Formal Verification meets Stochastic Analysis



Francesco Cosentino
Exeter College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Mathematical Institute

Trinity 2021

Acknowledgements

It was September 2017 when I first arrived in London, and almost four years later I am submitting my DPhil thesis, the last step to finish this incredible adventure.

First, I would like to recognise my deep thankfulness to Harald and Alessandro for their support and guidance. I will miss the pleasant atmosphere of our lunches together where most of the brainstorming happened. In particular, I would like to thank Harald for his perseverance and for sharing with me his knowledge, whilst Alessandro for trusting and encouraging my autonomous ideas. I will not forget their patience and dedication during my DPhil, and their availability to always share their experience. Moreover, I want to thank also my transfer, confirmation and viva examiners Prof Terry Lyons, Prof Stefan Kiefer, Dr Jan-Peter Calliess, Prof Samuel Cohen and Prof Christian Bayer for their detailed comments and recommendations.

Second, I would like to thank the Oxford Control and Verification group within the Department of Computer Science. A special mention goes to my "Oxford train mate" Andrea Peruffo, my "office mates" Hosein Hasanbeig and Mehran Hosseini, and Gareth Molyneux, Nathalie Cauchi, and Viraj Wijesuriya, for the kindness and guidance received during these years.

Third, I thank The Alan Turing Institute and the University of Oxford for their support. A distinct acknowledgement goes to Andrea Pizzoferrato, without him my journey would have been much more challenging than what it has been.

A further thought belongs to those who are not with us anymore.

Finally, the most significant credit is for my family Alfonso, Cristina and Lina and my partner Veronica, who has shared with me this adventure in all its facets, for always supporting and bearing me.

Abstract

The thesis goal is to explore the relations between Formal Verification techniques in Computer Science and Stochastic Analysis in Mathematics. They both deal with probabilistic dynamical systems, implying that the number of connections is significant. The Formal Verification studies dynamical systems' properties, which are specified in terms of logic statements. This discipline has been successfully applied to multiple industry problems, such as "verification of" Software, Microprocessors or Neural Networks. The models studied have a countable states space and this can be considered the main obstacle to extend Formal Verification techniques for the computation of likelihoods in the case of Stochastic Differential Equations. However, the most studied properties in Formal Verification have a clear characterization in Mathematics. In this thesis, we focus on the safety property "always", which can be characterized as an exit time in the theory of Stochastic Analysis.

The main difficulties are twofold. Firstly, if we are particularly interested in discretizing the states space - going from a continuous, uncountable space to a finite countable one - it is very likely to have an explosion of the cardinality of the states space. This leads to problems both in time and memory complexity. We find a new characterization of the barycenter of a discrete probability measure, which allows us to introduce a new algorithm for the recombination problem. Then, we use the algorithm to build recombining trees in order to approximate weakly stochastic processes, solving the exponential explosion of the cardinality of the states space. Secondly, one has to consider the approximation error: if the probability to exit from a safety region has not an analytical closed-form, a numerical scheme must be considered and therefore it is important to analyse the error. We introduce a grid-free approach for the computation of Probabilistic Safety Regions using Malliavin Calculus.

Thesis Outline

A visual structure of the dissertation is given in Figure 1.

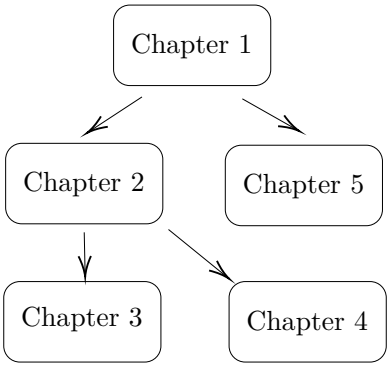


Figure 1: Structure of the thesis.

In Chapter 1, we present a short introduction to Formal Verification methods and the related literature we have examined for the material developed. It shows our initial point of view, the influence of exit times for the Formal Verification community, and the relevance of knowing exactly the approximation error, due to safety applications. Moreover, it introduces Probabilistic Model Checkers, which are the principal reason why the Formal Verification community is interested in discrete-space Markov chains.

In Chapter 2, we study a seemingly unrelated problem, namely the recombination problem. The recombination problem consists of reducing the support of discrete measures matching the expectation of some chosen functions. We find a new geometric characterization of the problem and apply it to build a new greedy randomized algorithm. Moreover, the geometric characterization allows us to state a new robustness result. Comparisons with other algorithms from the literature are given. This Chapter is based on Cosentino et al. [40]. In Chapter 4, we link the recombination problem with the numerical approximation of SDEs.

In Chapter 3, we give a new application of the recombination problem to optimization

theory. We present convergence results of the algorithms introduced and we prove the rate of convergence for a particular choice of the update directions under convexity assumptions. Comparisons with algorithms from the literature are given and we focus on how to apply the new technique to Block Coordinate Descent Algorithms. This Chapter is based on Cosentino et al. [41].

