times \overline{B(w_0, r)}$ with $r := 2T\sqrt{(F(w_0) - F^*)} > 0$, which is closed, bounded domain and contains $(0, w_0)$. Since D_F is a conservative gradient, Lemma 4.9, it is nonempty compact valued, and graph-closed. It is furthermore convex-valued since set-valued integration preserves convex-valuedness. We are in the setting of Theorem 4.1, hence there exists $d > 0$ with $d < T$ such that (4.3) admits a solution on $[0, d]$. Let w with $w(0) = w_0$ be an arbitrary such solution. By Theorem 4.2, w can be continued to the boundary of $\mathcal{D}$, i.e., to $t = T$ or $\|w_0 - w(t)\| = r$. Since D_F is a conservative gradient for F, one has for all $t \in [0, d]$,

$$\begin{aligned}\left(\frac{1}{t}\int_0^t \|\dot{w}(s)\| \,\mathrm{d}s\right)^2 \leq \frac{1}{t}\int_0^t \|\dot{w}(s)\|^2 \,\mathrm{d}s &= -\int_0^t \langle D_F(w(s)), \dot{w}(s)\rangle \,\mathrm{d}s \\ &= F(w_0) - F(w(t))\end{aligned}$$

so that $\|w_0 - w(t)\| \leq \int_0^t \|\dot{w}(s)\| \,\mathrm{d}s \leq t\sqrt{(F(w_0) - F^*)}$. However $t\sqrt{(F(w_0) - F^*)} < r$, hence the solution w can be continued to $t = T$. Since w_0, T and w were arbitrary, the flow of (4.3) is nonempty and well defined on $\mathbb{R}_+$.

Lyapunov function. Let $x \in \mathbb{R}^p$, $t \geq 0$ and $y \in \Phi_t(x)$. By definition of Φ_t, there exists $w : \mathbb{R}_+ \to \mathbb{R}^p$ a solution to the differential inclusion (4.3) with initial value $w(0) = x \in \mathbb{R}^p$ and such that $y = w(t)$. By definition of a conservative gradient and since w is absolutely continuous, we have:

$$F(w(t)) - F(w(0)) = \int_0^t \langle D_F(w(s)), \dot{w}(s)\rangle \,\mathrm{d}s. \tag{4.4}$$

Since w is a solution of (4.3), $\dot{w}(s) \in -D_F(w(s))$ for almost all $s \in [0, t]$ and we have $F(w(t)) - F(w(0)) = -\int_0^T \|\dot{w}(s)\|^2 \,\mathrm{d}s$, hence $F(w(t)) = F(y) \leq F(x)$.

Now we suppose that $x \in \mathbb{R}^p \setminus \operatorname{crit} D_F$ and $t > 0$. By upper semicontinuity of D_F, $\exists \epsilon > 0, \exists \delta > 0, \forall y \in \mathbb{R}^p$ such that $\|y - x\| \leq \delta$, we have $\forall v \in D_F(y), \|v\| \geq \epsilon$. By continuity of w, $\exists t_0 > 0, \forall s \in [0, t_0]\,, \|w(s) - x\| \leq \delta$ hence $\|\dot{w}(s)\| \geq \epsilon$ for almost all $s \in [0, t_0]$. Thus, by integration, $\int_0^T \|\dot{w}(s)\|^2 \,\mathrm{d}s$ is strictly positive and $F(y) < F(x)$. □

Noise extinction.

Lemma 4.13 (Noise extinction)**.** *Let $(w_k)_{k\in\mathbb{N}}$ be defined by* (4.2)*. Under Assumption 2 and Assumption 4 set for all $k \in \mathbb{N}$, $V(w_k) = \mathbb{E}[v(w_k, \xi_k)|w_k]$ and $u_k = v(w_k, \xi_k) - V(w_k)$. Then the following hold:*

1. $\sup_{k\in\mathbb{N}} \mathbb{E}[\|u_k\|^2|w_k]$ *is finite almost surely.*

2. *If* $\sum_{k\in\mathbb{N}} \alpha_k^2 < \infty$, *then* $\sum_{i=0}^k \alpha_i u_i$ *converges almost surely as* $k \to \infty$.

3. *If* $\alpha_k = o(1/\log(k))$ *and* $\phi\colon w \mapsto \sup_{s\in\operatorname{supp} P} \|v(w,s)\|$ *is locally bounded, then*

$$\forall T > 0, \qquad \lim_{k\to\infty} \sup_{m\geq k} \left\{ \left\| \sum_{i=k}^m \alpha_i u_i \right\|, \quad \text{s.t.} \sum_{i=k}^m \alpha_i \leq T \right\} = 0 \qquad a.s. \tag{4.5}$$

Proof: Let κ and ψ be given by Assumption 2.3. For the first statement, we have for all $k \in \mathbb{N}$,

$$\begin{aligned} \mathbb{E}\left[\|u_k\|^2 \mid w_k\right] &= \mathbb{E}_{\xi\sim P}\left[\|v(w_k,\xi) - V(w_k)\|^2\right] \leq \mathbb{E}_{\xi\sim P}\left[(\|v(w_k,\xi)\| + \|a_k\|)^2\right] \\ &\leq 4\mathbb{E}_{\xi\sim P}\left[\|D(w_k,\xi)\|^2\right] \leq 4\mathbb{E}_{\xi\sim P}\left[\kappa(\xi)^2\right]\psi(\|w_k\|), \end{aligned} \tag{4.6}$$

so that $\sup_{k\in\mathbb{N}} \mathbb{E}[\|u_k\|^2|w_k] \leq 4\mathbb{E}_{\xi\sim P}\left[\kappa(\xi)^2\right]\psi(R) < \infty$ where $R := \sup_{k\in\mathbb{N}} \|w_k\|$ is almost surely finite by Assumption 4.1. This proves the first statement.

Toward a proof of the second statement, let $\epsilon_k := \alpha_k u_k$ and $Z_k := \sum_{i=0}^k \epsilon_i$ for $k \in \mathbb{N}$. We want to prove that Z_k converges almost surely as $k \to \infty$. Fix an arbitrary $M > 0$, and define $\epsilon_k^M := \alpha_k u_k \mathbb{1}_{\{\|w_k\|\leq M\}}$ and $Z_k^M := \sum_{i=0}^k \epsilon_i^M$. $(\epsilon_k^M)_{k\in\mathbb{N}}$ is a martingale difference sequence. Indeed, for $k \in \mathbb{N}$, by independence of ξ_k we have $\mathbb{E}\left[u_k \mid w_k\right] = \mathbb{E}_{\xi\sim P}[v(w_k,\xi)] - V(w_k) = 0$, hence $\mathbb{E}\left[\epsilon_k^M | w_k\right] = 0$ and $(Z_k^M)_{k\in\mathbb{N}}$ is a martingale relatively to the filtration generated by $(\xi_k)_{k\in\mathbb{N}}$. We will apply a martingale convergence theorem [69, Theorem 4.5.2] on $(Z_k^M)_{k\in\mathbb{N}}$. We have to verify $\mathbb{E}[\|Z_k^M\|^2] < \infty$ and $\sum_{k=0}^\infty \mathbb{E}\left[\|\epsilon_k^M\|^2 \;\middle|\; w_k\right] < \infty$. By the inequality (4.6), one has

$$\begin{aligned} \mathbb{E}[\|\epsilon_k^M\|^2|w_k] &= \alpha_k^2 \mathbb{1}_{\{\|w_k\|\leq M\}} \mathbb{E}[\|u_k\|^2|w_k] \\ &\leq 4\alpha_k^2 \mathbb{1}_{\{\|w_k\|\leq M\}} \mathbb{E}_{\xi\sim P}\left[\kappa(\xi)^2\right]\psi(M) \\ &\leq 4\alpha_k^2 \mathbb{E}_{\xi\sim P}\left[\kappa(\xi)^2\right]\psi(M), \end{aligned}$$

and taking the expectation gives $\mathbb{E}[\|\epsilon_k^M\|^2] < \infty$, hence $\mathbb{E}[\|Z_k^M\|^2] < \infty$ for all $k \in \mathbb{N}$. We have furthermore

$$\sum_{k=0}^\infty \mathbb{E}\left[\|\epsilon_k^M\|^2 \;\middle|\; w_k\right] \leq 4\mathbb{E}_{\xi\sim P}\left[\kappa(\xi)^2\right]\psi(M) \sum_{k=0}^\infty \alpha_k^2 < \infty$$

since we assumed $\sum_{k=0}^\infty \alpha_k^2 < \infty$ for the second statement. Finally, [69, Theorem 4.5.2] applies and $\sum_{i=0}^k \epsilon_i^M$ converges almost surely as $k \to \infty$. In particular,

$$P^{\otimes\mathbb{N}}(\{\sum_{i=0}^\infty \epsilon_i^M \text{converges}\} \cap \{\sup_{k\in\mathbb{N}} \|w_k\| \leq M\}) = P^{\otimes\mathbb{N}}(\sup_{k\in\mathbb{N}} \|w_k\| \leq M). \tag{4.7}$$

We can finally prove the almost sure convergence of Z_k.

$$\begin{aligned} P^{\otimes\mathbb{N}}(\{\sum_{i=0}^\infty \epsilon_i \text{ converges}\}) &= P^{\otimes\mathbb{N}}(\{\sum_{i=0}^\infty \epsilon_i \text{ converges}\} \cap \{\sup_{k\in\mathbb{N}} \|w_k\| < \infty\}) \\ &= \lim_{M\to\infty} P^{\otimes\mathbb{N}}(\{\sum_{i=0}^\infty \epsilon_i \text{ converges}\} \cap \{\sup_{k\in\mathbb{N}} \|w_k\| \leq M\}) \\ &= \lim_{M\to\infty} P^{\otimes\mathbb{N}}(\{\sum_{i=0}^\infty \epsilon_i^M \text{converges}\} \cap \{\sup_{k\in\mathbb{N}} \|w_k\| \leq M\}) \\ &= \lim_{M\to\infty} P^{\otimes\mathbb{N}}(\sup_{k\in\mathbb{N}} \|w_k\| \leq M) \\ &= P^{\otimes\mathbb{N}}(\sup_{k\in\mathbb{N}} \|w_k\| < \infty) = 1. \end{aligned}$$

The first and the last equalities follow from $\sup_{k\in\mathbb{N}} \|w_k\| < \infty$ almost surely. The fourth equality follows from (4.7). The third equality follows from the fact that for $i \in \mathbb{N}$, $\epsilon_i^M = \epsilon_i$ whenever $\sup_{k\in\mathbb{N}} \|w_k\| \leq M$. The second and fifth equalities are obtained by monotone convergence since the event $\sup_{k\in\mathbb{N}} \|w_k\| \leq M$ is increasing to $\{\sup_{k\in\mathbb{N}} \|w_k\| < \infty\}$ with respect to M.

For the third statement, let for all $k \in \mathbb{N}$, $c_k = 2\max\{1, \max_{i=0,\dots,k} \phi(w_k)\}$, be almost surely increasing and convergent, such that $\|u_k/c_k\| \leq 1$ almost surely. Note that u_k/c_k are martingale increments: for $k \in \mathbb{N}$, u_k/c_k is integrable and $1/c_k$ is measurable with respect to $w_0, \dots, w_k$ hence $\mathbb{E}[u_k/c_k|w_0, \dots, w_k] = \mathbb{E}[u_k|w_0, \dots, w_k]/c_k = 0$, see [69, Theorem 4.1.14]. Fix $c > 0$, we have $2\log(k) = o(c/\alpha_k)$ so that as $k \to \infty$, $\exp(-c/\alpha_k)\, k^2 = \exp(-c/\alpha_k + 2\log(k)) \to 0$ and $\exp(-c/\alpha_k) = o(1/k^2)$ is summable. We invoke [19, Proposition 4.4]:

$$\forall T > 0 \qquad \lim_{k\to\infty} \sup_m \left\{ \left\| \sum_{i=k}^m \alpha_i u_i/c_i \right\|, \quad \text{s.t.} \sum_{i=k}^m \alpha_i \leq T \right\} = 0 \qquad a.s.$$

Note that [19, Proposition 4.4] can be applied to any subgaussian martingale difference sequence, here we apply it to $(u_k/c_k)_{k\in\mathbb{N}}$ uniformly bounded by 1, hence subgaussian. Fix $T > 0$ and set for all k, m_k the largest integer $m \geq k$ such that $\sum_{i=k}^m \alpha_i \leq T$. We now have for all $k \leq m \leq m_k$

$$\begin{aligned} &\left\| \sum_{i=k}^m \frac{\alpha_i u_i}{c_i} - \frac{1}{c_k} \sum_{i=k}^m \alpha_i u_i \right\| = \left\| \sum_{i=k}^m \left(\frac{1}{c_i} - \frac{1}{c_k} \right) \alpha_i u_i \right\| \\ \leq \quad &\left(\frac{1}{c_k} - \frac{1}{c_{m_k}} \right) \sum_{i=k}^m \alpha_i c_i \leq \left(\frac{1}{c_k} - \frac{1}{c_{m_k}} \right) c_{m_k} \sum_{i=k}^{m_k} \alpha_i \leq \left(\frac{1}{c_k} - \frac{1}{c_{m_k}} \right) c_{m_k} T, \end{aligned}$$

and the result follows because c_k converges almost surely. □

We can now prove our convergence theorems.

Proof of Theorem 4.10: This is an application of the general Theorem 4.8. The recursion (4.2) reads

$$w_{k+1} = w_k - \alpha_k(V(w_k) + u_k)$$

where $V(w_k) = \mathbb{E}_{\xi\sim P}[v(w_k, \xi)] \in D_F(w_k)$ and $u_k = v(w_k, \xi_k) - V(w_k)$.

F is a Lyapunov function for the differential inclusion (4.3) and $\operatorname{crit} D_F$ by Lemma 4.12. Furthermore, Assumption 4.1 corresponds to Assumption 1.1, Assumption 4.2 to Assumption 1.2 and Lemma 4.13.(1-2) hold and imply Assumption 1.3. As to the Sard property of statement 4 from Theorem 4.8, this holds under Assumption 3: by application of integration of definable (set-valued) functions, see Theorem 2.17 and Theorem 2.18, F and D_F are definable in $\mathbb{R}_{\text{an,exp}}$. In this case, by Sard's theorem for definable conservative gradients, Theorem 3.10, $F(\operatorname{crit} D_F)$ has an empty interior. □

Proof of Theorem 4.11: Since D is globally subanalytic, it is polynomially bounded hence

$$\|D(w, s)\| \leq K(1 + \|w\|^{p_0})(1 + \|s\|^{q_0})$$

for some positive constants K, p_0, q_0. In particular, since $\operatorname{supp} P$ is compact, $\sup_{s\in\operatorname{supp} P} \|D(\cdot, s)\|$ is locally bounded and Lemma 4.13.3 holds, leading to Assumption 1.3. The result then follows by application of Theorem 4.8. □

4.2.3 Stochastic heavy ball

In this part, we study a nonsmooth stochastic heavy ball method,

$$\begin{aligned} &w_0, w_1 \in \mathbb{R}^p, \\ &w_{k+1} = w_k - \mu_k v(w_k, \xi_k) + \nu_k(w_k - w_{k-1}). \quad \text{for } k \in \mathbb{N}^*, \end{aligned} \tag{4.8}$$

where $(\mu_k)_{k\in\mathbb{N}}$, $(\nu_k)_{k\in\mathbb{N}}$ are positive parameters to be specified, and the $(\xi_k)_{k\in\mathbb{N}}$ are i.i.d. and follow the distribution P. One easily verifies that (4.8) is equivalent to the first-order recursion

$$\begin{aligned} w_{k+1} =& w_k - \alpha_k y_k \\ y_{k+1} =& \beta_k v(w_{k+1}, \xi_{k+1}) + (1-\beta_k) y_k. \end{aligned} \tag{4.9}$$

with $\mu_k = \alpha_k\beta_{k-1}$ $\nu_k = \alpha_k(1-\beta_{k-1})/\alpha_{k-1}$, and $y_0 = \frac{w_0 - w_1}{\alpha_0}$.

We assume the following:

Assumption 5.

1. $\sup_{k\in\mathbb{N}} \|w_k\| < \infty$ *almost surely,*
2. *For all* $k \in \mathbb{N}$, $\alpha_k > 0$, $\sum_{k=0}^{\infty} \alpha_k = \infty$ *and* $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$,
3. *There exists* $r > 0$ *such that* $\alpha_k/\beta_k \to r$, *and* $\beta_k \in (0,1)$ *for all* $k \in \mathbb{N}$.

For simplicity, we only consider square summable stepsizes here, while indeed slow stepsizes could be considered under a compact distribution assumption as in Theorem 4.11. The Assumption 5.3 is also called exponential memory in the literature, see for instance [77]. Other parametrizations of the memory coefficient can however be considered like a polynomial one, see [77, 124]. For simplicity, we restrict our analysis to this case.

Our convergence results state as follows:

Theorem 4.14 (Convergence of nonsmooth stochastic heavy ball)**.** *Let* $(w_k, y_k)_{k\in\mathbb{N}}$ *be defined by* (4.9)*. Under Assumption 2, Assumption 5, the following hold almost surely:*

1. *The accumulation points of* $(w_k, y_k)_{k\in\mathbb{N}}$ *is a connected set,*
2. *Any accumulation point* (w^*, y^*) *of* $(w_k, y_k)_{k\in\mathbb{N}}$ *such that* $E(w^*, y^*) = \liminf_{k\to\infty} E(w_k, y_k)$ *satisfies* $0 \in D_F(w^*)$ *and* $y^* = 0$,
3. *Any essential accumulation point* (w^*, y^*) *of* $(w_k, y_k)_{k\in\mathbb{N}}$ *satisfies* $0 \in D_F(w^*)$ *and* $y^* = 0$,
4. *Under Assumption 3, any accumulation point* w^* *of* $(w_k)_{k\in\mathbb{N}}$ *satisfies* $0 \in D_F(w^*)$, $y_k \to 0$ *as* $k \to \infty$, *and* $F(w_k)$ *converges as* $k \to \infty$.

To show this result, as in the previous part, we consider the expectation $D_F = \mathbb{E}_{\xi\sim P}[D(\cdot, \xi)]$ and rely on the differential inclusion method. We consider the differential inclusion

$$\begin{aligned} \dot{w} &= -ry \\ \dot{y} &\in D_F(w) - y, \end{aligned} \tag{4.10}$$

and the function $E(w, y) := F(w) + \frac{r}{2}\|y\|^2$. E is bounded below by $F^* = \inf_{w\in\mathbb{R}^p} F(w)$.

Existence of a solution and Lyapunov function.

Lemma 4.15 (Existence of a solution and Lyapunov function)**.** *Under Assumption 2, the differential inclusion* (4.10) *admits solutions and E is a Lyapunov function for* (4.10) *and* $S := \operatorname{crit} D_F \times \{0\}$.