In Chapters 4 and 5, we present two applications. In Chapter 4, we discuss an application of the recombination problem to the construction of discrete space Markov chains, which avoids the exponential explosion of the number of states with respect to the time discretization. In Chapter 5, we study an application of Malliavin Calculus to the computation of "Probabilistic Safety Regions" in Formal Verification: given a region A, we want to find all the states x such that, starting an SDE from x, the probability of leaving A by a certain amount of time is greater than a chosen probability.

Publications. Part of the material presented in this doctoral thesis has appeared in top-tier international conference proceedings, or has been submitted in peer-reviewed journals:

- Chapter 2 has been accepted at NeurIPS 2020 as Spotlight Presentation, Cosentino et al. [40].
- Chapter 3 is based on Cosentino et al. [41], which has been submitted to a journal.
- Chapter 4 is based on Cosentino et al. [43], which has been submitted to a journal.
- Chapter 5 is based on Cosentino et al. [42], which has been submitted to a journal.

Contents

1 Formal Verification Introduction **1**
1.1 Informal Introduction . . . 1
1.2 Literature review . . . 4
1.2.1 Formal Verification, Discrete-time models . . . 4
1.2.2 Formal Verification, Continuous-time models . . . 6
1.2.3 Mathematical Literature . . . 8

2 A Randomized Algorithm to Reduce the Support of Discrete Measures **15**
2.1 Introduction . . . 15
2.2 Negative cones and a naive algorithm . . . 17
2.3 A geometrically greedy Algorithm . . . 26
2.4 Experiments . . . 33
2.5 Summary . . . 38
2.A Appendix - Divide and conquer, choice of the subgroup size . . . 39
2.B Appendix - A hybrid algorithm . . . 40

3 Carathéodory Sampling for Stochastic Gradient Descent **43**
3.1 Introduction . . . 43
3.2 Carathéodory Gradient Descent (CaGD) . . . 46
3.2.1 A first comparison of CaGD and GD: logistic regression . . . 52
3.3 Carathéodory Block Coordinate Descent . . . 54
3.3.1 From Block Coordinate Descent to Carathéodory Block Coordinate Descent . . . 54
3.3.2 Update rules . . . 55
3.3.3 Experiments: CaBCD vs ADAM vs SAG for LASSO . . . 57
3.3.4 Discussion of results . . . 60
3.3.5 Further Acceleration of the Carathéodory Block Coordinate Gradient Descent . . . 61
3.3.6 A list of rules . . . 63
3.3.7 Discussion of results . . . 65
3.4 Summary . . . 66

4 Recombined Lattice-Tree Models for Diffusions Approximation **67**
4.1 Introduction . . . 68
4.2 Background and Method . . . 73
4.2.1 Diffusion Approximations . . . 74
4.2.2 Matching Moments with Carathéodory's Theorem . . . 75
4.2.3 Constructing Lattice-Tree models . . . 76
4.3 Lattice-Tree Models for One-Dimensional Diffusions . . . 77
4.4 Lattice-Tree Models for Two-Dimensional Diffusions . . . 84
4.5 Lattice-Tree Models for Multi-Dimensional Diffusions . . . 95

4.6 Variations and Relation to Previous Work . . . 97
4.6.1 Transformations . . . 97
4.6.2 Optimal Control and Optimal Stopping . . . 100
4.6.3 Dealing with hypo-elliptic case. . . . 101
4.6.4 Switching Diffusions. . . . 101
4.6.5 More general processes. . . . 102
4.7 Algorithm and Experiments . . . 102
4.7.1 Solving the Moment Matching System . . . 103
4.7.2 Example: Mean reverting Heston model . . . 106
4.8 Summary . . . 109
4.A Appendix - Lemmas . . . 111

5 Grid-Free Computation of Probabilistic Safety Regions with Malliavin Calculus 115
5.1 Introduction . . . 115
5.2 Malliavin Calculus for stopping times . . . 121
5.3 Properties of the region $\mathbb{A}_T^p$ . . . 123
5.4 Exploration of the border $\partial\mathbb{A}_T^p$ . . . 126
5.5 Experiments . . . 128
5.6 Summary . . . 135

Bibliography 137

List of Figures

1 Structure of the thesis. . . . iii

1.1 Traffic light transition system. . . . 2

2.1 Cones and Negative Cones spanned by two and three points. . . . 18
2.2 Theoretical complexity of Algorithm 1. . . . 25
2.3 Proof of Theorem 2.11. . . . 27
2.4 Running time and number of iterations of the randomized algorithms as N varies. . . . 35
2.5 Top row: running time of randomized Algorithms vs deterministic algorithms as N varies. Bottom row: running time of the randomized Algorithm as n varies for different N. . . . 35
2.6 Histogram of the running time of Algorithm 2. . . . 36
2.7 Performance of the different algorithms on synthetic data. . . . 37
2.8 Divide and conquer, choice of the subgroup size. . . . 39
2.9 Running time of the Combined Algorithm 3. . . . 41