Proof: *Existence of a solution.* Since D_F is a conservative gradient, see Lemma 4.9, the right-hand side in (4.10) is graph-closed, nonempty, compact, and convex-valued hence (4.10) admits a local absolutely continuous solution (w, y) on $[0, T]$ for some $T > 0$. In order to show it can be extended to a solution on $\mathbb{R}_+$, it is sufficient to show explosion in finite time of the lengths of the curves $\int_0^T \|\dot{w}(t)\| \,\mathrm{d}t$ and $\int_0^T \|\dot{y}(t)\| \,\mathrm{d}t$ cannot happen, see Theorem 4.2. By the chain rule property of conservative gradients, we have for almost all $t \in [0, T]$,

$$\frac{\mathrm{d}(E \circ (w, y))}{\mathrm{d}t}(t) = \langle D_F(w(t)), \dot{w}(t)\rangle + r\langle y(t), \dot{y}(t)\rangle.$$

Integrating from 0 to T gives

$$\begin{aligned} E(w(T), y(T)) - E(w(0), y(0)) &= \int_0^T \frac{\mathrm{d}(E \circ (w, y))}{\mathrm{d}t}(t) \,\mathrm{d}t \\ &= \int_0^T \langle D_F(w(t)), \dot{w}(t)\rangle + r\langle y(t), \dot{y}(t)\rangle \,\mathrm{d}t \\ &= \int_0^T \langle \dot{y}(t) + y(t), -ry(t)\rangle + r\langle y(t), \dot{y}(t)\rangle \,\mathrm{d}t \\ &= -\int_0^T r\|y(t)\|^2 \,\mathrm{d}t. \end{aligned} \tag{4.11}$$

Hence for the component w, we have

$$\begin{aligned} \int_0^T \|\dot{w}(t)\| \,\mathrm{d}t = \int_0^T r\|y(t)\| \,\mathrm{d}t &\le \int_0^T r(1 + \|y(t)\|^2) \,\mathrm{d}t \\ &= rT + E(w(0), y(0)) - E(w(T), y(T)) \\ &\le rT + E(w(0), y(0)) - F^*. \end{aligned} \tag{4.12}$$

As to the component y, we have

$$\begin{aligned} \int_0^T \|\dot{y}(t)\| \,\mathrm{d}t &\le \int_0^T (\|D_F(w(t))\| + \|y(t)\|) \,\mathrm{d}t \\ &= \int_0^T (\|D_F(w(t))\| + \frac{1}{r}\|\dot{w}(t)\|) \,\mathrm{d}t \\ &\le \int_0^T \|D_F(w(t))\| \,\mathrm{d}t + T + \frac{1}{r}(E(w(0), y(0)) - F^*). \end{aligned}$$

Let κ and ψ be given by Assumption 2.3, then

$$\|D_F(w(t))\| \le \mathbb{E}_{\xi\sim}[\kappa(\xi)]\psi(\|w(t)\|) \le \mathbb{E}_{\xi\sim}[\kappa(\xi)] \sup_{t\in[0,T]} \psi(\|w(t)\|).$$

Finally,

$$\int_0^T \|\dot{y}(t)\| \, \mathrm{d}t \leq T\mathbb{E}_{\xi\sim}[\kappa(\xi)] \sup_{t\in[0,T]} \psi(\|w(t)\|) + T + \frac{1}{r}(E(w(0),y(0)) - F^*).$$

Since ψ is locally bounded and by (4.12), $\|w(t)\| \leq \|w_0\| + rt + E(w(0),y(0)) - F^*$ for $t \in [0,T]$, this holds for any horizon $T > 0$. The local solution (w,y) can be extended to a global solution on $\mathbb{R}_+$.

Lyapunov function. We now verify that E is a Lyapunov function. Let (w,y) be an absolutely continuous solution of (4.10) with $(w(0),y(0)) \in S$. In this case, the equation (4.11), holds for $T > 0$, hence $E(w(T),y(T)) - E(w(0),y(0)) \leq 0$.

Now suppose $(w(0),y(0)) \notin S$. In the case where $y(0) \neq 0$, then we have $E(w(t),y(t)) - E(w(0),y(0)) < 0$ from (4.11), by continuity of y. If $0 \notin D_F(w(0))$, suppose toward a contradiction that there exists $t > 0$ which is such that $E(w(t),y(t)) = E(w(0),y(0))$. It means by (4.11) that $y(s) = 0$ for almost all $s \in [0,t]$, hence $\dot{y}(s) = 0$ and then $0 \in D_F(w(s))$ for almost all $s \in [0,t]$. By graph-closedness of D_F, we would have $0 \in D(w(0))$ which is a contradiction.

Finally, for $(w(0),y(0)) \notin S$, one has for almost all $t > 0$, $E(w(t),y(t)) < E(w(0),y(0))$. □

For the sequence $(w_k,y_k)_{k\in\mathbb{N}}$ defined by (4.9), we denote for all $k \in \mathbb{N}^*$, $V(w_k) = \mathbb{E}_{\xi\sim P}[v(w_k,\xi)]$ and $u_k = v(w_k,\xi_k) - V(w_k)$. With these notations, the second equation in (4.9) writes

$$y_{k+1} = (1-\beta_k)y_k + \beta_k V(w_{k+1}) + \beta_k u_{k+1}. \tag{4.13}$$

Noise extinction.

Lemma 4.16 (Noise extinction)**.** *Under Assumption 5 and Assumption 2,*

1. $\sup_{k\in\mathbb{N}} \mathbb{E}[\|u_{k+1}\|^2|w_{k+1}] < \infty$ *almost surely,*
2. $\sum_{i=0}^{\infty} \beta_i u_{i+1}$ *converges almost surely.*

Proof: Item 1 is a consequence of Assumption 2.3. Item 2 is an application of square-integrable martingale convergence theorem, see for instance the proof of Lemma 4.13. □

The following lemma appears as an assumption in [29]. [142, Lemma 1], shows that this is a consequence of the boundedness of $(w_k)_{k\in\mathbb{N}}$. In our setting, we may recall the proof for clarity.

Lemma 4.17 (Velocity boundedness)**.** *Let $(w_k,y_k)_{k\in\mathbb{N}}$ be defined by* (4.9)*. Under Assumption 5 and Assumption 2, $(y_k)_{k\in\mathbb{N}}$ is bounded almost surely.*

Proof: For all $k \in \mathbb{N}$, we define the quantity $\tilde{y}_k := y_k + \sum_{i=k}^{\infty} \beta_i u_{i+1}$. Adding $\sum_{i=k+1}^{\infty} \beta_i u_{i+1}$ on both sides in (4.13) gives

$$\tilde{y}_{k+1} = (1-\beta_k)\tilde{y}_k + \beta_k(V(w_{k+1}) + \tilde{y}_k - y_k),$$

hence we have $\|\tilde{y}_{k+1}\| \leq \max\{\|\tilde{y}_k\|, \|V(w_{k+1})\| + \|\tilde{y}_k - y_k\|\}$ for all k. By direct recurrence we have $\|\tilde{y}_{k+1}\| \leq \max\{\|\tilde{y}_0\|, \|V(w_{i+1})\| + \|\tilde{y}_i - y_i\|, i = 0,\dots,k\}$. Furthermore, we have almost surely

$$\sup_{k\in\mathbb{N}} \|V(w_{k+1})\| < \mathbb{E}_{\xi\sim P}[\kappa(\xi)] \sup_{k\in\mathbb{N}} \psi(\|w_{k+1}\|) < \infty$$

by Assumption 5.1 and Assumption 2.3. By Lemma 4.16.2, the quantity $\|\tilde{y}_k - y_k\| = \|\sum_{i=k}^{\infty} \beta_i u_{i+1}\|$ goes to 0 as $k \to \infty$. Finally, we have

$$\limsup_{k\to\infty} \|y_k\| \leq \max\{\|\tilde{y}_0\|, \sup_{k\in\mathbb{N}} \|V(w_{k+1})\| + \|\tilde{y}_k - y_k\|\} < \infty.$$

□

We are now ready to prove Theorem 4.14.

Proof of Theorem 4.14:

Let $(w_k, y_k)_{k\in\mathbb{N}}$ be defined by (4.9). Our goal is to apply Theorem 4.8.

Under Assumption 2 and Assumption 5, Assumption 1.3 holds by Lemma 4.16, and Assumption 1.1 holds under Assumption 5.1 and by Lemma 4.17.

We then verify that the sequence writes as a recursion of the form

$$\begin{aligned} w_{k+1} &= w_k - \alpha_k y_k \\ y_{k+1} &\in y_k + \frac{\alpha_k}{r}(D_F^{\delta_k}(w_k) - y_k + u_{k+1}), \end{aligned}$$

where for $\delta > 0$ and $w \in \mathbb{R}^p$, we recall the fattened map,

$$\begin{aligned} D_F^\delta(w) := \big\{ z \in \mathbb{R}^p : \exists (w', z') \in \mathbb{R}^p \times \mathbb{R}^p, \\ z' \in D_F(w'), \|w' - w\| \leq \delta, \|z' - z\| \leq \delta \big\}, \end{aligned}$$

and almost surely, $\delta_k \to 0$ as $k \to \infty$.

For simplicity, we may assume $r\beta_k = \alpha_k$. We have for all $D_F(w_{k+1}) = D_F(w_k - \alpha_k y_k) \subset D_F^{\alpha_k\|y_k\|}(w_k)$, hence

$$\begin{aligned} w_{k+1} &= w_k - \alpha_k y_k \\ y_{k+1} &\in y_k + \frac{\alpha_k}{r}(D_F^{\alpha_k\|y_k\|}(w_k) + u_{k+1} - y_k) \end{aligned}$$

where $u_{k+1} = v(w_{k+1}, \xi_{k+1}) - \mathbb{E}_{\xi\sim P}[v(w_{k+1}, \xi_{k+1})]$. Under Assumption 5.2 and Lemma 4.17, $\alpha_k\|y_k\|$ goes to 0 as $k \to \infty$.

We can apply Theorem 4.8 to deduce statements 1 to 3. As to statement 4, under Assumption 3, Sard's theorem (Theorem 3.10) holds by application of integration of globally subanalytic functions, (Theorem 2.17, Theorem 2.18). Furthermore by compactness, for any accumulation point w^* of $(w_k)_{k\in\mathbb{N}}$, one can find y^* such that (w^*, y^*) is an accumulation point of $(w_k, y_k)_{k\in\mathbb{N}}$, hence by application of statement 4 from Theorem 4.8, such a w^* satisfies $0 \in D_F(w^*)$. Similarly, $y_k \to 0$ as $k \to \infty$, and $E(w_k, y_k)$ converges, hence $F(w_k) = E(w_k, y_k) - \frac{r}{2}\|y_k\|^2$ converges.

□

4.3 Avoidance of calculus artifacts

The convergence results obtained in Section 4.2.2 and Section 4.2.3 rely on a conservative criticality

$$0 \in D_F(w^*)$$

which raises some concerns. Indeed, we recall that D_F is the expectation $\mathbb{E}_{\xi\sim P}[D(\cdot, \xi)]$. As D_F is a conservative gradient, it might differ from the Clarke subgradient $\partial^c F$ at some points, which we call *calculus artifacts*.

The presence of these artifacts is natural since they are generated by the nonsmooth calculus used in practice. For instance, the stochastic oracle $D(w, \xi)$ may be built by the composition of Clarke Jacobians, in order to model backpropagation, or it may use nonsmooth implicit differentiation in the case of implicit models and bi-level programming. Other artifacts can also be generated in the value of D_F by the integral rule of conservative gradient (Theorem 3.13) which justifies first-order sampling. The convergence to critical points $0 \in D_F(w^*)$ hence suggests that

the convergence of the algorithms can be impacted by applying calculus rules and in particular by the implementation of the algorithms.

Theorem 3.6 states that D_F is gradient almost everywhere, in particular, the Clarke subgradient is the minimal convex-valued conservative gradient. In order to tighten the validity of the first-order methods as implemented in practice, we seek to retrieve a calculus-free convergence and to show that accumulation points w^* "often" satisfy

$$0 \in \partial^c F(w^*),$$

which can be seen as a minimal conservative criticality. This question was untreated in previous works [72, 142] using generalized semismooth derivatives which share the same concern.

While a noise injection could enforce such a result, see [72], we would like to study algorithms without modifying them. We undertake a similar approach to [28] and [36] where randomization of the initial point $w_0 \in \mathbb{R}^p$ can be sufficient for the whole sequence to avoid a zero or meager [1] set. First-order algorithms can be seen as iterations

$$z_{k+1} = \Phi_{\lambda_k, \xi_k}(z_k) \tag{4.14}$$

where $(z_k)_{k\in\mathbb{N}}$ is a parameter sequence from $\mathbb{R}^q$, $\xi_k \in S$ is the sample value and λ_k denotes the stepsizes. We seek to show that the iteration map is at each step a local diffeomorphism at the iterate z_k, hence preserving randomization with respect to a continuous distribution: for all definable $Z \subset \mathbb{R}^m$ one should have the property

$$\dim Z \leq q-1 \Rightarrow \dim \Phi^{-1}_{\lambda_k,\xi_k}(Z) \leq q-1.$$

Where dim is the Hausdorff dimension. This means that if Z has a low dimension, then the set of points leading to Z has also a low dimension. In particular, regarding the recursion (4.14), for all k will z_{k+1} avoid Z for Lebesgue almost all z_k, hence randomizing the initialization z_0 is sufficient to avoid a low dimensional set.

4.3.1 The case of the stochastic subgradient method

As in Section 4.2.2, we consider the algorithm (4.2) for the minimization of $F := \mathbb{E}_{\xi\sim P}[f(\cdot,\xi)]$:

$$\begin{aligned} &w_0 \in \mathbb{R}^p, \\ &w_{k+1} = w_k - \alpha_k v(w_k, \xi_k) \quad \text{for } k \in \mathbb{N}, \end{aligned}$$

where v is a selection in the stochastic oracle D. We show the following result:

Theorem 4.18 (Clarke criticality for randomized initialization)**.** *Assume f and v are jointly definable and P has a continuous distribution with respect to Lebesgue. Then there exists a full measure and residual [2] set $W \subset \mathbb{R}^p$, and a set $\Gamma \subset \mathbb{R}$ with finite complement, such that if $w_0 \in W$ and $\{\alpha_k\}_{k\in\mathbb{N}} \subset \Gamma$, Theorem 4.10 holds with $\partial^c F$ in place of D_F.*

For a definable set $L \subset \mathbb{R}^p \times \mathbb{R}^m$, we define for all $(w,s) \in \mathbb{R}^p \times \mathbb{R}^m$, $L_w := \{s \in \mathbb{R}^m \ : \ (w,s) \in L\}$ and $L_s := \{w \in \mathbb{R}^p \ : \ (w,s) \in L\}$.

Our proof will use recursively the following lemma:

[1] Countable union of manifolds with dimension at most $p-1$.

[2] which complement is a countable union of manifolds of dimension at most $p-1$

Lemma 4.19. *Let $g\colon \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}^p$ be a definable function. Then there exists a definable dense open set $L \subset \mathbb{R}^p \times \mathbb{R}^m$, a subset $\Gamma \subset \mathbb{R}$ which complement is finite as well as a definable dense set $\Delta \subset \Gamma \times \mathbb{R}^m$, such that g is C^2 on L, for every $\alpha \in \Gamma$, the definable set $\{s \in \mathbb{R}^m \ : \ (\alpha, s) \in \Delta\}$ is dense open in $\mathbb{R}^m$ and for all $(\alpha, s) \in \Delta$, L_s is dense and open. Furthermore, denoting $\Phi_{\alpha,s} = \mathrm{Id} - \alpha \nabla_w g(\cdot, s)$ from L_s dense open to $\mathbb{R}^p$, we have*

$$\forall Z \subset \mathbb{R}^p \text{ definable}, \ \dim Z \leq p-1 \Rightarrow \dim \Phi_{\alpha,s}^{-1}(Z) \leq p-1.$$

Proof: Denote by L a definable dense open set such that g is C^2 on L (such sets exist by stratification, see Definition 2.15). Let $\lambda : L \subset \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}^p$ be a definable representation of the eigenvalues of $\nabla_w^2 g$, where $\nabla_w^2 g$ denotes the partial Hessian of g with respect to the variable w. Refine L so that λ is jointly differentiable in (w, s). L is open and dense by the definability of g. We further set $S_0 \subset \mathbb{R}^m$ the definable dense set obtained from Lemma 2.19 such that for all $s \in S_0$ the set L_s is open dense in $\mathbb{R}^p$.

Let F be the complement of the critical values of the function λ_i, for $i = 1, \ldots, p$ on L. The set of critical values F^c is finite by the definable Sard's theorem [40]. Set $\Gamma := \left\{\alpha \in \mathbb{R} \ : \ \alpha \neq 0, \alpha^{-1} \in F\right\}$. For $i = 1, \ldots, p$, set

$$E_i := \{(\alpha, s) \in \Gamma \times S_0 \ : \ \exists w \in L, \alpha \lambda_i(w, s) = 1, \nabla_w \lambda_i(w, s) = 0\}.$$

This set is definable because it is defined by a first-order formula involving definable functions and L, F, S_0 which are definable sets. Let us fix an arbitrary $\alpha \in \Gamma$, and show that the set $E_{\alpha,i} := \{s \in \mathbb{R}^m \ : \ (\alpha, s) \in E_i\}$ has empty interior. By definable choice, Proposition 2.11, there exists $\tilde{w} : E_{\alpha,i} \to \mathbb{R}^p$ definable, such that $\forall s \in E_{\alpha,i}$, $\alpha \lambda_i(\tilde{w}(s), s) = 1$ and $\nabla_w \lambda_i(\tilde{w}(s), s) = 0$. Assume for the sake of contradiction that there exists a nonempty open subset $U \subset E_{\alpha,i}$. By definability of $\tilde{w}$ and stratification, U can be chosen so that $\tilde{w}$ is continuously differentiable on U. Then denoting $\tilde{\lambda}_i : s \mapsto \lambda_i(\tilde{w}(s), s)$ we have for all $s \in U$, $\nabla \tilde{\lambda}_i(s) = 0$. The chain rule applied on $\tilde{\lambda}_i$ yields

$$\forall s \in U, \ \nabla \tilde{\lambda}_i(s) = \mathrm{Jac}\, \tilde{w}(s)^{\mathsf{T}} \nabla_w \lambda_i(\tilde{w}(s), s) + \nabla_s \lambda_i(\tilde{w}(s), s) = \nabla_s \lambda_i(\tilde{w}(s), s) = 0,$$

hence we have for all $s \in U$, $\nabla \lambda_i(\tilde{w}(s), s) = 0$. In other words, since $\lambda_i(\tilde{w}(s), s) = \alpha^{-1}$ for all $s \in U$, then α^{-1} is a critical value of λ_i which contradicts $\alpha \in \Gamma$. This shows that $E_{\alpha,i}$ has an empty interior for all α in Γ, therefore E_i also has an empty interior.

Set $\Delta = (\bigcup_{i=1}^p E_i)^c$, Δ is the complement of a finite union of definable sets with empty interiors so it is definable and dense. Lemma 2.19 implies that there are only finitely many values α such that $\{s \in \mathbb{R}^m \ : \ (\alpha, s) \in \Delta\}$ is not dense in $\mathbb{R}^m$. Therefore, we may refine further Γ by removing finitely many points, and refine Δ accordingly such that it satisfies the desired projection property: for every $\alpha \in \Gamma$, the set $\{s \in \mathbb{R}^m \ : \ (\alpha, s) \in \Delta\}$ is dense in $\mathbb{R}^m$.

Now, fix $\alpha \in \Gamma$ and s such that $(\alpha, s) \in \Delta$. Consider the set

$$K_{\alpha,s} = \left\{w \in L_s \ : \ \Phi'_{\alpha,s}(w) = I_p - \alpha \nabla_w^2 g(w, s) \quad \text{is not invertible}\right\}$$

where I_p is the identity matrix of size p. Diagonalizing $\nabla_w^2 g(w, s)$, the determinant of $\Phi'_{\alpha,s}(w)$ is $\prod_{i=1}^p (1 - \alpha \lambda_i(w, s))$. It is equal to zero if and only if there exists $i \in \{1, \ldots, p\}$ such that $\alpha \lambda_i(w, s) = 1$ hence $K_{\alpha,s} = \bigcup_{i=1}^p \{w \in L_s \ : \ \alpha \lambda_i(w, s) = 1\}$. Since $\alpha \in \Gamma$ and $(\alpha, s) \in \Delta$, by construction of Δ, α^{-1} is a regular value for the functions $w \mapsto \lambda_i(w, s)$, defined for $w \in L_s$, for all $i = 1, \ldots, p$. So the set $K_{\alpha,s}$ is a union of $p-1$ dimensional submanifolds in L_s and $K_{\alpha,s}^c$ is open and dense set in L_s. Then, let $Z \subset \mathbb{R}^p$ be definable and such that $\dim Z \leq p-1$. Assume for the sake of contradiction that there exists a nonempty open set $V \subset \Phi_{\alpha,s}^{-1}(Z)$. The intersection

$V \cap K^c_{\alpha,s}$ is open and nonempty because $K^c_{\alpha,s}$ is dense and both sets are open. Since $\Phi_{\alpha,s}$ is a local diffeomorphism on $K^c_{\alpha,s}$, the image $\Phi_{\alpha,s}(V \cap K^c_{\alpha,s})$ has a nonempty interior but is included in Z of dimension $p-1$, which is a contradiction. The claim is proved. □

The following claim is a consequence of Lemma 2.19.

Claim 4.20. *Let $g\colon \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}$ be a definable function and $v\colon \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}^p \times \mathbb{R}^m$ be a definable map such that $\nabla_w g = v$ on a definable dense open set $C \subset \mathbb{R}^p \times \mathbb{R}^m$. Then there is a definable set $Z \subset \mathbb{R}^p$, dense, such that for all $w \in Z$, $\nabla_w g(w,s) = v(w,s)$ for all s in a definable dense open set in $\mathbb{R}^m$.*

We can now prove the avoidance result, which is slightly stronger than the avoidance of calculus artifacts:

Theorem 4.21 (Genericity of gradient sequences)**.** *Let $g\colon \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}$ and $v : \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}^p$ be definable functions. Assume there is a definable dense open set $C \subset \mathbb{R}^p \times \mathbb{R}^m$ such that for all $(w,s) \in C$, $\nabla_w g(w,s) = v(w,s)$. Let $R \subset \mathbb{R}^p$ be the definable dense set such that for all $w \in R$,*

- *$v(w,s) = \nabla_w g(w,s)$ for all s in a dense definable set.*
- *$g(\cdot, s)$ is C^2 in a neighborhood of w, for all s in a dense definable set.*

Given an arbitrary sequence $(s_k)_{k\in\mathbb{N}}$ in $\mathbb{R}^m$ and an arbitrary $w_0 \in \mathbb{R}^p$, consider the recursion

$$w_{k+1} = w_k - \alpha_k v(w_k, s_k) \text{ for all } k \in \mathbb{N}. \tag{4.15}$$

Then there is $\Gamma \subset \mathbb{R}$ which complement is finite such that if $\{\alpha_k\}_{k\in\mathbb{N}} \subset \Gamma$, then for each $k \in \mathbb{N}$, there exists a dense definable subset $\Sigma_k \subset \mathbb{R}^p \times (\mathbb{R}^m)^k$ such that if $(w_0, s_0, \dots, s_{k-1}) \in \Sigma_k$, then $w_i \in R$, for all $i = 0, \dots, k$. In particular, there is a full measure residual set, $W \subset \mathbb{R}^p$ such that if $w_0 \in W$ and $(s_0, \dots, s_{k-1})$ belongs to some definable dense set, then $w_k \in R$.

Proof: Let $L \subset \mathbb{R}^p \times \mathbb{R}^m$, $\Gamma \subset \mathbb{R}$, $\Delta \subset \Gamma \times \mathbb{R}^m$ be given by Lemma 4.19. g is C^2 on L which is definable dense and open, and for every $\alpha \in \Gamma$, the definable set $\{s \in \mathbb{R}^m \ : \ (\alpha, s) \in \Delta\}$ is dense open in $\mathbb{R}^m$.

Recall that $L \cap C \subset \mathbb{R}^p \times \mathbb{R}^m$ is definable dense. By Lemma 2.19 there exists a definable dense set $\Sigma_0 \subset \mathbb{R}^p$ such that for all $w \in \Sigma_0$, the set $\{s \in \mathbb{R}^m | (w,s) \in L \cap C\}$ is dense. It satisfies the desired property, that is $\Sigma_0 \subset R$. Indeed for any $w \in \Sigma_0$, the set $\{s \in \mathbb{R}^m | (w,s) \in L \cap C\}$ is dense and for each such s, $(w,s) \in L \cap C$, that is g is C^2 at (w,s) and $v(w,s) = \nabla_w g(w,s)$ so that $w \in R$.

We set for all k, $\Delta_k = \{s \in \mathbb{R}^m \ : \ (\alpha_k, s) \in \Delta\}$. By Lemma 4.19, for any $(\alpha, s) \in \Delta$, $\Phi_{\alpha,s} := \mathrm{Id}_p - \alpha \nabla_w g(\cdot, s)$ from $\{w \in \mathbb{R}^p \ : \ (w,s) \in L\}$, dense and open, to $\mathbb{R}^p$ verifies

$$\forall Z \subset \mathbb{R}^p \text{ definable, } \dim Z \leq p-1 \implies \dim \Phi^{-1}_{\alpha,s}(Z) \leq p-1. \tag{4.16}$$

Remark that $w_{k+1} = \Phi_{\alpha_k, s_k} \circ \ldots \circ \Phi_{\alpha_0, s_0}(w_0)$, as long as $s_i \in \Delta_i$ for $i = 0, \dots, k$.

Let us proceed by induction, fix $k \in \mathbb{N}$ and assume that we have $\Sigma_k \subset R \times \Delta_0 \times \ldots \times \Delta_{k-1}$, definable dense, such that for all $(w_0, s_0, \dots, s_{k-1}) \in \Sigma_k$, $w_i \in R$ for all $i = 0, \dots, k$. Note that for $k = 0$, Σ_0 constructed above satisfies the desired hypothesis with the convention that the product set from 0 to -1 is empty.

Let us construct Σ_{k+1}. Remark that $s_i \in \Delta_i$ for $i = 0, \dots, k$ so that $w_{k+1} = \Phi_{\alpha_k, s_k} \circ \ldots \circ \Phi_{\alpha_0, s_0}(w_0)$, as long as $(w_0, s_0, \dots, s_k) \in \Sigma_k \times \Delta_k$.

Consider the set-valued map $N_k\colon s_k \rightrightarrows \Phi^{-1}_{\alpha_k, s_k}(R)$ and by backward recursion for $i = k-1, \dots, 0$, $N_i\colon (s_i, \dots, s_k) \rightrightarrows \Phi^{-1}_{\alpha_i, s_i}(N_{i+1}(s_{i+1}, \dots, s_k))$. Set

$$\Sigma_{k+1} = \{(w, s_0, \dots, s_k) \in \Sigma_k \times \Delta_k | w \in N_0(s_0, \dots, s_k)\}.$$

Let us verify that Σ_{k+1} satisfies the desired properties. We have $\Sigma_{k+1} \subset \Sigma_k \times \Delta_k$ so that for any $(w_0, s_0, \dots, s_k) \in \Sigma_{k+1}$, $(w_0, s_0, \dots, s_{k-1}) \in \Sigma_k$ and $w_i \in R$ for $i = 0, \dots, k$ by induction hypothesis. Furthermore, $s_i \in \Delta_i$ for all $i = 0, \dots, k$ so that $w_{k+1} = \Phi_{\alpha_k, s_k} \circ \dots \circ \Phi_{\alpha_0, s_0}(w_0)$. Note that by construction, $w_0 \in N_0(s_0, \dots, s_k)$ if and only if $\Phi_{\alpha_k, s_k} \circ \dots \circ \Phi_{\alpha_0, s_0}(w_0) \in R$ which is the desired property. It remains to show that Σ_{k+1} is dense, and the induction will be complete. Note that $\Sigma_{k+1} = \Sigma_k \times \Delta_k \cap \{(w, s_0, \dots, s_k) | ((s_0, \dots, s_k), w) \in \mathrm{Graph}(N_0)\}$. Since $\Sigma_k \times \Delta_k$ is definable dense and $\mathrm{Graph}(N_0)$ is definable, it suffices to check that $\mathrm{Graph}(N_0)$ is dense. This is done by backward induction.

Let us first check that $\mathrm{Graph}(N_k)$ is dense. We have $(s_k, w_k) \notin \mathrm{Graph}(N_k)$ if and only if $(w_k, s_k) \notin L$ (Φ_{α_k, s_k} is not defined at w_k), or $w_k \in \Phi^{-1}_{\alpha_k, s_k}(R^c)$ so that $\mathrm{Graph}(N_k)^c = L^c \cup \{(s_k, w_k) | (w_k, s_k) \in L,\ w_k \in \Phi^{-1}_{\alpha_k, s_k}(R^c)\}$. Recall that R^c is definable and is the complement of a dense set, therefore it has at most dimension $p-1$ so that if $s_k \in \Delta_k$, $\Phi^{-1}_{\alpha_k, s_k}(R^c)$ also has dimension at most $p-1$. On the other hand, the set $\{w_k \in \mathbb{R}^p | (w_k, s_k) \in L^c\}$ is the complement of L_{s_k} (with the notation of Lemma 4.19) definable and dense for $s_k \in \Delta_k$. Therefore for all $s_k \in \Delta_k$, the set $\{w_k \in \mathbb{R}^p | (s_k, w_k) \notin \mathrm{Graph}(N_k)\}$ has dimension at most $p-1$ and the set $\{w_k \in \mathbb{R}^p | (s_k, w_k) \in \mathrm{Graph}(N_k)\}$ is dense so that $\mathrm{Graph}(N_k)$ is dense by Lemma 2.19.

This extends by backward induction. Assume that $\mathrm{Graph}(N_{i+1})$ is dense for some $i \in \{0, \dots, k-1\}$. We have $(s_i, \dots, s_k, w_i) \notin \mathrm{Graph}(N_i)$ if and only if $(w_i, s_i) \notin L$ (Φ_{α_i, s_i} is not defined at w_i) or $w_i \in \Phi^{-1}_{\alpha_i, s_i}(N_{i+1}(s_{i+1}, \dots, s_k)^c)$. As N_{i+1} has a dense graph, then by Lemma 2.19, for all $(s_{i+1}, \dots, s_k)$ in a dense definable set R_i, $N_{i+1}(s_{i+1}, \dots, s_k)$ is dense and $N_{i+1}(s_{i+1}, \dots, s_k)^c$ has dimension at most $p-1$. Therefore, similarly as above, for all $s_i \in \Delta_i$ and $(s_{i+1}, \dots, s_k) \in R_i$, the set $\{w_i | (s_i, \dots, s_k, w_i) \in \mathrm{Graph}(N_i)\}$ is dense and $\mathrm{Graph}(N_i)$ is dense. By induction, $\mathrm{Graph}(N_0)$ is dense and this shows that Σ_{k+1} has the correct property.

This proves the first statement. Now by Lemma 2.19, for each $k \in \mathbb{N}$, there is $W_k \subset \mathbb{R}^p$ definable dense such that for each $w_0 \in W_k$, for all $(s_0, \dots, s_{k-1})$ in a dense definable set, $w_i \in R$ for all $i = 0, \dots, k$. We set $W = \cap_{k \in \mathbb{N}} W_k$, W is a residual set by the countable intersection of residual sets (with dense interior), and it has full measure as a countable intersection of full measure sets. □

Remark 4.22. *With the notation of Theorem 4.21, if $(s_i)_{i \in \mathbb{N}}$ are independent and identically distributed with a density with respect to Lebesgue measure, $w_0 \in W$ and $\alpha_k \in \Gamma$ for all $k \in \mathbb{N}$, then almost surely, $w_k \in R$ for all $k \in \mathbb{N}$.*

We can now apply it to obtain our convergence to Clarke critical points, Theorem 4.18.

Proof of Theorem 4.18: We may apply Theorem 4.21 with $g = f$ to obtain W of full measure and residual, and Γ having finite complement, such that if $w_0 \in W$ and $\{\alpha_k\}_{k \in \mathbb{N}} \subset \Gamma$, then for all k, $v(w_k, s_k) = \nabla_w f(w_k, s_k)$ for P-almost all s_k. In particular, $\mathbb{E}_{\xi \sim P}[v(w_k, \xi)] = \mathbb{E}_{\xi \sim P}[\nabla_w f(w_k, \xi)]$.

Let κ, ψ be given by Assumption 2.3. Then $\|\nabla_w f(w_k, s_k)\| \leq \kappa(s_k)\psi(\|w_k\|)$, where κ is integrable. It follows that by dominated convergence theorem, integrating with respect to s_k yields $\mathbb{E}_{\xi \sim P}[\nabla_w f(w_k, \xi)] = \nabla \mathbb{E}_{\xi \sim P}[f(w_k, \xi)] = \nabla F(w_k)$.

Finally, $\mathbb{E}_{\xi \sim P}[v(w_k, \xi)] = \nabla F(w_k)$ and the update $w_{k+1} = w_k - \alpha_k v(w_k, \xi_k)$ reads

$$w_{k+1} = w_k - \alpha_k(\nabla F(w_k) + u_k) \in w_k - \alpha_k(\partial^c F(w_k) + u_k),$$

where $u_k = v(w_k, \xi_k) - \nabla F(w_k)$.

Since $\partial^c F$ is a conservative gradient for F by Lemma 4.9, one may repeat the differential inclusion method of Section 4.2.2 with $\partial^c F$ in place of D_F to obtain the desired result, whenever $w_0 \in W$ and $\{\alpha_k\}_{k \in \mathbb{N}} \subset \Gamma$. □

4.3.2 The case of stochastic heavy ball

We now consider the algorithm (4.8) with stepsizes $(\mu_k)_{k\in\mathbb{N}}, (\nu_k)_{k\in\mathbb{N}}$:

$$w_{k+1} = w_k - \mu_k v(w_k, \xi_k) + \nu_k(w_k - w_{k-1}).$$

We show the following:

Theorem 4.23 (Clarke criticality for randomized initializations)**.** *There exists a subset $W \subset \mathbb{R}^p \times \mathbb{R}^p$ of full measure such that if $(w_1, w_0) \in W$, then Theorem 4.14 holds with $\partial^c F$ in place of D_F.*

If P has a density with respect to Lebesgue and f and v are jointly definable, then W^c is a countable union of manifolds of dimension at most $2p-1$.

In the proofs to come in this section, we will use the following notations:

- We denote $\Sigma \subset S$ of full measure given by Assumption 2.(1-3), which is such that for $s \in \Sigma$, $f(\cdot, s)$ and $v(\cdot, s)$ are definable, and $\|D(w, s)\| \leq \kappa(s)\psi(\|w\|)$ for all $w \in \mathbb{R}^p$.
- We denote $R_s \subset \mathbb{R}^p$, given by Assumption 2.(1-2), the definable and dense subset such that for all $w \in R_s$, $\nabla_w f(\cdot, s) = v(\cdot, s)$, and $f(\cdot, s)$ is C^2 on a neighborhood of w. Such a subset is given by the property of a conservative gradient to be gradient almost everywhere, see Theorem 3.6, and the stratification property of definable sets, see Definition 2.15.

In order to prove the results of this part, we write the update rule as follows:

$$\begin{pmatrix} w_{k+1} \\ w_k \end{pmatrix} = \begin{pmatrix} w_k - \mu_k v(w_k, \xi_k) + \nu_k(w_k - w_{k-1}) \\ w_k \end{pmatrix} = \Psi_{\mu_k, \nu_k, \xi_k}(w_k, w_{k-1}),$$

where for $s \in S$ and $(x, y) \in \mathbb{R}^p \times \mathbb{R}^p$, we let $\Psi_{\mu,\nu,s}(x, y) := (x - \mu v(x, s) + \nu(x - y), x)$. With an abuse of notation and for simplicity, we will write for a sequence $(s_k)_{k\in\mathbb{N}}$, for all $k \in \mathbb{N}$, $\Psi_{\mu_k,\nu_k,s_k} = \Psi_{s_k}$. In this case, for all $k \in \mathbb{N}^*$, $(w_{k+1}, w_k) = (\Psi_{s_k} \circ \Psi_{s_{k-1}} \circ \ldots \circ \Psi_{s_1})(w_1, w_0)$.

The following lemma is the counterpart of Lemma 4.19.

Lemma 4.24. *Under Assumption 2, for $s \in \Sigma$ and $\nu \neq 0$, one has for all definable subset $Z \subset \mathbb{R}^p \times \mathbb{R}^p$,*

$$\dim Z \leq 2p - 1 \Rightarrow \dim \Psi^{-1}_{\mu,\nu,s}(Z) \leq 2p - 1.$$

Proof: Let $Z \subset \mathbb{R}^p$ be definable, of dimension at most $2p-1$. Toward a contradiction, suppose $\dim \Psi^{-1}_{\mu,\nu,s}(Z) = 2p$. Since $\Psi^{-1}_{\mu,\nu,s}(Z)$ is definable, then by stratification, there exists an open subset U of $\mathbb{R}^p \times \mathbb{R}^p$ included in $\Psi^{-1}_{\mu,\nu,s}(Z)$. On $R_s \times \mathbb{R}^p$, $\Psi_{\mu,\nu,s}(x, y) = (x - \mu\nabla_w f(x, s) + \nu(x - y), x)$, and its Jacobian is well defined, given by

$$\operatorname{Jac} \Psi_{\mu,\nu,s}(x, y) = \begin{pmatrix} I_p - \mu\nabla^2_w f(x, s) + \nu I_p & -\nu I_p \\ I_p & 0_p \end{pmatrix}.$$

$\operatorname{Jac} \Psi_{\mu,\nu,s}(x, y)$ is clearly invertible whenever $\nu \neq 0$. In particular, $\Psi_{\mu,\nu,s}$ is a local diffeomorphism on $R_s \times \mathbb{R}^p$. Since $R_s \times \mathbb{R}^p$ is dense and open, and U is open, $U \cap (R_s \times \mathbb{R}^p)$ has nonempty interior. This implies $\Psi_{\mu,\nu,s}(U \cap (R_s \times \mathbb{R}^p))$ has nonempty interior in $\mathbb{R}^p \times \mathbb{R}^p$, in particular it has dimension $2p$, but it is included in Z by definition of U, which is a contradiction since $\dim Z < 2p$. □

Proposition 4.25 (Avoidance of nonsmooth set)**.** *Let $(w_k)_{k\in\mathbb{N}}$ be defined by (4.8). Under Assumption 2 and Assumption 5, for all $k \in \mathbb{N}^*$, there exists a subset $W_k \subset \mathbb{R}^p \times \mathbb{R}^p$ of full Lebesgue measure such that if $(w_1, w_0) \in W_k$, then for $(s_i)_{i=1,\dots,k}$ in a set of full measure with respect to P^k, it holds that for $i = 1, \dots, k$, $\nabla_w f(w_i, s_i) = v(w_i, s_i)$, and $\mathbb{E}_{\xi\sim P}[v(w_i, \xi)] = \nabla F(w_i)$.*

In particular, for (w_1, w_0) in a set of full Lebesgue measure $W \subset \mathbb{R}^p \times \mathbb{R}^p$, $\mathbb{E}_{\xi\sim P}[v(w_k, \xi)] = \nabla F(w_k)$ for all $k \in \mathbb{N}^*$, $P^{\otimes\mathbb{N}}$-almost surely in $(\xi_k)_{k\in\mathbb{N}}$.

Proof: We first show that for $k \in \mathbb{N}^*$, for P^k-almost all $(s_i)_{i=1,\dots,k}$, there exists a set $Z_k \subset \mathbb{R}^p \times \mathbb{R}^p$ of full measure such that for all $(w_1, w_0) \in Z_k$, $\nabla_w f(w_i, s_i) = v(w_i, s_i)$ for $i = 0, \dots, k$. We then deduce the desired result by Fubini's theorem.

For each $k \in \mathbb{N}^*$, assume $s_k \in \Sigma$. Fix $k \in \mathbb{N}^*$. For $i = 1, \dots, k$, we let $V_i = (\Psi_{s_i} \circ \Psi_{s_{i-1}} \circ \dots \circ \Psi_{s_1})^{-1}(R_{s_{i+1}} \times R_{s_i})$. By construction, if $(w_1, w_0) \in V_i$, then $(w_{i+1}, w_i) \in R_{s_{i+1}} \times R_{s_i}$. Consequently, by definition of the R_s for $s \in \Sigma$, if $(w_1, w_0) \in Z_k := \bigcap_{i=1}^k V_k$ then for all $i = 1, \dots k$, $\nabla_w f(w_i, s_i) = v(w_i, s_i)$.

It remains to verify for all $k \in \mathbb{N}^*$, V_k is dense definable, or V_k^c has dimension at most $2p - 1$. This is done by induction with Lemma 4.24.

Fix $k \in \mathbb{N}^*$, $R_{s_{k+1}}$ and R_{s_k} are open and dense, hence $\dim(R_{s_{k+1}} \times R_{s_k})^c \leq 2p-1$. Since $s_k \in \Sigma$, then by Lemma 4.24, $\Psi_{s_k}^{-1}((R_{s_{k+1}} \times R_{s_k})^c)$ has dimension at most $2p - 1$. Applying recursively Lemma 4.24 for the function Ψ_{s_i} and the set $(\Psi_{s_k} \circ \dots \circ \Psi_{s_{i+1}})^{-1}((R_{s_{k+1}} \times R_{s_k})^c)$ for $i = k-1$ to 1, proves that $V_k^c = (\Psi_{s_k} \circ \Psi_{s_{k-1}} \circ \dots \circ \Psi_{s_1})^{-1}((R_{s_{k+1}} \times R_{s_k})^c)$ has dimension at most $2p-1$. Finally, the intersection $Z_k = \bigcap_{i=1}^k V_k$ is dense definable.

For any $k \in \mathbb{N}^*$, we proved that for P^k-almost all $(s_i)_{i=1,\dots,k}$ in Σ^k, there exists Z_k of full measure such that if $(w_1, w_0) \in Z_k$, then $\nabla_w f(w_i, s_i) = v(w_i, s_i)$ for $i = 1, \dots, k$. By Fubini's theorem on the product $\Sigma^k \times (\mathbb{R}^p \times \mathbb{R}^p)$, this implies that there exists a set of full Lebesgue measure $W_k \subset \mathbb{R}^p \times \mathbb{R}^p$, such that if $(w_1, w_0) \in W_k$, then for P^k-almost all $(s_i)_{i=1,\dots,k}$ in Σ^k, $\nabla_w f(w_i, s_i) = v(w_i, s_i)$ for $i = 1, \dots, k$. By Assumption 2.3, we have $\|\nabla_w f(w_i, s_i)\| \leq \kappa(s_i)\psi(\|w_i\|)$ for almost all s_i. By dominated convergence theorem, we can interchange integral with respect to s_i and gradient with respect to w_i, to write $\nabla F(w_i) = \mathbb{E}_{\xi\sim P}[\nabla_w f(w_i, \xi)] = \mathbb{E}_{\xi\sim P}[v(w_i, \xi)]$.

Finally, set $W := \bigcap_{k\in\mathbb{N}^*} W_k$. By definition of the W_k, if $(w_1, w_0) \in W$, $\mathbb{E}_{\xi\sim P}[v(w_k, \xi)] = \nabla F(w_k)$ for all $k \in \mathbb{N}^*$. □

Under a further assumption on the distribution P, the set of initializations leading to stochastic gradient sequences is residual

Corollary 4.26. *Assume $S = \mathbb{R}^m$, P has a density with respect to Lebesgue and f and v are jointly definable. Then Proposition 4.25 holds with the additional properties: W_k^c is a finite union of manifolds with dimension at most $2p-1$ and W^c is a countable union of manifolds with dimension at most $2p - 1$.*

Proof: For $k \in \mathbb{N}^*$, consider the set

$$L_k := \{(w_1, w_0, s_1, \dots, s_k) \ : \forall i \in \{1, \dots, k\}, v(w_i, s_i) = \nabla_w f(w_i, s_i)\}.$$

By stability properties of definable sets, L_k is definable. By Proposition 4.25, for Lebesgue almost all (w_1, w_0) and for almost all $(s_1, \dots, s_k)$, $v(w_i, s_i) = \nabla_w f(w_i, s_i)$ for $i = 1, \dots, k$. Applying Lemma 2.19, L_k is dense. Also by Lemma 2.19, there exists a definable set W_k open and dense such that if $(w_1, w_0) \in W_k$, then it holds that $\forall i \in \{1, \dots, k\}, v(w_i, s_i) = \nabla_w f(w_i, s_i)$ for almost all s_i. Under Assumption 2.3, we then may apply the dominated convergence theorem as in the proof of Proposition 4.25 to obtain $\nabla F(w_i) = \mathbb{E}_{\xi\sim P}[v(w_i, \xi)]$.

By stratification, W_k^c is a finite union of manifolds with dimension at most $2p-1$, and the complementary of $W = \bigcap_{k\in\mathbb{N}} W_k$ is a countable union of manifolds with dimension at most $2p-1$. □

We can finally prove Theorem 4.23:

Proof of Theorem 4.23: In the setting of Theorem 4.14, Assumption 2 and Assumption 5 hold. We can then apply Proposition 4.25.

Let W be given by Proposition 4.25. Assume $(w_1, w_0) \in W$, then by definition of W, we have $P^{\otimes\mathbb{N}}$-almost surely, for all $k \in \mathbb{N}$,

$$\begin{aligned} w_{k+1} &= w_k - \alpha_k y_k \\ y_{k+1} &= \beta_k(\nabla F(w_{k+1}) + u_{k+1}) + (1-\beta_k) y_k, \end{aligned} \tag{4.17}$$

with $u_{k+1} = v(w_{k+1}, \xi_{k+1}) - \mathbb{E}_{\xi\sim P}[v(w_{k+1}, \xi)]$. In particular, $(y_k)_{k\in\mathbb{N}}$ satisfies the relation (4.13) with $V(w_{k+1}) \in \partial^c F(w_{k+1})$.

Since $\partial^c F$ is a conservative gradient for F, we may follow the differential inclusion method from Section 4.2.3 with $\partial^c F$ in place of D_F to obtain the result.

The last part of the theorem is simply a consequence of Corollary 4.26. □

Conclusion and perspectives

This thesis studied nonsmooth aspects of stochastic machine learning problems with a focus on calculus. Initially introduced in the context of automatic differentiation, the conservative calculus [37] proves to be a versatile way to model several machine learning practices. We proposed two extensions to this framework, an integral rule (Theorem 3.13) and a nonsmooth implicit differentiation formula (Theorem 3.14).

Nonsmooth implicit differentiation was explored in several contexts such as implicit models, differentiable programming, and hyperparameter optimization Section 3.3.3. Some concerning behaviors were showcased in Section 3.3.4 when applying the implicit differentiation formula outside the invertibility condition. This limitation raises the question of designing first-order algorithms in these situations. Some heuristics were recently proposed to ignore this limitation [14] but without proper theoretical support.

The integral rule provides a theoretical basis for algorithms using first-order sampling with automatic differentiation. Furthermore, it justifies path differentiability of a general risk function. This stands in contrast to prior works that relied on rigid assumptions, such as semialgebraicity, to obtain this regularity condition. A question that remains open yet, is that of the Sard condition for general risk functions, a condition which was essential to obtain sharp convergence results such as the criticality of all accumulation points and the convergence of the objective function (Theorem 4.8.4). We provided a partial answer regarding a specific class of absolutely continuous distributions with definable densities (Assumption 3), going beyond finite distributions that were considered in previous works [36, 37, 63]. However, this assumption still relies on the definable setting. Something desirable would be to define an operational class of path differentiable functions, encompassing various risk functions, satisfying the Sard condition but without necessarily being definable.

The chain rule along curves of conservative gradients naturally led to ODE methods in Chapter 4. While our convergence results are readily applicable in a broad context, it is important to note that they remain purely asymptotic. This prompts consideration of the algorithms' complexity. Although definable geometry proved to be leverageable to obtain convergence rates for continuous time dynamics [32, 51], the question as to how the algorithms benefit from this rigidity remains open in the nonsmooth nonconvex setting. Furthermore, the link between continuous and discrete-time dynamics is not well understood, which suggests a gap to fill in the non-asymptotic regime.

The versatility of the ODE approach encourages to explore further algorithmic extensions. Adaptive stepsizes, e.g. AdaGRAD [150] and Adam [91], play a significant role in deep learning. They raise various questions, including the existence of limiting dynamics for trajectory-dependent stepsizes and the stochastic setting. Further investigation into noise treatment could be considered as well, such as variance reduction techniques [66] and non-independent samples [74].

In Chapter 4, Section 4.3, we dealt with the question of the calculus artifacts avoidance in order to certify the validity of first-order methods as implemented in practice. Our proofs were tailored

independently to two algorithms although we suggested that this question could be formulated under a general form by a measure-preserving property. Developing a general theory on this question is indeed a desirable extension in order to tackle a multitude of first-order methods. Moreover, while our guarantees depend on theoretical genericity notions, such as full Lebesgue measure and residual sets, practical considerations were not explored, notably the impact of quantization and floating point representation [24, 103].

This thesis was essentially driven by certifiability [3], aimed at justifying and providing guarantees to prevalent practices observed in the machine learning community. Another research direction could be to develop and enhance algorithms in the nonsmooth and nonconvex setting. Partial smoothness of nonsmooth functions possessing a specific structure has previously been leveraged to develop fast algorithms [16, 64]. In a more general setting, such as in deep learning, the presence of smooth structures in the optimization landscape is characterized by the stratification property (Definition 2.15). This raises the question of how to design algorithms that would efficiently harness this property. To develop scalable algorithms, one may wonder if such an improvement is possible in the stochastic setting since the integrability results of definable functions [56] show that stratification persists under integration, whenever a rigidity assumption holds on the distribution. Given the experimental aspect of deep learning models, more tailored compositional framework and automatic differentiation methods [95, 158] could be of interest in this perspective.

All these questions, explored here through the narrow lens of optimization, require consideration within the broader context of machine learning. The success of recent models and their capacities to assimilate complex representations from large datasets remains mostly unexplained through traditional viewpoints. For instance, despite the prevalent existence of nonsmoothness, its precise impact remains uncertain. In deep learning, it seemingly stems from simplistic representations from earlier times. While modern architectures tend to incorporate smooth operations efficiently [87,145], the ReLU function continues to find widespread usage. And now, we wonder whether it will persist as a reliable component in the long run, or if it is merely a vestige from the early stages of deep learning.

[3] During my Ph.D. thesis, I was actually part of the certifiability cohort of the Toulouse AI Institute, ANITI.

Bibliography

[1] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.

[2] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[3] Akshay Agrawal, Shane Barratt, Stephen Boyd, Enzo Busseti, and Walaa M Moursi. Differentiating through a cone program. *J. Appl. Numer. Optim*, 1(2):107–115, 2019.

[4] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

[5] C.D. Aliprantis and K.C. Border. *Infinite Dimensional Analysis*. Springer Berlin, Heidelberg, 2006.

[6] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pages 136–145. PMLR, 2017.

[7] Cem Anil, James Lucas, and Roger Grosse. Sorting out Lipschitz function approximation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 291–301. PMLR, 09–15 Jun 2019.

[8] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017.

[9] F.M. Arscott and A.F. Filippov. *Differential Equations with Discontinuous Righthand Sides: Control Systems*. Mathematics and its Applications. Springer Netherlands, 1988.

[10] J.P. Aubin and A. Cellina. *Differential Inclusions: Set-Valued Maps and Viability Theory*. Grundlehren der mathematischen Wissenschaften. Springer Berlin, Heidelberg, 2012.

[11] Robert J Aumann. Integrals of set-valued functions. *Journal of Mathematical Analysis and Applications*, 12(1):1–12, 1965.

[12] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 690–701. Curran Associates, Inc., 2019.

[13] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Trellis networks for sequence modeling. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[14] Antoine Bambade, Fabian Schramm, Sarah El Kazdadi, Stéphane Caron, Adrien Taylor, and Justin Carpentier. PROXQP: an Efficient and Versatile Quadratic Programming Solver for Real-Time Robotics Applications and Beyond. working paper or preprint, September 2023.

[15] Anas Barakat and Pascal Bianchi. Convergence and dynamical behavior of the adam algorithm for nonconvex stochastic optimization. *SIAM Journal on Optimization*, 31(1):244–274, 2021.

[16] Gilles Bareilles, Franck Iutzeler, and Jé rôme Malick. Harnessing structure in composite nonsmooth minimization. *SIAM Journal on Optimization*, 33(3):2222–2247, aug 2023.

[17] Walter Baur and Volker Strassen. The complexity of partial derivatives. *Theoretical Computer Science*, 22(3):317–330, 1983.

[18] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[19] Michel Benaïm. Dynamics of stochastic approximation algorithms. In *Seminaire de probabilites XXXIII*, pages 1–68. Springer, 1999.

[20] Michel BenaÏm and Morris W. Hirsch. Asymptotic pseudotrajectories and chain recurrent flows, with applications. *Journal of Dynamics and Differential Equations*, 8(1):141–176, Jan 1996.

[21] Michel Benaïm, Josef Hofbauer, and Sylvain Sorin. Stochastic approximations and differential inclusions. *SIAM Journal on Control and Optimization*, 44(1):328–348, 2005.

[22] Michel Benaïm and Sebastian J. Schreiber. Ergodic properties of weak asymptotic pseudotrajectories for semiflows. *Journal of Dynamics and Differential Equations*, 12(3):579–598, Jul 2000.

[23] Yoshua Bengio. Gradient-based optimization of hyperparameters. *Neural Computation*, 12(8):1889–1900, 2000.

[24] David Bertoin, Jérôme Bolte, Sébastien Gerchinovitz, and Edouard Pauwels. Numerical influence of relu'(0) on backpropagation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 468–479. Curran Associates, Inc., 2021.

[25] Quentin Bertrand, Quentin Klopfenstein, Mathieu Blondel, Samuel Vaiter, Alexandre Gramfort, and Joseph Salmon. Implicit differentiation of lasso-type models for hyperparameter optimization. In *International Conference on Machine Learning*, pages 810–821. PMLR, 2020.

[26] Quentin Bertrand, Quentin Klopfenstein, Mathurin Massias, Mathieu Blondel, Samuel Vaiter, Alexandre Gramfort, and Joseph Salmon. Implicit differentiation for fast hyperparameter selection in non-smooth convex learning. *arXiv preprint arXiv:2105.01637*, 2021.

[27] Pascal Bianchi, Walid Hachem, and Adil Salim. Constant step stochastic approximations involving differential inclusions: stability, long-run convergence and applications. *Stochastics*, 91(2):288–320, 2019.

[28] Pascal Bianchi, Walid Hachem, and Sholom Schechtman. Convergence of constant step stochastic gradient descent for non-smooth non-convex functions. *Set-Valued and Variational Analysis*, 30(3):1117–1147, Sep 2022.

[29] Pascal Bianchi and Rodolfo Rios-Zertuche. A closed-measure approach to stochastic approximation. *arXiv preprint arXiv:2112.05482*, 2021.

[30] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1613–1622. JMLR.org, 2015.

[31] Jerome Bolte, Ryan Boustany, Edouard Pauwels, and Béatrice Pesquet-Popescu. On the complexity of nonsmooth automatic differentiation. In *The Eleventh International Conference on Learning Representations*, 2023.

[32] Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007.

[33] Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. Tame functions are semismooth. *Mathematical Programming*, 117(1):5–19, Mar 2009.

[34] Jérôme Bolte, Tam Le, and Edouard Pauwels. Subgradient sampling for nonsmooth nonconvex minimization. *SIAM Journal on Optimization*, 33(4):2542–2569, 2023.

[35] Jérôme Bolte, Tam Le, Edouard Pauwels, and Tony Silveti-Falls. Nonsmooth implicit differentiation for machine-learning and optimization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 13537–13549. Curran Associates, Inc., 2021.

[36] Jérôme Bolte and Edouard Pauwels. A mathematical model for automatic differentiation in machine learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 10809–10819. Curran Associates, Inc., 2020.

[37] Jérôme Bolte and Edouard Pauwels. Conservative set valued fields, automatic differentiation, stochastic gradient methods and deep learning. *Mathematical Programming*, 188(1):19–51, Jul 2021.

[38] Jerome Bolte, Edouard Pauwels, and Samuel Vaiter. Automatic differentiation of nonsmooth iterative algorithms. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 26404–26417. Curran Associates, Inc., 2022.

[39] Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. A nonsmooth morse–sard theorem for subanalytic functions. *Journal of Mathematical Analysis and Applications*, 321(2):729–740, 2006.

[40] Jérôme Bolte, Aris Daniilidis, Adrian Lewis, and Masahiro Shiota. Clarke subgradients of stratifiable functions. *SIAM Journal on Optimization*, 18:556–572, 01 2007.

[41] Jérôme Bolte, Edouard Pauwels, and Rodolfo Ríos-Zertuche. Long term dynamics of the subgradient method for lipschitz path differentiable functions. *Journal of the European Mathematical Society*, 12 2022.

[42] Jérôme Bolte, Edouard Pauwels, and Antonio José Silveti-Falls. Differentiating nonsmooth solutions to parametric monotone inclusion problems, 2022.

[43] V.S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint;second Edition.* Texts and Readings in Mathematics Series. Hindustan Book Agency, 2022.

[44] Jonathan M. Borwein and Warren B. Moors. Essentially smooth lipschitz functions: Compositions and chain rules. In Felipe Cucker and Michael Shub, editors, *Foundations of Computational Mathematics*, pages 16–22, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[45] Jonathan M. Borwein and Xianfu Wang. Lipschitz functions with maximal clarke subdifferentials are generic. *Proceedings of the American Mathematical Society*, 128(11):3221–3229, 2000.

[46] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, Jan 2018.

[47] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[48] Haim Brézis. *Opérateurs maximaux monotones et semi-groupes de contractions dans les espaces de Hilbert.* Amsterdam : North-Holland Pub. Co., 1973.

[49] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[50] Enzo Busseti, Walaa M Moursi, and Stephen Boyd. Solution refinement at regular points of conic problems. *Computational Optimization and Applications*, 74(3):627–643, 2019.

[51] Camille Castera, Jérôme Bolte, Cédric Févotte, and Edouard Pauwels. An inertial newton algorithm for deep learning. *Journal of Machine Learning Research*, 22(134):1–31, 2021.

[52] Kumar Chellapilla, Sidd Puri, and Patrice Simard. High Performance Convolutional Neural Networks for Document Processing. In Guy Lorette, editor, *Tenth International Workshop*

on Frontiers in Handwriting Recognition, La Baule (France), October 2006. Université de Rennes 1, Suvisoft. http://www.suvisoft.com.

[53] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[54] F.H. Clarke. *Optimization and Nonsmooth Analysis*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1990.

[55] Frank H. Clarke. Generalized gradients and applications, 1975.

[56] Raf Cluckers and Daniel J. Miller. Stability under integration of sums of products of real globally subanalytic functions and their logarithms. *Duke Mathematical Journal*, 156(2):311–348, Feb 2011.

[57] Patrick L Combettes and Jean-Christophe Pesquet. Deep neural network structures solving variational inequalities. *Set-Valued and Variational Analysis*, 28(3):491–518, 2020.

[58] Patrick L Combettes and Valérie R Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

[59] Michel Coste. *An introduction to o-minimal geometry*. Institut de Recherche Mathématique de Rennes, 1999.

[60] Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 113–123, 2019.

[61] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989.

[62] Damek Davis and Dmitriy Drusvyatskiy. Conservative and semismooth derivatives are equivalent for semialgebraic maps. *Set-Valued and Variational Analysis*, pages 1–11, 2021.

[63] Damek Davis, Dmitriy Drusvyatskiy, Sham Kakade, and Jason D. Lee. Stochastic subgradient method converges on tame functions. *Foundations of Computational Mathematics*, 20(1):119–154, Feb 2020.

[64] Damek Davis and Liwei Jiang. A nearly linearly convergent first-order method for nonsmooth functions with quadratic growth, 2022.

[65] Stephan Dempe. *Foundations of Bilevel Programming*. Nonconvex Optimization and Its Applications. Springer New York, NY, 2002.

[66] Aymeric Dieuleveut, Gersende Fort, Eric Moulines, and Hoi-To Wai. Stochastic approximation beyond gradient for signal processing and machine learning, 2023.

[67] L. Dries and C. Miller. On the real exponential field with restricted analytic functions. *Israel Journal of Mathematics*, 92:427, 1995.

[68] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011.

[69] R. Durrett. *Probability: Theory and Examples*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2010.

[70] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407 – 499, 2004.

[71] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *Annals of statistics*, 32(2):407–499, 2004.

[72] Yuri Ermoliev and Vladimir Norkin. Stochastic generalized gradient method for nonconvex nonsmooth stochastic optimization. *Cybernetics and Systems Analysis*, 34(2):196–215, 1998.

[73] Lawrence C Evans and Ronald F Gariepy. *Measure theory and fine properties of functions*, volume 5. CRC press Boca Raton, 1992.

[74] Mathieu Even. Stochastic gradient descent under markovian sampling schemes. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[75] Mathieu Faure and Gregory Roth. Ergodic properties of weak asymptotic pseudotrajectories for set-valued dynamical systems. *Stochastics and Dynamics*, 13(01):1250011, 2013.

[76] Vincent Franç ois-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare, and Joelle Pineau. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354, 2018.

[77] Sébastien Gadat, Fabien Panloup, and Sofiane Saadane. Stochastic heavy ball. *Electronic Journal of Statistics*, 12(1):461–529, 2018.

[78] Zhenglin Geng, Daniel Johnson, and Ronald Fedkiw. Coercing machine learning to output physically accurate results. *Journal of Computational Physics*, 406:109099, 2020.

[79] Andreas Griewank and Andrea Walther. *Evaluating Derivatives*. Society for Industrial and Applied Mathematics, second edition, 2008.

[80] Fangda Gu, Heng Chang, Wenwu Zhu, Somayeh Sojoudi, and Laurent El Ghaoui. Implicit graph neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11984–11995. Curran Associates, Inc., 2020.

[81] Fangda Gu, Heng Chang, Wenwu Zhu, Somayeh Sojoudi, and Laurent El Ghaoui. Implicit graph neural networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[82] Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, 116:1–8, 2018.

[83] M. Gürbüzbalaban, A. Ozdaglar, and P. A. Parrilo. Why random reshuffling beats stochastic gradient descent. *Mathematical Programming*, 186(1):49–84, Mar 2021.

[84] Mert Gürbüzbalaban, Andrzej Ruszczyński, and Landi Zhu. A stochastic subgradient method for distributionally robust non-convex and non-smooth learning. *Journal of Optimization Theory and Applications*, 194(3):1014–1041, Sep 2022.

[85] Shunsuke Hayashi, Nobuo Yamashita, and Masao Fukushima. A combined smoothing and regularization method for monotone second-order cone complementarity problems. *SIAM Journal on Optimization*, 15(2):593–615, 2005.

[86] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[87] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[88] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[89] Franck Iutzeler and Jérôme Malick. Nonsmoothness in machine learning: Specific structure, proximal identification, and applications. *Set-Valued and Variational Analysis*, 28(4):661–678, Dec 2020.

[90] Sham M Kakade and Jason D Lee. Provably correct automatic sub-differentiation for qualified programs. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[91] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[92] Anastasia Koloskova, Nikita Doikov, Sebastian U. Stich, and Martin Jaggi. Shuffle sgd is always better than sgd: Improved analysis of sgd with arbitrary data orders. *ArXiv*, abs/2305.19259, 2023.

[93] Jakub Konečný, Brendan McMahan, and Daniel Ramage. Federated optimization:distributed optimization beyond the datacenter, 2015.

[94] Lingchen Kong, Levent Tunçel, and Naihua Xiu. Clarke generalized jacobian of the projection onto symmetric cones. *Set-Valued and Variational Analysis*, 17(2):135–151, 2009.

[95] Timo Kreimeier, Sebastian Pokutta, Andrea Walther, and Zev Woodstock. On a frank-wolfe approach for abs-smooth functions. *arXiv preprint arXiv:2303.09881*, 2023.

[96] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[97] Daniel Kuhn, Peyman Mohajerin Esfahani, Viet Anh Nguyen, and Soroosh Shafieezadeh-Abadeh. *Wasserstein Distributionally Robust Optimization: Theory and Applications in Machine Learning*. INFORMS TutORials in Operations Research null(null):130-166, 2019.

[98] Harold J. Kushner and Dean S. Clark. *Stochastic approximation methods for constrained and unconstrained systems*. Applied mathematical sciences, Springer, New York, 1978.

[99] Harold J. Kushner and George Yin. *Stochastic approximation and recursive algorithms and applications*. Applications of mathematics, Springer, New York, 2003.

[100] Tam Le. Nonsmooth nonconvex stochastic heavy ball, 2023.

[101] Quentin Le Lidec, Igor Kalevatykh, Ivan Laptev, Cordelia Schmid, and Justin Carpentier. Differentiable simulation for physical system identification. *IEEE Robotics and Automation Letters*, 6(2):3413–3420, 2021.

[102] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[103] Wonyeol Lee, Sejun Park, and Alex Aiken. On the correctness of automatic differentiation for neural networks with machine-representable parameters. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[104] Adrian S. Lewis and Tonghua Tian. The structure of conservative gradient fields. *SIAM Journal on Optimization*, 31(3):2080–2083, 2021.

[105] Lennart Ljung. Analysis of recursive stochastic algorithms. *IEEE Transactions on Automatic Control*, 22(4):551–575, 1977.

[106] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. *Proceedings of the 30th International Conference on Machine Learning*, 28(3.), 2013.

[107] Julien Mairal and Bin Yu. Complexity analysis of the lasso regularization path. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1835–1842, 2012.

[108] Szymon Majewski, Błażej Miasojedow, and Éric Moulines. Analysis of nonsmooth stochastic approximation: the differential inclusion approach. *arXiv: Optimization and Control*, 2018.

[109] Jérôme Malick and Hristo S Sendov. Clarke generalized jacobian of the projection onto the cone of positive semidefinite matrices. *Set-Valued Analysis*, 14(3):273–293, 2006.

[110] Swann Marx and Edouard Pauwels. Path differentiability of ode flows. *Journal of Differential Equations*, 338:321–351, 2022.

[111] Pertti Mattila. *Fourier Analysis and Hausdorff Dimension*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2015.

[112] Robert Mifflin. An algorithm for constrained optimization with semismooth functions. *Mathematics of Operations Research*, 2(2):191–207, 1977.

[113] Robert Mifflin. Semismooth and semiconvex functions in constrained optimization. *SIAM Journal on Control and Optimization*, 15(6):959–972, 1977.

[114] V. S. Mikhalevich, A. M. Gupal, and V. I. Norkin. *Nonconvex Optimization Methods*. Nauka, Moscow, 1987.

[115] Konstantin Mishchenko, Ahmed Khaled, and Peter Richtarik. Random reshuffling: Simple analysis with vast improvements. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17309–17320. Curran Associates, Inc., 2020.

[116] Jean Jacques Moreau. Décomposition orthogonale d'un espace hilbertien selon deux cônes mutuellement polaires. *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, 255:238–240, 1962.

[117] Jean Jacques Moreau. Fonctionnelles sous-différentiables. *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, 257:4117–4119, 1963.

[118] Jean Jacques Moreau. Evolution problem associated with a moving convex set in a hilbert space. *Journal of Differential Equations*, 26(3):347–374, 1977.

[119] Vladimir Norkin. Nonlocal minimization algorithms of nondifferentiable functions. *Cybernetics and Systems Analysis*, 14:704–707, 09 1978.

[120] Vladimir Norkin. Generalized-differentiable functions. *Cybernetics and Systems Analysis*, 16:10–12, 01 1980.

[121] Vladimir Norkin. Stochastic generalized-differentiable functions in the problem of nonconvex nonsmooth stochastic optimization. *Cybernetics and Systems Analysis*, 22:804–809, 11 1986.

[122] Vladimir Norkin. Stochastic generalized gradient methods for training nonconvex nonsmooth neural networks. *Cybernetics and Systems Analysis*, 57(5):714–729, Sep 2021.

[123] Evgeni Nurminski. Minimization of nondifferentiable functions in the presence of noise. *Cybernetics and Systems Analysis*, 10:619–621, 07 1974.

[124] Antonio Orvieto, Jonas Kohler, and Aurelien Lucchi. The role of memory in stochastic optimization. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 356–366. PMLR, 22–25 Jul 2020.

[125] Michael R. Osborne, Brett Presnell, and Berwin A. Turlach. On the lasso and its dual. *Journal of Computational and Graphical Statistics*, 9(2):319–337, 2000.

[126] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[127] Edouard Pauwels. Conservative parametric optimality and the ridge method for tame min-max problems, 2023.

[128] Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[129] David Preiss and Gareth Speight. Differentiability of lipschitz functions in lebesgue null sets. *Inventiones mathematicae*, 197, 04 2013.

[130] Liqun Qi and Jie Sun. A nonsmooth version of newton's method. *Mathematical Programming*, 58(1):353–367, Jan 1993.

[131] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[132] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, Los Alamitos, CA, USA, jun 2016. IEEE Computer Society.

[133] Rodolfo Ríos-Zertuche. Examples of pathological dynamics of the subgradient method for lipschitz path-differentiable functions. *Mathematics of Operations Research*, 47(4):3184–3206, 2022.

[134] Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951.

[135] Ralph Tyrrell Rockafellar. *The theory of subgradients and its applications to problems of optimization: Convex and nonconvex functions*. Heldermann, 1981.

[136] Ralph Tyrrell Rockafellar and Roger J.-B. Wets. *Variational Analysis*. Springer Verlag, Heidelberg, Berlin, New York, 1998.

[137] Cédric Rommel, Thomas Moreau, Joseph Paillard, and Alexandre Gramfort. CADDA: Class-wise automatic differentiable data augmentation for EEG signals. In *International Conference on Learning Representations*, 2022.

[138] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[139] Halsey L. Royden. *Real Analysis*. Collier Macmillan international editions. Macmillan, 1968.

[140] W. Rudin. *Principles of Mathematical Analysis*. International series in pure and applied mathematics. McGraw-Hill, 1976.

[141] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[142] Andrzej Ruszczyński. Convergence of a stochastic subgradient method with averaging for nonsmooth nonconvex constrained optimization. *Optimization Letters*, 14(7):1615–1625, Oct 2020.

[143] Andrzej Ruszczyński. A stochastic subgradient method for nonsmooth nonconvex multilevel composition optimization. *SIAM J. Control. Optim.*, 59:2301–2320, 2021.

[144] Andrzej Ruszczyński and Alexander Shapiro. *Optimization of Risk Measures*, pages 119–157. Springer London, London, 2006.

[145] Michael E. Sander, Joan Puigcerver, Josip Djolonga, Gabriel Peyré, and Mathieu Blondel. Fast, differentiable and sparse top-k: A convex analysis perspective. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[146] Arthur Sard. The measure of the critical values of differentiable maps. *Bulletin of the American Mathematical Society*, 48(12):883 – 890, 1942.

[147] James Serrin and Dale E Varberg. A general chain rule for derivatives and the change of variables formula for the lebesgue integral. *The American Mathematical Monthly*, 76(5):514–520, 1969.

[148] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on Stochastic Programming*. Society for Industrial and Applied Mathematics, 2009.

[149] Alexander Shapiro and Huifu Xu. Uniform laws of large numbers for set-valued mappings and subdifferentials of random functions. *Journal of Mathematical Analysis and Applications*, 325:1390–1399, 01 2007.

[150] M Streeter and H. McMahan. Less regret via online conditioning. In *Proc. of the 23rd Conference on Learning Theory*, pages 244–256, 2010.

[151] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[152] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[153] Ryan J. Tibshirani. The lasso problem and uniqueness. *Electronic Journal of statistics*, 7:1456–1490, 2013.

[154] Tijmen Tieleman and Geoffrey Hinton. Lecture 6. *5-rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude. COURSERA: Neural Networks for Machine Learning*, 4:26–31, 2012.

[155] Michel Valadier. *Entrainement unilateral, lignes de descente.* fonctions lipschitziennes non pathologiques, 1989.

[156] Lou van den Dries and Chris Miller. Geometric categories and o-minimal structures. *Duke Math. J.*, 84(2):497–540, 08 1996.

[157] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[158] Andrea Walther and Andreas Griewank. Getting started with adol-c. In U. Naumann and O. Schenk, editors, *Combinatorial Scientific Computing*, chapter 7, pages 181–202. Chapman-Hall CRC Computational Science, 2012.

[159] Jack Warga. Fat homeomorphisms and unbounded derivate containers. *Journal of Mathematical Analysis and Applications*, 81:545–560, 1981.

[160] Hassler Whitney. Tangents to an analytic variety. *Annals of Mathematics*, 81(3):496–549, 1965.

[161] Hassler Whitney. *A Function Not Constant on a Connected Set of Critical Points*, pages 290–293. Birkhäuser Boston, Boston, MA, 1992.

[162] Ezra Winston and J. Zico Kolter. Monotone operator equilibrium networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[163] J. J. Ye and D. L. Zhu. Optimality conditions for bilevel programming problems. *Optimization*, 33(1):9–27, Jan 1995.

[164] Zygmunt Zahorski. Sur l'ensemble des points de non-dérivabilité d'une fonction continue. *Bulletin de la Société Mathématique de France*, 74:147–178, 1946.

[165] Matt Johnson Zico Kolter, David Duvenaud. Deep implicit layers - neural odes, deep equilibirum models, and beyond, 2020. NeurIPS Tutorial.

[166] Hui Zou. The adaptive lasso ad its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 02 2006.

Résumé en français

Les problèmes d'apprentissage automatique se formulent souvent comme la minimisation d'un risque présentant de la non-lissité et de la non-convexité. Des sources importantes de non-lissité sont l'utilisation privilégiée d'instructions conditionnelles et la présence de sous-problèmes.

Les méthodes stochastiques du premier ordre sont largement utilisées pour aborder ces problèmes en raison de leur simplicité et de leur scalabilité, ce qui en fait un choix attrayant pour les applications à grande échelle. Bien que les notions classiques de dérivées pour les fonctions non-lisses soient difficilement applicables dans de tels contextes, nous observons une tendance générale consistant à remplacer les dérivées classiques par l'algorithme de rétropropagation. Un modèle récemment introduit de dérivées appelé "gradients conservatifs" fournit une justification à une telle pratique en étendant les règles simples du calcul aux fonctions non-lisses, telles que la règle de la chaîne ou la somme.

Nous proposons deux extensions du calcul conservatif qui trouvent un large éventail d'applications en apprentissage automatique. Un premier résultat répond à la question de l'interversion de la dérivation et de l'intégrale, ce qui permet de justifier l'échantillonnage du premier ordre dans un cadre non-lisse et non-convexe. Un deuxième résultat est une formule de différentiation implicite non-lisse afin de justifier les approches du premier ordre pour les problèmes bi-niveaux non-lisses, tels que l'optimisation des hyperparamètres et l'entraînement de couches implicites dans l'apprentissage profond.

Nous utilisons ce calcul afin de mettre en place un cadre général stochastique non-lisse compatible avec les implémentations pratiques. Une règle de la chaîne fondamentale le long des courbes permet d'appliquer des méthodes d'ODE non-lisses afin de démontrer la convergence de la méthode du sous-gradient stochastique et de sa version avec boule pesante. Certains résultats d'intégration de fonctions définissables sont explorés afin de garantir une propriété de Sard pour des distributions continues.

En tant que modèle fidèle à la pratique, l'approche des gradients conservatifs conduit à une convergence vers un ensemble critique qui peut dépendre du calcul et conduire à des points limites absurdes. Pour la méthode du sous-gradient stochastique et sa version boule pesante, nous montrons que ces artefacts de calcul sont évités en randomisant l'initialisation, ce qui conduit à la convergence vers des points critiques classiques.

Chapitre 0

Introduction en français

L'apprentissage automatique est désormais un véritable atout dans notre société et est utilisé pour de nombreuses tâches complexes, notamment les systèmes de recommandation, la reconnaissance d'images et de la parole, les chatbots, les jeux et la compréhension de scènes. Le deep learning [102] a révolutionné ce domaine et a connu une croissance rapide au cours de la dernière décennie.

L'évolution du deep learning a été marquée par des réalisations remarquables. Tout a commencé avec le succès du réseau neuronal convolutif AlexNet [96] lors du défi ImageNet de 2012. Cette percée a démontré le potentiel du deep learning dans les tâches de vision par ordinateur. Par la suite, des modèles de deep learning tels qu'AlphaGo, qui excellait dans le jeu de société Go, et AlphaFold, qui a fait des avancées significatives dans la prédiction de la structure des protéines, ont encore démontré l'efficacité du deep learning dans des domaines complexes. D'autres exemples notables comprennent Dall-E, un modèle d'IA[1] pour l'art génératif, et le chatbot ChatGPT. La performance empirique de ces algorithmes de deep learning est souvent privilégiée par les praticiens par rapport aux garanties théoriques. Les démonstrations de performances supérieures alimentent la tendance actuelle dans le domaine, tandis que les aspects non expliqués des modèles de deep learning en font un domaine de recherche actif.

Pour une compréhension plus approfondie et une amélioration des modèles d'apprentissage automatique, des concepts et des techniques issus de domaines classiques tels que la statistique et l'optimisation ont été redécouverts dans ce contexte. Par exemple, l'entraînement d'un réseau neuronal peut être considéré comme un problème d'optimisation, ce qui permet de tirer parti d'algorithmes d'optimisation efficaces [46]. Néanmoins, en raison de leur développement historique ou de leurs applications pratiques, les modèles d'apprentissage automatique présentent souvent des propriétés qui posent des défis du point de vue traditionnel. L'interprétation des prédictions d'un réseau neuronal et la détermination de la convergence pendant le processus d'entraînement sont quelques-unes des questions complexes qui se posent dans ce domaine.

Dans cette thèse, nous accordons une attention particulière à l'aspect non-lisse[2] des problèmes d'apprentissage automatique, qui pose des problèmes d'optimisation. Les opérations courantes en apprentissage automatique, telles que la prise du maximum, le seuillage des valeurs ou l'incorporation de contraintes polyédrales, introduisent des points de non-différentiabilité qui nécessitent une analyse spécifique.

[1] Intelligence artificielle

[2] Le terme "non-lisse" a plusieurs interprétations dans la littérature. Dans notre contexte, nous le définirons comme un manque de différentiabilité en certains points.

0.1 Problèmes d'apprentissage automatique non-lisses

L'entraînement des modèles d'apprentissage automatique peut être vu comme la minimisation d'un risque :

$$\min_{w\in\mathbb{R}^p} F(w) := \mathbb{E}_{\xi\sim P}[f(w,\xi)], \tag{1}$$

où P représente une distribution d'échantillons de données, et f est un critère à minimiser en moyenne. Dans de nombreux problèmes d'apprentissage automatique, et en particulier dans le deep learning, la fonction F à minimiser est non-lisse et non-convexe. Alors que la non-lissité peut être appréhendée dans plusieurs contextes, par exemple lorsqu'elle est accompagnée de convexité ou d'une structure spécifique [18, 89], les situations que nous considérons nécessitent un traitement spécifique en raison de leur aspect général non-lisse et non-convexe.

Dans cette thèse, nous examinerons deux sources de non-lissité qui trouvent un large éventail d'applications. La première concerne les réseaux neuronaux utilisés pour de nombreuses tâches de prédiction, et l'autre concerne les problèmes bi-niveaux qui se posent, par exemple, dans l'optimisation des hyperparamètres.

0.1.1 Réseaux neuronaux

Apprentissage supervisé. Le problème de minimisation (1) englobe une large classe de problèmes en apprentissage automatique appelés *apprentissage supervisé*. Dans ce type de problème, l'objectif est de prédire une variable cible $Y \in \mathbb{R}^I$ étant donné une entrée $X \in \mathbb{R}^d$, en d'autres termes, apprendre une relation $h(X) \approx Y$. Y peut être continue (régression) ou discrète (classification). Dans ce contexte, la variable aléatoire ξ est le couple entrée-sortie (X, Y), et l'intégrande f dans (1) s'écrit généralement comme suit :

$$f(w, X, Y) = \ell(h(w, X), Y). \tag{2}$$

$h(w,\cdot)$ est une fonction de prédiction paramétrée par w et ℓ est une mesure de dissimilarité. Les choix classiques pour la fonction ℓ sont la distance quadratique, en régression, $\ell(u,v) = \|u - v\|^2$, et l'entropie croisée en classification. La distribution P est souvent inconnue, donnée par la nature. En pratique, on peut disposer de plusieurs échantillons de P, $(x_i, y_i)_{i=1,\dots,n}$, supposés tirés indépendamment. Ainsi, pour apprendre un prédicteur, on peut minimiser la perte empirique :

$$\min_{w\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n} f(w, x_i, y_i). \tag{3}$$

Dans des contextes en ligne, les échantillons peuvent également être obtenus à partir d'un flux de données, auquel cas l'espérance (1) est minimisée séquentiellement.

Réseaux neuronaux. Dans (2), la fonction de prédiction peut prendre de nombreuses formes. Nous nous intéressons aux prédicteurs utilisés dans le deep learning, appelés *réseaux neuronaux* (artificiels). Les réseaux neuronaux classiques sont construits à partir de la composition de fonctions non linéaires $(\sigma_l)_{l=0,\dots,L}$ et de transformations affines paramétrées par $(A_l, b_l)_{l=0,\dots,L}$:

$$\begin{aligned} h(w,x) &= \sigma_L(A_L h_L + b_L) \\ h_L &= \sigma_{L-1}(A_{L-1}h_{L-1} + b_{L-1}) \\ &\dots \\ h_1 &= \sigma_0(A_0 x + b_0), \end{aligned} \tag{4}$$

où $L+1$ est le nombre de couches. Lorsque σ_l est une fonction de $\mathbb{R}$ dans $\mathbb{R}$ appliquée de manière composante par composante, elle est souvent appelée une *fonction d'activation*, par analogie avec l'activation d'un neurone. Le paramètre w à optimiser est la concaténation vectorielle de toutes les matrices $(A_l)_{l=0,\dots,L}$ et de tous les vecteurs $(b_l)_{l=0,\dots,L}$. À partir d'une telle structure compositionnelle, on cherche à apprendre des relations complexes $h(x) \approx y$, et le choix des fonctions non linéaires σ_l est donc essentiel.

Alors qu'un réseau neuronal à deux couches peut approximer n'importe quelle fonction continue [61], il a été démontré empiriquement que les réseaux neuronaux avec plus de couches étaient plus performants pour diverses tâches impliquant de grands ensembles de données. Parmi ces tâches, on peut citer la reconnaissance de caractères, la reconnaissance d'objets et de la parole, ou encore le traitement du langage naturel. En conséquence, le nombre de paramètres et de couches peut être très élevé en pratique. Par exemple, AlexNet [96] comporte 60 millions de paramètres pour 8 couches et a été entraîné sur un ensemble de données de 1,2 million d'images. Les réseaux résiduels [86] ont jusqu'à 1,7 million de paramètres répartis sur 110 couches. GPT-3 [49], un modèle de langage, possède 175 milliards de paramètres.

De plus, il existe de nombreuses architectures de réseaux neuronaux. Les réseaux convolutionnels tels que AlexNet sont souvent utilisés sur des données d'images et utilisent des convolutions matricielles qui peuvent être représentées par des matrices circulantes $(A_l)_{l=1,\dots,L}$. Les réseaux résiduels utilisent des sauts de connexion et les réseaux neuronaux récurrents [88], utilisés sur des données textuelles, utilisent une réinjection d'entrée qui peut être représentée dans (4) en fixant certaines valeurs des matrices. Les matrices $(A_l)_{l=1,\dots,L}$ peuvent être contraintes d'être égales, comme dans les réseaux en treillis [13].

L'une des fonctions d'activation les plus populaires est la fonction positive, communément appelée "ReLU" (abrégé de "Rectified Linear Unit") par la communauté de l'apprentissage profond (figure 1). Cette fonction est récurrente dans l'apprentissage profond et est utilisée dans de nombreux modèles tels que les réseaux convolutifs et résiduels [86, 96] ou les blocs transformers [157] utilisés dans les modèles de langage.

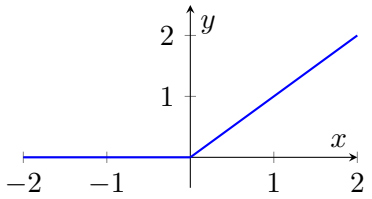


FIGURE 1 : Fonction ReLU

Une autre transformation non linéaire largement utilisée sur les données d'images est le MaxPooling. Étant donné une matrice d'entrée X écrite sous forme d'une matrice de blocs de taille d, avec le choix habituel $d=2$, la fonction MaxPooling renvoie la matrice où chaque composante est le maximum d'un bloc, permettant une réduction de l'image. Voici un exemple de la fonction MaxPooling avec une fenêtre de taille 2 :

$$\text{MaxPooling}\left(\begin{bmatrix} 3 & 1 & \mathbf{13} & 8 \\ 7 & \mathbf{20} & 6 & 2 \\ \mathbf{7} & 3 & \mathbf{5} & 4 \\ 6 & 2 & 1 & 0 \end{bmatrix}\right) = \begin{bmatrix} \mathbf{20} & \mathbf{13} \\ \mathbf{7} & \mathbf{5} \end{bmatrix}.$$

La fonction MaxPooling apparaît dans de nombreuses architectures telles que les réseaux résiduels pour la classification d'images ou les modèles YOLO [132] pour la détection d'objets en temps réel.

Comme nous pouvons le constater, les modèles d'apprentissage profond sont souvent non-lisses. La fonction ReLU n'est pas différentiable en zéro, tandis que la fonction MaxPooling n'est pas différentiable lorsque certaines composantes d'un bloc sont égales. Cela peut soulever des préoccupations si nous considérons l'entraînement comme un problème d'optimisation continue, où la différentiabilité est généralement souhaitable. Pourtant, l'omniprésence de la non-lissité n'est pas vraiment justifiée et bien que l'utilisation de la fonction ReLU démontre un certain succès, on

ne sait pas si la non-lissité est strictement nécessaire. Par exemple, certaines parties du modèle GPT-3 utilisent des fonctions d'activation lisses telles que GeLU ou softmax [49].

En fait, avant leur succès, le développement des réseaux de neurones artificiels était plutôt indépendant du domaine de l'optimisation. Les premières tentatives pour résoudre des tâches de classification impliquaient des représentations simplifiées des réseaux de neurones, notamment le modèle du perceptron introduit par Rosenblatt [138]. Dans ces modèles primaires, les connexions synaptiques étaient représentées à l'aide de produits matriciels et les activations neuronales étaient exprimées par des sorties binaires ou des valeurs seuillées. Malgré leur non-lissité, certains de ces aspects simplistes semblent persister même dans les modèles de pointe, comme l'illustre la fonction ReLU.

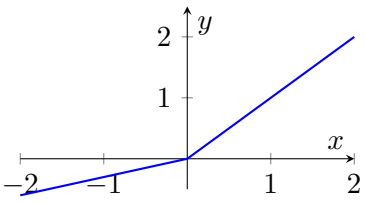


FIGURE 2 : Leaky ReLU

Les modèles d'apprentissage profond évoluent continuellement, s'adaptant à de nouveaux problèmes. Le domaine explore des architectures et des fonctions d'activation alternatives, par exemple, la fonction de tri [7] a été utilisée pour promouvoir le caractère lipschitzien. De nombreuses variantes de la fonction ReLU existent également, comme la fonction "leaky ReLU" [106], figure 2. Dans cette thèse, nous nous intéresserons à des classes de couches introduites récemment et appelées couches implicites et couches d'optimisation. Les couches d'optimisation seront discutées dans la section 0.1.2 en raison de leur aspect bi-niveau.

Couches implicites. Introduits dans des travaux récents [12, 80], certains réseaux de neurones utilisent des couches dont la sortie est définie par une équation implicite. Par exemple, dans les réseaux d'équilibre profonds [12], la sortie z d'une couche est définie par une équation de point fixe :

$$z = \sigma(Wz + b + Ux), \tag{5}$$

(W, U, b) sont les paramètres à entraîner, x est l'entrée et σ joue le rôle d'une fonction d'activation. Par exemple, σ peut être la fonction ReLU. Ce type de couche est inspiré du modèle des réseaux à treillis [13], où les matrices des couches sont contraintes à être égales afin d'avoir moins de paramètres à entraîner. L'équation du point fixe (5) permet non seulement d'obtenir un modèle avec moins de paramètres, mais élimine également la nécessité de stocker les sorties des couches intermédiaires, réduisant ainsi les coûts de mémoire lors de l'entraînement. Malgré la réduction du nombre de paramètres, ces modèles présentent des performances compétitives par rapport aux réseaux neuronaux profonds traditionnels.

0.1.2 Problèmes bi-niveaux

Dans certaines situations, la fonction F à minimiser est définie à partir d'un autre problème d'optimisation, ce qui entraîne de la non-lissité. Dans les problèmes *bi-niveaux*, l'objectif implique un terme argmin :

$$\min_{w \in \mathbb{R}^p} F(w, z) \quad \text{tel que } z \in \operatorname*{argmin}_{\theta \in \mathcal{C}} g(w, \theta).$$

Ce type de problème a été étudié en optimisation auparavant [65, 163] et suscite aujourd'hui un intérêt renouvelé en apprentissage automatique avec plusieurs applications telles que l'augmentation de données [60, 137] et l'optimisation d'hyperparamètres [23, 25, 26]. Deux applications seront d'intérêt dans le chapitre 3, section 3.3.3 de cette thèse, l'optimisation d'hyperparamètres pour les

modèles de type lasso [25] et les couches d'optimisation convexe [6], en accord avec la programmation conique différentiable [2, 3].

Couches d'optimisation. Les couches d'optimisation, qui ont été récemment introduites dans [2, 6], représentent un nouveau type d'architecture où la sortie de la couche est obtenue comme solution d'un problème d'optimisation convexe. L'entraînement de tels modèles s'écrit donc comme un problème bi-niveau. Ces couches d'optimisation ont démontré leur efficacité dans de nombreuses applications lorsqu'il s'agit de modéliser des connaissances a priori et d'apprendre des contraintes [6, 78, 101].

Optimisation d'hyperparamètres. Plusieurs problèmes d'apprentissage automatique intègrent un terme de régularisation R :

$$\min_{w \in \mathbb{R}^p} F(w) + R(\lambda, w). \tag{6}$$

Les choix classiques sont la norme au carré, $R(\lambda, w) = \lambda \|w\|^2$, utilisée en apprentissage profond sous le nom de "weight decay" [96], ou la norme ℓ_1 $\lambda \|w\|_1$. La pénalisation de la norme ℓ_1 suscite un grand intérêt en apprentissage automatique et en statistiques en raison de sa propriété de fournir des solutions parcimonieuses, permettant la sélection de variables en haute dimension, avec l'estimateur lasso [152], ou la récupération de signaux parcimonieux par poursuite de base [53]. Une question récurrente lors de l'ajout d'un terme de régularisation est celle du choix du niveau de pénalité, c'est-à-dire de l'hyperparamètre λ. Une approche courante consiste à procéder par validation croisée et à maximiser un critère par rapport au niveau de pénalité, ce qui peut être formulé comme un problème bi-niveau. Lorsque l'hyperparamètre est unidimensionnel, une recherche sur une grille est généralement suffisante.

Cependant, dans certaines variantes de l'estimateur lasso, le terme de régularisation peut contenir plus d'un hyperparamètre, comme dans le lasso adaptatif [166], qui vise à réduire le biais de l'estimateur tout en préservant la parcimonie. Ainsi, une question pertinente se pose quant à l'application d'algorithmes d'optimisation plus efficaces en utilisant des méthodes du premier ordre [25]. Dans le cas de l'estimateur lasso, le chemin de solution est morceau linéaire par rapport à l'hyperparamètre [70], ce qui conduit à un problème bi-niveau non-lisse.

D'autres problèmes, qui ne sont pas traités dans cette thèse, peuvent également impliquer la fonction valeur du problème de sous-niveau et générer également une non-lissité. Dans les problèmes min-max, F est définie comme un maximum ponctuel,

$$F(w) := \max_{\theta \in \mathcal{C}} g(w, \theta).$$

Certaines applications typiques de cette configuration sont les réseaux génératifs antagonistes pour la génération d'images [8], l'optimisation robuste sous incertitude et l'optimisation averse au risque [84, 97, 144].

0.2 Optimisation du premier ordre en apprentissage automatique

Pour le moment, mettons de côté l'aspect non-lisse. Dans le cadre différentiable, les méthodes du premier ordre, comme la méthode du gradient (7), sont souvent utilisées pour traiter le problème de minimisation (1).

$$\text{pour } k \in \mathbb{N}, \quad w_{k+1} = w_k - \alpha_k \nabla F(w_k). \tag{7}$$

Cette popularité peut s'expliquer par leur simplicité, le développement récent de logiciels efficaces pour calculer les gradients, la *différentiation automatique* [79] ou la *rétropropagation* [141], et des moyens plus efficaces d'exploiter la puissance de calcul avec l'utilisation adaptée des GPU[3] [52] afin de traiter un grand nombre de paramètres.

0.2.1 Mise en pratique des méthodes du premier ordre

Dans cette partie, nous exposons quelques pratiques en ce qui concerne la mise en œuvre des méthodes du premier ordre en apprentissage automatique. Pour calculer le gradient d'une fonction différentiable $f : \mathbb{R}^p \to \mathbb{R}$, une approche simple consisterait à approximer les dérivées partielles de f par différences finies :

$$\nabla f(w) \approx \frac{1}{t} \begin{bmatrix} f(w + te_1) - f(w) \\ f(w + te_2) - f(w) \\ \vdots \\ f(w + te_p) - f(w) \end{bmatrix}$$

où t est suffisamment petit et, pour $i = 1, \dots, p$, e_i est le i-ème élément de la base canonique de $\mathbb{R}^p$. Le coût de cette méthode est d'environ $p \times \text{coût}(f)$ où $\text{coût}(f)$ est le coût de calcul de f. Dans de nombreuses situations d'apprentissage automatique telles que l'apprentissage profond, la dimension des paramètres p est élevée, ce qui rendrait l'utilisation de cette méthode déraisonnable.

Différentiation automatique. La différentiation automatique [79], également appelée "rétropropagation" dans la communauté de l'apprentissage profond [102, 141], est un algorithme efficace pour calculer le gradient d'une fonction $f : \mathbb{R}^p \to \mathbb{R}$ en automatisant la règle de la chaîne sur les fonctions élémentaires disponibles dans un langage de programmation (exponentielle, logarithme, sinus, cosinus...). Cela permet de calculer les gradients de manière plus efficace et exacte. La différentiation automatique est disponible dans des bibliothèques Python telles que TensorFlow [1], PyTorch [126] ou JAX [47].

Pour les fonctions rationnelles, un résultat fondamental de Baur et Strassen [17] montre que le coût de la différentiation automatique est au plus 5 fois le coût de calcul de la fonction à différencier. Ce résultat a été étendu aux fonctions différentiables [79]. Comparé à la méthode des différences finies, le coût de calcul en termes de coût de la fonction n'augmente pas avec la dimension.

En apprentissage profond, cette méthode peut poser certains problèmes. En particulier, le calcul de la formule de la règle de chaîne pour la composition (4) nécessite que les valeurs intermédiaires $(h_l)_{l=0,\dots,L}$ de la composition soient stockées, ce qui peut être coûteux en termes de mémoire lorsque le nombre de couches L est élevé. Certaines solutions ont été proposées dans la littérature pour réduire ce coût en mémoire, telles que l'utilisation de couches implicites (section 0.1.1).

Différentiation implicite. Nous nous concentrerons particulièrement sur la différentiation des fonctions définies implicitement qui apparaissent par exemple dans les modèles implicites, présentés dans la section 0.1.1. Dans le cadre différentiable, pour une équation $H(w, y) = 0$, le théorème des fonctions implicites garantit, sous certaines conditions, l'existence et la différentiabilité d'une application solution $y^*(w)$, conduisant à une dérivée de y par rapport à w :

$$\operatorname{Jac} y^*(w) = -(\operatorname{Jac}_y H(w, y))^{-1} \operatorname{Jac}_x H(w, y). \tag{8}$$

[3] Unité de traitement graphique.

Certains travaux [2, 3, 25] ont proposé de l'utiliser afin de traiter des problèmes bi-niveaux, en dérivant les conditions d'optimalité du problème de sous-niveau. Dans le cas de problèmes convexes simples, cette méthode s'avère être très flexible car la dérivation des conditions d'optimalité peut être automatisée via la programmation convexe disciplinée [3, 4].

Échantillonnage du premier ordre. Dans le problème de minimisation stochastique (1) où F s'écrit sous la forme d'une espérance, le calcul du gradient ∇F ou l'application de la différentiation automatique n'est généralement pas réalisable dans les contextes à grande échelle. Dans les procédures d'apprentissage impliquant un ensemble de données volumineux, le risque empirique (3) devient une somme importante, de sorte que calculer ∇F à chaque itération est trop coûteux. Dans les contextes en ligne, les échantillons d'une distribution inconnue P arrivent successivement, auquel cas la méthode classique du gradient ne peut pas être appliquée.

Pour faire face à ces situations, il est courant de considérer une méthode de gradient stochastique, remontant au travail fondateur de Robbins et Monro [134]. Cette méthode s'écrit dans le cadre différentiable comme suit : pour $k \in \mathbb{N}$, on effectue les étapes suivantes :

$$\begin{aligned} &\text{échantillonner } \xi_k \sim P \\ &w_{k+1} = w_k - \alpha_k \nabla_w f(w_k, \xi_k) \end{aligned} \tag{9}$$

où $\nabla_w f(\cdot, \xi_k)$ est le gradient de $f(\cdot, \xi_k)$. Bien que cela ne soit pas étudié dans cette thèse, des améliorations supplémentaires peuvent être apportées à cet algorithme. Par exemple, la moyenne de plusieurs gradients stochastiques peut aider à réduire la variance. En pratique, une permutation aléatoire de la base de données est également utilisé à la place de l'échantillonnage aléatoire, ce qui conduit à une convergence plus rapide [83, 92, 115].

Cette méthode est cohérente avec la méthode du gradient déterministe, car l'espérance du gradient stochastique $\nabla f(\cdot, \xi)$, avec $\xi \sim P$, est égale au gradient de l'espérance F, grâce à la règle de l'intégrale de Leibniz, voir par exemple [140, Chapitre 9], qui permet d'intervertir l'espérance et le gradient.

0.2.2 Sur les variantes de la méthode du gradient utilisées en apprentissage automatique

Dans le contexte de l'apprentissage profond, nous constatons le développement de nombreuses variantes de la méthode du gradient stochastique dans le but d'accélérer les phases d'entraînement intensives.

Méthode inertielle. Introduite par Polyak [128], la méthode de boule pesante est une variante de la méthode du gradient qui inclut un terme inertiel,

$$w_{k+1} = w_k - \mu_k \nabla F(w_k) + \nu_k (w_k - w_{k-1}),$$

et peut être considérée avec un échantillonnage du premier ordre. Bien qu'aucune étude théorique n'explique son succès dans le cadre non-lisse et non-convexe, cette méthode est largement utilisée en apprentissage profond, popularisée par des travaux pionniers [96, 151].

Méthodes adaptatives. Une question récurrente qui se pose avec la méthode du gradient est le choix des pas $(\alpha_k)_{k\in\mathbb{N}}$. Dans le contexte des modèles d'apprentissage profond, le choix optimal des pas reste inconnu. En raison des longues phases d'entraînement, les pas décroissants suggérés par la littérature en approximation stochastique [134] peuvent entraîner des problèmes numériques qui ralentissent le processus d'entraînement. D'un autre côté, ajuster un pas constant peut être coûteux. Pour ces raisons, on préfère souvent des pas adaptatifs. Par exemple, AdaGrad [68, 150] définit des pas adaptatifs en fonction des évaluations passées des gradients. Sa version scalaire s'écrit comme suit :

$$w_{k+1} = w_k - \frac{1}{\sqrt{\epsilon + \sum_{i=0}^{k} \|\nabla F(w_i)\|^2}} \nabla F(w_k)$$

D'autres variantes de pas adaptatifs existent, comme RMSProp [154]. De plus, l'inertie et les pas adaptatifs peuvent également être combinés, comme dans ADAM [91] et AMSGrad [131]. La méthode ADAM est largement utilisée en apprentissage profond en raison de son succès empirique et elle est incluse dans les bibliothèques de différentiation automatique [1, 126].

0.2.3 Mise en œuvre des méthodes du premier ordre sur des fonctions non-lisses

Revenons maintenant au cadre non-lisse. Malgré la présence ubiquitaire de la non-lissité en apprentissage automatique, comme illustré dans les exemples de la section 0.1, les praticiens n'ont pas été dissuadés d'utiliser des méthodes du premier ordre. En fait, on observe une tendance générale consistant à utiliser la différentiation automatique sur des fonctions non-lisses et à remplacer les dérivées classiques par les sorties de la différentiation automatique.

Différentiation automatique pour les fonctions non-lisses. En apprentissage profond, la différentiation automatique peut être appliquée aux fonctions non-lisses. Les points de non-différentiabilité des fonctions implémentées en pratique sont générés par des instructions conditionnelles (*if* et *else*), ce qui permet d'appliquer la différentiation automatique à chaque partie de l'arbre computationnel. Pour comprendre ce mécanisme, considérons la représentation suivante de ReLU avec des instructions conditionnelles :

$$\mathrm{relu}(x) = \begin{cases} x, & \text{si } x > 0 \\ 0, & \text{sinon.} \end{cases}$$

La sortie de la différentiation automatique sera la suivante :

$$\mathrm{backprop}\,\mathrm{relu}(x) = \begin{cases} 1, & \text{si } x > 0 \\ 0, & \text{sinon.} \end{cases}$$

Un point important à noter ici est que la sortie de la différentiation automatique dépend de l'implémentation de la fonction. La différentiation automatique agit en fait sur le programme représentant la fonction, et non sur la fonction elle-même. Par exemple, changer l'instruction conditionnelle $x > 0$ en $x \geq 0$ ne changerait pas la valeur de la fonction ReLU, mais cela modifierait la valeur de la différentiation automatique en zéro, passant de 0 à 1.

Cette procédure permet d'obtenir une "différentiation automatique formelle". Par exemple, pour fournir un oracle du premier ordre à la composition de fonctions non-lisses, la formule de la règle de la chaîne peut être appliquée en remplaçant les Jacobians classiques par les sorties de la différentiation automatique. Cette façon de différentier les fonctions non-lisses est à la base

de l'entraînement des réseaux neuronaux [102, 141]. Un travail récent [31] étend le résultat de complexité de Baur et Strassen [17] pour les compositions non-lisses, justifiant ainsi son efficacité computationnelle dans les applications d'apprentissage profond. Certains modèles mathématiques [37, 122], qui seront exposés dans la section 0.4, ont été proposés dans le contexte de l'apprentissage profond pour justifier cette règle de différentiation des fonctions non-lisses.

Nous examinerons deux situations où la différentiation automatique est appliquée formellement : la différentiation implicite et l'échantillonnage du premier ordre.

Différentiation implicite non-lisse. Dans les cas pratiques mis en évidence dans la section 0.1, la relation implicite H est souvent non-lisse. Par exemple, les couches implicites (5) peuvent impliquer une fonction non-lisse σ telle que ReLU.

Les conditions d'optimalité des problèmes de sous-niveau qui se posent dans l'optimisation des hyperparamètres ou dans les couches d'optimisation convexe sont généralement exprimées par une application lipschitzienne et non-lisse.

Malgré cela, en pratique [12, 25], la formule de différentiation implicite (8) est appliquée aux fonctions non-lisses grâce à l'algorithme de rétropropagation. Pour fournir un oracle pour le chemin de solution $y^*(w)$ de $H(w, y) = 0$, on peut remplacer le Jacobien classique (8) par la sortie de la rétropropagation appliquée à H :

$$\text{Implicitdiff}\, y^*(w) := -(\text{backprop}_y\, H(w, y))^{-1}\, \text{backprop}_x\, H(w, y) \tag{10}$$

Bien que cette formule coïncide avec le Jacobien de y^* dans le cadre différentiable, la théorie actuelle ne la justifie pas dans le cas des fonctions non-lisses. Une contribution majeure de cette thèse dans le Chapitre 3, Section 3.3 est une formule de différentiation implicite non-lisse qui justifie cette pratique.

Échantillonnage du premier ordre non-lisse. L'échantillonnage du premier ordre peut être appliqué avec l'oracle de rétropropagation. Par exemple, dans l'apprentissage supervisé avec un prédicteur de réseau neuronal, nous pouvons considérer la méthode du gradient stochastique avec rétropropagation :

$$\begin{aligned} &\text{échantillon } \xi_k \sim P \\ &w_{k+1} = w_k - \alpha_k\, \text{backprop}_w\, f(w_k, \xi_k) \end{aligned}$$

Dans ce cas, il faut justifier une telle procédure. En particulier, il faut justifier si l'échantillonnage de $\text{backprop}_w(f(\cdot, \xi_k))$, qui est le résultat de la différentiation automatique non-lisse, approxime une direction de descente pour F en w_k.

Dans cette thèse, nous établissons un résultat dans le Chapitre 3, Section 3.2, qui permet l'interchangeabilité des opérations d'intégrale et de dérivation pour les fonctions non-lisses. Ce résultat vise à fournir une base théorique pour justifier l'échantillonnage du premier ordre dans le cadre non-lisse et non-convexe. Nous sommes motivés par l'ubiquité des environnements stochastiques dans les approches basées sur les données et une variété croissante de scénarios en ligne dans l'apprentissage automatique, y compris l'apprentissage par renforcement [76], l'apprentissage décentralisé et fédéré [82, 93]. D'autres environnements stochastiques généraux peuvent être l'utilisation de fonctions de perte moyennées sur une distribution absolument continue, comme dans l'inférence bayésienne où cette procédure permet d'incorporer de l'incertitude dans les prédictions du modèle [30].

En utilisant les procédures décrites ci-dessus, les méthodes du premier ordre trouvent leurs homologues non-lisses implémentables grâce à la différentiation automatique. De cette manière, la méthode du gradient et ses variantes vues dans la section 0.2.2 peuvent être transposées dans le cadre non-lisse.

0.3 Analyse des algorithmes du premier ordre

Dans cette thèse, nous cherchons à justifier la convergence des méthodes du premier ordre en apprentissage automatique du point de vue de l'optimisation. En particulier, nous souhaitons étudier les méthodes du premier ordre telles qu'elles sont mises en œuvre dans la pratique, par exemple la méthode du gradient stochastique avec rétropropagation (11) ou avec l'utilisation de la différentiation implicite (10). Nos principales questions seront les suivantes : les itérés convergent-ils vers des "points critiques", dans un certain sens ? La fonction objectif converge-t-elle ?

Comme nous le verrons, répondre à ces questions dans le cadre non-lisse nécessite l'utilisation d'outils spécifiques. Afin d'appréhender ces outils, il est important de comprendre que l'analyse dans le cadre non-lisse repose sur des mécanismes similaires à ceux utilisés dans le cadre lisse.

0.3.1 Analyse de Lyapunov dans le cadre lisse non-convexe

Dans le cadre lisse, une approche courante pour analyser les méthodes du premier ordre consiste à les considérer comme des approximations d'un système dynamique en temps continu. Cette approche, souvent appelée méthode des équations différentielles ordinaires (EDO), a été introduite pour la première fois dans des travaux antérieurs [98, 105] et a été largement explorée dans la littérature sur l'optimisation stochastique [15, 19, 27, 66, 77, 99]. Dans [19], l'auteur considère un processus stochastique général écrit comme

$$w_{k+1} = w_k + \alpha_k(H(w_k) + \epsilon_k)$$

avec des pas de tendant vers zéro $\alpha_k \to 0$ et un terme de bruit centré ϵ_k. Il montre que ce processus peut être étudié à travers les solutions de l'équation différentielle $\dot{w} = H(w)$ car il satisfait la propriété d'être une *pseudo-trajectoire asymptotique*.

Le terme $H(w_k) + \epsilon_k$ représente une mesure bruitée de $H(w_k)$. Par exemple, dans le cas de la méthode du gradient stochastique (9), $H(w_k)$ est le gradient du risque $\nabla F(w_k)$, et $H(w_k) + \epsilon_k$ est le gradient stochastique $\nabla_w f(w_k, \xi_k)$. Cette méthode peut alors être étudiée par l'intermédiaire du flot de gradient limite,

$$\dot{w} = -\nabla F(w). \tag{11}$$

Cela permet de justifier la convergence de la méthode du gradient stochastique vers l'ensemble critique $\{w \ : \ 0 = \nabla F(w)\}$. Cette méthode s'applique plus généralement aux algorithmes du premier ordre qui admettent une fonction de Lyapunov décroissante le long des trajectoires du système dynamique en temps continu. Dans le cas du flot de gradient, la fonction F agit comme une fonction de Lyapunov et décroît le long des courbes de gradient (11) en dehors de l'ensemble critique. La méthode de la boule pesante peut être étudiée comme dans [77] en considérant la fonction de Lyapunov $E(w, \dot{w}) = F(w) + \frac{1}{2}\|\dot{w}\|$. Des algorithmes adaptatifs ont également été étudiés en utilisant cette approche [15].

Propriété de Sard. Afin de mener une analyse de Lyapunov appropriée, il est courant de supposer que les valeurs critiques aient un intérieur vide [19, 66, 77]. Cette hypothèse peut être appelée propriété de Sard, d'après le théorème de Sard [146], qui affirme qu'un degré suffisant de différentiabilité de la fonction permet de satisfaire cette condition. Dans le cas de la méthode du gradient, par exemple, cette condition permet de montrer la convergence de la fonction objectif et que les points d'accumulation des itérés sont des points critiques.

Approche ergodique. Les algorithmes peuvent également être examinés du point de vue des mesures [19, 22]. En particulier, la trajectoire des itérations peut être représentée comme un continuum de mesures de Dirac, ce qui est appelé la *mesure d'occupation*. Les notions de convergence et de limite sont alors comprises comme dans l'espace des mesures. Par exemple, les points d'accumulation de l'algorithme sont associés à des mesures limites supportées sur les points stationnaires du flot.

Cette méthodologie permet la dérivation de résultats de convergence faible dans un cadre plus général. Dans [19], l'auteur considère des situations avec des pas plus lents, tandis que les auteurs de [27] traitent le cas des pas constants. Notamment, cette méthode permet la dérivation de résultats de convergence même en l'absence de la propriété de Sard.

Vers le cadre non-lisse. L'analyse des algorithmes du premier ordre non-lisses repose sur des principes similaires. Afin d'avoir une fonction de Lyapunov décroissante, des notions spécifiques de régularité pour les fonctions non-lisses peuvent être considérées et seront présentées dans la section 0.3.2. Dans la section 0.3.4, nous verrons que la méthode des équations différentielles ordinaires (ODE) et l'approche ergodique peuvent être adaptées aux algorithmes non-lisses en considérant des inclusions différentielles au lieu d'équations différentielles. Quant à la propriété de Sard, elle peut être satisfaite en considérant des fonctions semi-algébriques ou définissables, souvent rencontrées dans des cas pratiques, où les points de non-différentiabilité sont organisés en variétés lisses (section 0.3.3).

0.3.2 Régularité des fonctions non-lisses

Bien avant l'émergence des problèmes modernes d'apprentissage automatique, tels que ceux que nous avons mis en évidence dans la section 0.1, les problèmes non-lisses et non-convexes avaient déjà suscité l'intérêt de la communauté de l'optimisation. En ce qui concerne les fonctions convexes, on sait que la notion de gradient peut être étendue aux fonctions non-lisses $f : \mathbb{R}^p \to \mathbb{R}$ en considérant l'application à valeurs multiples ∂f qui satisfait pour tout x, pour tout $v \in \partial f(x)$,

$$f(z) \geq f(x) + \langle v, z - x \rangle \text{ pour tout } z \in \mathbb{R}^p. \tag{12}$$

Cette définition a été introduite par Rockafellar [135,136] et également par Moreau dans [117], où il appelle ∂f le sous-gradient de f. Dans le cas convexe, une observation que l'on peut faire à partir de l'inégalité (12) est que l'application ∂f possède une interprétation variationnelle inhérente et rend compte des variations locales de la fonction. Elle peut également être vue comme une approximation du premier ordre unilatérale.

Motivé par la minimisation de fonctions de valeur maximale, le concept de sous-gradient a ensuite été étendu aux fonctions localement lipschitziennes non-lisses par Clarke [55]. Il le définit comme la fermeture convexe-graphique de l'ensemble des gradients. Plus précisément, pour une fonction localement lipschitzienne $f : \mathbb{R}^p \to \mathbb{R}$, le sous-gradient de Clarke est donné pour tout $x \in \mathbb{R}^p$ par

$$\partial^c f(x) = \text{conv}\{v \in \mathbb{R}^p \ : \ v = \lim_{k\to\infty} \nabla f(x_k), x_k \underset{k\to\infty}{\to} x, \{x_k\}_{k\in\mathbb{N}} \subset \text{diff}_f\},$$

où diff_f est l'ensemble de différentiabilité de f. Cependant, bien que cet oracle soit bien défini pour toute fonction localement lipschitzienne, il manque généralement d'information variationnelle.

En effet, certaines fonctions lipschitziennes ont un sous-gradient de Clarke maximal [45, 135] égal à la boule unité partout, rendant ainsi impossible l'accès à des directions de descente. Un exemple tiré de [135] est le suivant : soit $A \subset \mathbb{R}$, tel que A et A^c sont denses, et pour tout intervalle ouvert I, ni $A \cap I$ ni son complémentaire dans I n'ont une mesure de Lebesgue nulle. Ensuite, considérons la fonction

$$f : x \to \int_0^x 2\mathbb{1}_A(s) - 1 \, \mathrm{d}s. \tag{13}$$

La dérivée de Clarke de f est égale à $[-1, 1]$ partout. En l'absence d'information variationnelle dans le cadre localement lipschitzien général, une notion supplémentaire de régularité est nécessaire pour donner un sens aux algorithmes du premier ordre non-lisses.

Fonctions semi-lisses. Mifflin [113] a introduit une classe de fonctions non-lisses appelées *semi-lisses*, qui présentent des propriétés variationnelles favorables par rapport au sous-gradient de Clarke. Les fonctions semi-lisses $f : \mathbb{R}^p \to \mathbb{R}$ satisfont, en chaque point $x \in \mathbb{R}^p$, pour y dans un voisinage de x,

$$f(y) = f(x) + \langle v, y - x\rangle + o(\|x - y\|), \quad \text{lorsque } y \to x, \ \text{ pour tout } v \in \partial^c f(y).$$

Des exemples de fonctions semi-lisses sont les fonctions convexes et le maximum ponctuel d'une famille compacte de fonctions lisses. Ces fonctions sont également stables par composition. Mifflin a proposé un algorithme [112] pour un objectif semi-lisse qui converge vers les points stationnaires d'un problème contraint. Les fonctions semi-lisses ont également été explorées dans le contexte des méthodes de Newton dans [130].

Fonctions chemins-différentiables. En lien avec un lemme de règle de la chaîne pour les fonctions convexes de Brézis [48, Lemma 3.3], Valadier [155] a introduit la notion de "fonctions non pathologiques" dans le cadre du processus de balayage de Moreau [118]. Ce sont des fonctions localement lipschitziennes f satisfaisant une règle de la chaîne le long de courbes absolument continues $\gamma : [0, 1] \to \mathbb{R}^p$: pour presque tout $t \in [0, 1]$,

$$\frac{\mathrm{d}(f \circ \gamma)}{\mathrm{d}t}(t) = \langle \partial^c f(\gamma(t)), \dot{\gamma}(t)\rangle. \tag{14}$$

Cette propriété de règle de la chaîne semble favorable pour l'optimisation car elle implique que la fonction f décroît le long des courbes du flot du sous-gradient $\dot{\gamma} \in -\partial^c f(\gamma)$. Elle a récemment été redécouverte en optimisation non-lisse dans plusieurs travaux, d'abord [63] qui l'appelle une propriété de "règle de la chaîne" puis [37, 41] qui utilisent la terminologie de fonctions "chemins-différentiables".

Liens entre les deux notions. Bien que les deux notions aient été introduites dans des contextes différents, plusieurs travaux ont mis en évidence leurs liens. Borwein et Moors [44] ont montré que les fonctions semi-lisses satisfont la propriété d'être essentiellement lisses, donc satisfaisant une règle de la chaîne plus faible. Ruszczyński [142] a montré une règle de la chaîne le long de courbes semi-lisses pour les fonctions semi-lisses. Dans cette thèse, dans la section 3.4 du chapitre 3, nous établissons que la semi-lissité implique la chemin-différentiabilité, mais que la réciproque est fausse en général.

Des fonctions simples rencontrées en apprentissage automatique telles que les fonctions semi-algébriques satisfont en réalité les deux notions [33,63] et des fonctions pathologiques telles que (13) sont rarement rencontrées. Un cadre excluant les cas pathologiques peut être considéré de manière rigoureuse avec les *structures o-minimales*.

0.3.3 Structures o-minimales : un cadre favorable pour l'optimisation

Les fonctions définissables sont semi-lisses et chemins-différentiables. D'un point de vue topologique, les fonctions présentant un sous-gradient de Clarke égal à la boule unité, donc ni semi-lisses ni chemins-différentiables, sont prédominantes dans l'espace des fonctions lipschitziennes [45]. Cependant, il est difficile de concevoir que des fonctions simples rencontrées en apprentissage automatique, par exemple les fonctions semi-algébriques, puissent présenter un aspect pathologique de ce type. En réalité, les ensembles de non-différentiabilité de ces fonctions sont plutôt bien structurés. Dans le cas de la norme ℓ_1 ou des réseaux ReLU, par exemple, les ensembles de non-différentiabilité sont organisés en sous-espaces affines.

On peut imaginer un phénomène similaire lorsqu'on considère des compositions impliquant d'autres fonctions courantes telles que l'exponentielle ou le logarithme. En fait, les fonctions semi-algébriques peuvent être généralisées à des dictionnaires plus larges de fonctions élémentaires, incluant par exemple l'exponentielle, avec les structures o-minimales [59, 156]. En considérant des fonctions appartenant à de telles structures, appelées *définissables* ou *modérées*, on peut exclure formellement les cas pathologiques tels que la fonction (13).

Dans le cas des fonctions semi-algébriques ou définissables non-lisses, les ensembles de non-différentiabilité sont bien structurés et organisés en variétés lisses. Cela peut être formulé avec la propriété de *stratification de Whitney* [160]. Cette propriété conduit à une formule de projection [40] pour le sous-gradient de Clarke. Sur la base de ce résultat, [33] a montré que les fonctions définissables et localement lipschitziennes sont semi-lisses, et plus tard dans [63], que ces fonctions sont également chemins-différentiables.

Théorème de Sard définissable. Une conséquence importante du cadre définissable est qu'il exclut les situations pathologiques où la propriété de Sard n'est pas vérifiée. En effet, les fonctions définissables satisfont la propriété de Sard [39]. Comme nous l'avons mentionné dans la section 0.3.1, cette condition joue un rôle fondamental dans le développement de résultats de convergence, mais elle apparaît souvent comme une hypothèse abstraite [21, 66, 72, 77, 142]. Le cadre définissable permet de l'obtenir de manière plus raisonnable que le théorème de Sard classique, qui requiert un degré de différentiabilité au moins égal à la dimension du paramètre.

En dehors du cadre définissable, de nombreux comportements pathologiques peuvent se produire. Un contre-exemple au théorème de Sard a été donné pour la première fois par Whitney [161] avec une construction fractale. Un autre contre-exemple exhibant des séquences pathologiques de sous-gradient a été proposé dans [133].

Les structures o-minimales et leurs conséquences en l'optimisation seront présentées en détail

dans le chapitre 2. Des exemples issus de l'apprentissage automatique seront également exposés dans le chapitre 2, section 2.2.3.

0.3.4 Méthode d'inclusion différentielle

La méthode des EDO présentée dans la section 0.3.1 a été étendue aux dynamiques multivaluées dans [21]. Les auteurs de [21] adaptent la notion de pseudo-trajectoires asymptotiques aux récursions multivaluées $w_{k+1} \in w_k + \alpha_k(H(w_k) + \epsilon)$ où H est maintenant une application à valeurs compactes, convexes et ayant un graphe fermé. De manière similaire à la méthode des EDO lisses, la récursion discrète est vue comme une approximation de l'inclusion différentielle $\dot{w} \in H(w)$.

Ce cadre englobe les méthodes du premier ordre non-lisses. Par exemple, il permet d'étudier une méthode de sous-gradient stochastique

$$w_{k+1} \in w_k - \alpha_k(\partial^c f(w_k) + \epsilon_k), \tag{15}$$

comme une approximation stochastique discrète du flot de sous-gradient $\dot{\gamma} \in -\partial^c f(\gamma)$. Il a été utilisé, par exemple, pour étudier la méthode de sous-gradient stochastique (15) dans [63, 108], sa version de boule pesante [142] et une méthode inertielle [51].

L'approche par mesure fermée. L'approche ergodique [19, 22] a été étendue au cadre multivalué [29, 75] et explorée également dans le cas de la méthode du sous-gradient [41]. Les auteurs de [29, 41] proposent une interprétation algorithmique de la convergence faible en termes de *points d'accumulation essentiels*. Comme dans le cas lisse, l'un des principaux avantages de cette approche est de fournir des résultats de convergence dans un cadre plus général, par exemple sans la propriété de Sard, permettant d'obtenir des résultats de convergence au-delà du cas définissable. De plus, le cadre proposé dans [29, 41] permet d'étudier plus précisément le comportement de l'algorithme, ce qui permet de capturer des oscillations qui peuvent être négligées dans les analyses de convergence traditionnelles.

0.3.5 Un résumé illustratif : comment analyser la méthode du sous-gradient ?

En regroupant les notions précédentes, nous résumons la méthodologie typique pour analyser la méthode du sous-gradient. Cette approche est canonique et peut être appliquée à d'autres algorithmes admettant un système de Lyapunov. Certains détails techniques sont en effet omis dans cette introduction et un résumé complet de l'analyse sera présenté dans le chapitre 4. Considérons une fonction F à optimiser et une méthode de sous-gradient stochastique

$$w_{k+1} \in w_k - \alpha_k(\partial^c F(w_k) + \epsilon_k),$$

avec des pas évanescents α_k et où ϵ_k est un bruit conditionnel centré. Si nous supposons que F est semi-algébrique ou définissable, F est différentiable le long des courbes du flot de sous-gradient,

$$\dot{w} \in -\partial^c F(w),$$

ce qui fait de F une fonction de Lyapunov. Dans ce cas, nous pouvons nous appuyer sur la méthode d'inclusion différentielle (section 0.3.4) afin d'étudier la méthode du sous-gradient stochastique. La propriété de Sard, obtenue grâce au cadre définissable, permet de faire une analyse de Lyapunov précise et d'obtenir la convergence de la fonction objectif, ainsi que la criticité de tous les points d'accumulation w^*, c'est-à-dire $0 \in \partial^c F(w^*)$. En dehors de cette condition, l'approche par mesure fermée permet d'obtenir des résultats de convergence plus faibles.

La chemin-différentiabilité a été exploitée dans [37, 63] pour démontrer la convergence de la méthode du sous-gradient (15). Une analyse de la méthode du sous-gradient déterministe pour les fonctions chemins-différentiables a été réalisée dans [41] avec une étude des oscillations grâce à l'approche par mesure fermée.

Sur l'analyse de convergence pour les fonctions semi-lisses. La semi-lissité permet d'obtenir un lemme de descente asymptotique dans [72], conduisant aux mêmes résultats de convergence que la méthode des EDO pour l'algorithme du sous-gradient stochastique. Cependant, la méthode des EDO (non-lisse) est souvent privilégiée dans la littérature en raison de sa polyvalence. Par exemple, plusieurs travaux qui considèrent des fonctions semi-lisses [84,142,143] utilisent l'approche des EDO à travers une forme faible de chemin-différentiabilité.

0.3.6 Limites de la théorie non-lisse.

La théorie présentée ci-dessus présente plusieurs limitations lorsqu'on considère des problèmes pratiques d'apprentissage automatique.

Une première lacune : la géométrie modérée et l'optimisation stochastique. Comme souligné au début de cette introduction, les problèmes d'apprentissage automatique impliquent la minimisation d'une espérance générale (1). Alors que l'intégrande f peut raisonnablement être considéré semi-algébrique ou définissable, cette propriété n'est pas préservée lorsqu'on applique l'espérance. Certaines considérations techniques ont donc été faites dans les travaux mentionnés précédemment. Par exemple, pour obtenir la chemin-différentiabilité du risque, [63] considère une distribution discrète et finie. La propriété de Sard est également obtenue avec cette configuration dans [37,63] grâce à la stabilité des fonctions semi-algébriques et définissables sous une somme finie.

Une partie de cette thèse vise à aller au-delà de ces considérations. En particulier, nous prouvons la chemin-différentiabilité des fonctions de risque générales grâce à une règle de différentiation sous intégrale dans le chapitre 3, section 3.2. Cela permet d'appliquer la méthode des EDO, en particulier l'approche par mesure fermée, pour obtenir des résultats de convergence en dehors de la propriété de Sard. Pour cette dernière, nous proposons de l'obtenir dans le cas d'un large éventail de distributions absolument continues en utilisant un résultat sur l'intégration des fonctions définissables [56], qui sera présenté dans le chapitre 2, section 2.2.5. Ce résultat n'a pas été introduit dans la littérature sur l'optimisation stochastique jusqu'à présent, mais il permet de justifier la propriété de Sard en dehors du cas de la minimisation du risque empirique [37, 63].

Une deuxième lacune : le sous-différentiel de Clarke et le calcul. Alors que la théorie de l'optimisation non-lisse repose sur le sous-différentiel de Clarke, cet oracle n'est pas vraiment utilisé en apprentissage automatique. Les modèles d'apprentissage automatique sont complexes, en particulier en apprentissage profond, et leur entraînement repose fortement sur l'application de formules de calcul pour construire des oracles du premier ordre.

Comme nous l'avons vu dans la section 0.2.3, la rétropropagation est l'application de la règle de la chaîne du calcul différentiel aux compositions de fonctions non-lisses. La différentiation implicite non-lisse est utilisée pour la différentiation des couches implicites ou des solutions de problèmes d'optimisation. Nous rappelons également que l'échantillonnage du premier ordre, par exemple la méthode du gradient stochastique, est justifié dans le cas lisse par l'interversion de l'espérance et du gradient $\mathbb{E}_{\xi\sim P}[\nabla_w f(\cdot,\xi)] = \nabla F$. Dans le cas non-lisse, l'échantillonnage du premier ordre est utilisé avec la rétropropagation en supposant qu'une direction de descente est approximée et qu'une version non-lisse de la règle d'interversion est valide.

Malheureusement, en général, l'application de ces règles dans le cadre de la théorie du sous-différentiel de Clarke reste heuristique, et les règles de calcul conventionnelles ne sont pas valides avec les sous-différentiels de Clarke. Par exemple, la règle de la chaîne ne s'applique pas. En ce qui concerne la différentiation implicite, un théorème existe pour la régularité et l'existence de la fonction implicite dans le cas d'une équation Lipschitz non-lisse [54], mais il ne s'accompagne pas d'une formule de différentiation implicite non-lisse.

En ce qui concerne l'échantillonnage du premier ordre, il n'existe pas de règle d'interversion pour l'intégrale et le sous-gradient de Clarke. Plusieurs travaux [63,108] considèrent la méthode du sous-gradient stochastique écrite sous la forme (15), alors qu'elle ne représente pas la pratique. En plus de l'utilisation de la rétropropagation, la méthode du sous-gradient stochastique, qui échantillonne des sous-gradients stochastiques, s'écrit

$$w_{k+1} \in w_k - \alpha_k \partial_w^c f(w_k, \xi_k),$$

où $\partial_w^c f$ est le sous-différentiel de Clarke par rapport à la première variable. En général,

$$\partial^c F \subset \mathbb{E}_{\xi \sim P}[\partial_w^c f(w_k, \xi)],$$

mais l'égalité n'est pas vraie.

0.4 Une vision sans opérateur du calcul non-lisse

Avant l'introduction des réseaux neuronaux profonds et de la différenciation automatique, le besoin d'un calcul non-lisse était déjà exprimé dans la littérature sur l'optimisation convexe. En effet, il n'est pas toujours possible d'obtenir le sous-gradient d'une somme en additionnant les sous-gradients, et il n'existe pas d'analogue général de la règle de la chaîne pour calculer les sous-dérivées de compositions. Par conséquent, des conditions ont été établies dans [54, 136] pour assurer la validité de règles de calcul usuelles telles que la somme ou la composition, nécessitant la convexité, la régularité de Clarke, ou des conditions de qualification dans le cas de problèmes contraints.

En ce qui concerne les problèmes stochastiques, l'interchangeabilité $\partial F = \mathbb{E}_{\xi \sim P}[\partial_w f(\cdot, \xi)]$ est vérifiée lorsque f est convexe par rapport à son premier argument. Cette règle est fondamentale dans les problèmes stochastiques. Par exemple, elle justifie l'échantillonnage du sous-gradient pour concevoir des algorithmes de premier ordre en ligne, la consistence statistique des problèmes stochastiques, avec une loi des grands nombres, et des applications à l'optimisation sous incertitude et aux inégalités variationnelles stochastiques, voir par exemple [148].

Comme cela a été souligné dans la partie précédente, le sous-différentiel de Clarke ne permet pas un calcul dans le cadre général non-lisse et non-convexe. Deux modèles variationnels [37, 120] ont été proposés pour étendre les règles de calcul aux fonctions non-lisses non-convexes. Dans ces modèles, les dérivées d'une fonction sont des applications à valeurs dans des ensembles, considérées comme des entités non uniques :

Les dérivées semi-lisses de Norkin. La notion de gradients semi-lisses généralisés a été proposée par Norkin [120]. Ce sont des applications à valeurs dans des ensembles, dont le graphe est fermé, et localement bornées, que l'on note D_f. Elles satisfont la propriété de semi-lissité par rapport à f : pour tout $x \in \mathbb{R}^p$,

$$f(y) = f(x) + \langle v, y - x \rangle + o(\|x - y\|) \text{ lorsque } y \to x, \text{ pour tout } v \in D_f(y).$$

Les dérivées conservatives Bolte et Pauwels [37] ont proposé la notion de gradients conservatifs, qui sont des applications à valeurs dans des ensembles D_f satisfaisant la règle de la chaîne le long de courbes absolument continues $\gamma : \mathbb{R}_+ \to \mathbb{R}^p$: pour presque tout $t \geq 0$,

$$\frac{\mathrm{d}(f \circ \gamma)}{\mathrm{d}t}(t) = \langle v, \dot{\gamma}(t) \rangle \text{ pour tout } v \in D_f(\gamma(t)).$$

Comme on peut le remarquer, ces deux modèles sont basés sur les propriétés de semi-lissité et de chemin-différentiabilité vues en section 0.3.2. Ces modèles peuvent être considérés comme "sans opérateur" car de nombreuses dérivées différentes peuvent exister pour une même fonction. Auparavant, la construction de l'oracle du premier ordre, dans le cadre lisse ou dans le cas du sous-différentiel de Clarke, était basée sur une formule impliquant la fonction, ce qui conduisait à un oracle unique. Dans ces nouveaux modèles, un oracle du premier ordre est défini par sa propriété à rendre compte des variations de la fonction.

Ces modèles justifient la plupart des règles de calcul simples. Par exemple, pour deux fonctions semi-algébriques f et g, la somme des sous-différentiels $\partial^c f + \partial^c g$ peut ne pas être égale au sous-différentiel de $f+g$, mais c'est un gradient conservatif pour $f+g$. Un mécanisme similaire est obtenu lorsque l'on utilise la règle de chaîne sur les Jacobiens de Clarke : en appliquant la règle de chaîne à une composition $F \circ G$ avec les Jacobiens de Clarke $\mathrm{Jac}^c F$ et $\mathrm{Jac}^c G$, on obtient l'application à valeurs dans des ensembles $\mathrm{Jac}^c F(G)\, \mathrm{Jac}^c G$, qui est un Jacobien conservatif pour $F \circ G$. Cela fournit un modèle variationnel pour la différentiation automatique non-lisse. Différentes versions de la différentiation automatique utilisées en pratique, comme les modes avant et arrière [36, 37], peuvent être modélisées de cette manière.

Les mêmes règles s'appliquent aux dérivées semi-lisses : par exemple, la somme de deux gradients semi-lisses donne un gradient semi-lisse pour la somme. Une règle d'intégrale est valable pour les dérivées semi-lisses [114, 121], en particulier si $D(\cdot, \xi)$ est un gradient semi-lisse pour $f(\cdot, \xi)$ presque sûrement pour $\xi \sim P$, alors en prenant l'espérance $\mathbb{E}_{\xi \sim P}[D(\cdot, \xi)]$, on obtient un gradient semi-lisse pour $\mathbb{E}_{\xi \sim P}[f(\cdot, \xi)]$, justifiant ainsi l'échantillonnage du premier ordre pour les gradients semi-lisses.

L'avantage de ces approches sans opérateur est qu'elles permettent une justification fidèle et simple des règles de calcul. Afin de justifier la différentiation automatique en accord avec le modèle du sous-différentiel de Clarke, un modèle a été proposé dans [90]. Cependant, cette approche nécessite de satisfaire des conditions de qualification et des conventions d'implémentation qui peuvent ne pas refléter la pratique courante.

Il a été démontré dans le cas précis des fonctions semi-algébriques ou définissables que les deux notions sont équivalentes [62]. Dans le chapitre 3, section 3.4, nous étudions les différences entre les deux notions dans le cas général et montrons que les gradients semi-lisses sont des gradients conservatifs mais que la réciproque est fausse.

Dans cette thèse, nous nous concentrons sur le modèle des dérivées conservatives que nous présentons dans le chapitre 3 et proposons deux extensions. En section 3.2, une première extension du calcul conservatif sera une règle d'intégrale, justifiant l'échantillonnage du premier ordre dans le cas chemin-différentiable. Une deuxième extension, en section 3.3, sera une formule de différentiation implicite non-lisse avec des applications aux modèles implicites et à la programmation bi-niveau en section 3.3.3.

Les gradients conservatifs trouvent d'autres applications, justifiant par exemple l'optimalité paramétrique des fonctions de valeur [127], la différentiabilité le long des flots d'EDO [110], la différentiation des algorithmes itératifs [38] et de solutions paramétrées d'inclusions monotones [42].

Convergence des algorithmes non-lisses implémentés en pratique. Grâce à leur définition, les gradients conservatifs permettent une théorie de convergence des algorithmes du premier ordre tels qu'ils sont implémentés en pratique. Par exemple, supposons que nous minimisons une composition $F := F_1 \circ \ldots \circ F_r$. L'application de la rétropropagation sur F correspond à l'application de la règle de la chaîne aux dérivées de Clarke et donne un gradient conservatif D_F. La méthode du sous-gradient avec rétropropagation peut alors être écrite sous la forme d'une récursion multivaluée :

$$w_{k+1} \in w_k - \alpha_k D_F(w_k),$$

où D_F est un gradient conservatif pour F. Grâce à la méthode d'inclusion différentielle (section 0.3.4), cette récursion peut être étudiée à travers le flot de gradient conservatif $\dot{w} \in -D_F(w)$. Par la définition du gradient conservatif, F diminue le long des courbes de ce flot, conduisant ainsi à la convergence vers des points critiques $\{w : 0 \in D_F(w)\}$. En d'autres termes, l'analyse en section 0.3.4 peut être répétée en remplaçant $\partial^c F$ par D_F.

La méthode du sous-gradient stochastique avec rétropropagation et distribution finie est étudiée dans [37]. Pour un risque empirique $F := \frac{1}{n}\sum_{i=1}^n f_i$, la sortie de la rétropropagation appliquée à une fonction f_i appartient à un gradient conservatif D_i pour f_i. Si maintenant nous échantillonnons de manière uniforme à partir des sorties de la rétropropagation des fonctions $(f_i)_{i=1,\ldots,n}$, cela approximera un élément de $D_F := \frac{1}{n}\sum_{i=1}^n D_i$, qui est un gradient conservatif pour F. La méthode du sous-gradient stochastique avec rétropropagation peut alors être écrite sous la forme

$$w_{k+1} \in w_k - \alpha_k (D_F(w_k) + \epsilon_k),$$

avec un bruit ϵ_k centré conditionnellement à w_k, et elle peut être étudiée à l'aide de la méthode d'inclusion différentielle.

Dans cette thèse, nous étendons cette analyse à une distribution générale en utilisant une règle d'intégrale conservative que nous montrons en chapitre 3, section 3.2. En conséquence, nous proposons un cadre général pour analyser les méthodes stochastiques du premier ordre telles qu'elles sont implémentées en pratique (section 4.2). Nous appliquons la méthode d'inclusion différentielle pour étudier la méthode du sous-gradient stochastique et sa version boule pesante.

Vers des garanties indépendantes du calcul. L'utilisation de ce type d'oracles "opérateur-indépendant" conduit à la convergence vers des points critiques généralisés $\{\bar{w} : 0 \in D_F(\bar{w})\}$. Une telle convergence suggère que l'utilisation de règles de calcul génère des artefacts qui peuvent avoir un impact sur la convergence de la méthode. Par exemple, dans le cas de la différentiation automatique, les auteurs de [36] ont souligné que la non-unicité des implémentations pourrait conduire à des points critiques artificiels ayant des emplacements absurdes. Par conséquent, on peut chercher à certifier que les règles de calcul, utilisées pour définir un gradient conservatif D_F, n'ont pas d'impact sur la convergence, ce qui peut être exprimé par la convergence vers les points critiques de Clarke $\{w : 0 \in \partial^c F(w)\}$.

Les gradients à la fois semi-lisses et conservatifs ont de bonnes propriétés variationnelles, car ce sont des gradients presque partout, voir [142, Appendice A] ou [114] pour les dérivées semilisses et [37, Théorème 1] pour les gradients conservatifs. Une solution simple proposée dans [121] pour retrouver la convergence vers les points critiques de Clarke consiste à ajouter un bruit uniforme à chaque étape.

Un autre axe de recherche, initié par [28, 36], vise à certifier cette convergence sans modifier la méthode. Il est démontré dans [36] que lorsque la fonction objectif est une somme finie et définissable, la randomisation de l'initialisation et l'évitement d'un ensemble fini de tailles de pas suffisent à éviter tous les artefacts. Cette question a également été étudiée dans [28] dans le cas

de la méthode du sous-gradient stochastique avec une taille de pas constante. Les auteurs de [28] ont montré pour des échantillons provenant d'un espace de probabilité général, que l'évitement se produit pour une initialisation aléatoire et chaque fois que les tailles de pas évitent un ensemble de mesure de Lebesgue nulle.

Dans cette thèse, chapitre 4, section 4.3, nous étudions cette question dans le cas de la méthode du sous-gradient stochastique et de sa version avec boule pesante. Pour la méthode du sous-gradient stochastique, nous considérons le cas d'une distribution absolument continue et déduisons une caractérisation de l'ensemble d'initialisation plus précise que dans [28].

0.5 Structure de la thèse et contributions

Dans le chapitre 2, nous rappelons les notions d'analyse multivaluée. En effet, les applications à valeurs multiples sont récurrentes dans ce travail, et les notions d'intégration et de mesurabilité doivent être définies pour de tels objets. Dans le même chapitre, nous présentons les structures o-minimales qui sont centrales dans cette thèse. Certains résultats sont démontrés et peuvent être d'intérêt indépendant, tels qu'un résultat d'intégration pour les applications à valeurs multiples définissables et un lemme de type Fubini pour les ensembles définissables denses.

En vue d'étudier des algorithmes avec des méthodes d'EDO [21, 29], nous nous intéressons au modèle des gradients conservatifs que nous présentons dans le chapitre 3. Nous proposons ensuite deux extensions du calcul conservatif : une formule de différentiation implicite non-lisse et une règle d'intégrale non-lisse. Nous appliquons notre formule de différentiation implicite non-lisse afin de justifier des méthodes d'optimisation du premier ordre pour les problèmes bi-niveaux, par exemple l'optimisation des hyperparamètres et les modèles implicites. La motivation de la règle d'intégrale est double. Premièrement, elle permet d'obtenir la chemin-différentiabilité pour une fonction intégrale générale, justifiant ainsi une propriété de descente pour la minimisation du risque en dehors du cas semi-algébrique. Deuxièmement, elle justifie un échantillonnage non-lisse du premier ordre dans les implémentations pratiques qui peuvent utiliser la différentiation automatique.

À la fin du chapitre 3, nous étudions la relation entre les dérivées semi-lisses et les dérivées conservatives, et établissons que les dérivées semi-lisses sont des dérivées conservatives en général. Des exemples comparatifs en dimension une sont fournis.

Nous utilisons le calcul développé dans le chapitre 3 dans le chapitre 4 afin de proposer un cadre général de stochasticité non-lisse compatible avec le calcul, par exemple la différentiation automatique et la différentiation implicite non-lisse. Une caractéristique notable de notre cadre est de s'appuyer sur un résultat d'intégration des fonctions définissables afin d'obtenir une propriété de Sard pour une large famille de distributions absolument continues. Basé sur la chemin-différentiabilité, notre cadre permet d'utiliser les méthodes d'inclusion différentielle [21,29] pour obtenir des résultats de convergence pour la méthode du sous-gradient stochastique et sa version avec balle lourde.

À la fin du chapitre 4, nous étudions la question de l'évitement des artefacts. Dans le cas de la méthode du sous-gradient stochastique, nous nous concentrons sur le cas d'une distribution absolument continue. Nous étudions la question pour la méthode de la boule pesante stochastique et complétons les analyses précédentes [29, 142] de cet algorithme.

Références. Cette thèse est basée sur les articles suivants :

- [35] Jérôme Bolte, Tam Le, Edouard Pauwels, Antonio Silveti-Falls, Nonsmooth implicit differentiation for machine learning and optimization, Neurips (2021).

- [34] Jérôme Bolte, Tam Le, Edouard Pauwels, Subgradient sampling for nonsmooth nonconvex minimization, SIAM Journal on Optimization (2023).
- [100] Tam Le. Nonsmooth nonconvex stochastic heavy ball, soumis (2023).