3.1 CaGD vs GD. . . . 53
3.2 Paths generated by CaGD and GD. . . . 57
3.3 Paths generated by CaBCD and GD. . . . 57
3.4 CaBCD vs ADAM vs SAG. . . . 61
3.5 CaBCD using the rules from [142], Part I. . . . 62
3.6 CaBCD using the rules from [142], Part II. . . . 65

4.1 Proof of Theorem 4.8. . . . 78
4.2 Building the transition matrix. . . . 108
4.3 Relative errors. . . . 109
4.4 Growth of the number of states. . . . 109
4.5 Sparse Matrix. . . . 110

5.1 Walk on the border. . . . 132
5.2 The 3 dimensional surface $\partial \mathbb{A}_1^{0.5}$. . . . 134
5.3 Experiments: points found while exploring the border. . . . 136

List of Algorithms

1 Basic measure reduction algorithm 21
2 Optimized measure reduction algorithm 28
3 Combined measure reduction algorithm 42
4 Carathéodory Sampling Acceleration 49
5 Carathéodory BCD - Modified Gauss-Southwell (GS) rule 58
6 Carathéodory BCD - Random 59
7 Recombining Lattice-Tree Construction 102
8 Exploration of the border $\partial \mathbb{A}_T^p$ 127
9 Adaptive constraint for the GD procedure 133

1 Formal Verification Introduction

Probabilistic Formal Verification (PFV) is a relatively new field of research, at the border between logic and probability theory. The purpose is to precisely compute the likelihood that a system verifies certain safety properties, expressed by means of Temporal Logic. One recent theoretical direction from a PVF research perspective is to extend its effectiveness to continuous-time and continuous-space models, but precision and the curse of dimensionality are consistent drawbacks. PFV motivation is strongly related to computers and electronic systems, and the techniques used are largely based on theoretical computer science, logic, decision diagrams and satisfiability theory. Model Checkers are the most regarded tools in Formal Verification, they allow to verify temporal logic properties of models in an automatic fashion. They are successfully used to verify protocols, softwares and other industrial applications, for both probabilistic and deterministic systems.
Model Checkers can deal with Markov chains in continuous or discrete-time, even with control inputs, but only if the space is discrete and finite.

1.1 Informal Introduction

The scope of this Section is to introduce the main ideas and procedures of formal verification, in particular Model Checking, Baier and Katoen [10]. In this Section, we prefer to be concise, with the hope of not neglecting the precision.

Definition 1.1 (Transition systems (TS))**.** *A transition system is a tuple (S, Act, $\rightarrow$, I, AP, L), where S is a set of countable finite states, Act is a set of actions, $\rightarrow \subseteq S \times Act \times S$ is a transition relation, $I \subseteq S$ is the set of initial states, AP is a finite set of atomic propositions, and $L : S \rightarrow 2^{AP}$ is a labelling function.*

The function L specifies for every state s a set of properties taken from 2^{AP}. In Figure 1.1 is represented a traffic light as an example of a transition system: five states, L maps any

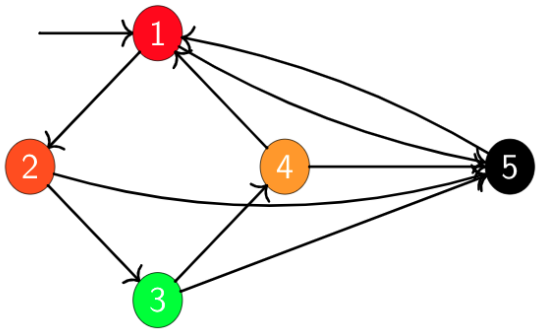


Figure 1.1: A traffic light transition system. Taken from the course "Computer-Aided Formal Verification" at University of Oxford, Prof Alessandro Abate.

state to a property, represented in this model by the colours {red, green, amber, black (not working)}, no actions are considered, the initial state set is the singleton state 1, and the transition relations are represented by the arrows.

A path fragment (or run or execution) is a sequence of states $\pi = s_0 s_1 s_2 ...$ such that the transition relation is respected. The notations used are $\pi[j] = s_j$, $\pi[j..] = s_j s_{j+1} ...$ and $\pi[..j] = s_0 ... s_j$. A trace is the output of a path: $L(s_0)L(s_1)L(s_2)....$ $Trace(TS)$ indicates all the possible traces that can be generated from TS. Similarly, $Trace(s_0)$ indicates all the possible traces that can be generated from the initial state s_0.

Definition 1.2. *A linear-time (LT) property P is a subset of all the possible sequences of properties in AP. The transition system satisfies P, $TS \models P$, if and only if $Traces(TS) \subseteq P$. Similarly, state $s \in S$ satisfies P, namely $s \models P$, if and only if $Traces(s) \subseteq P$.*

An example of an LT property for the traffic light, Figure 1.1, is that "a green light is always preceded by an amber one". The LT properties are defined with specific languages based on modal logics, which express temporal characteristics of the systems. This Section focuses on Linear Temporal Logic (LTL).

Definition 1.3 (Syntax of LTL)**.** *[10] "Recursively" a property ϕ in LTL is*

$$\phi := \quad true \quad | \quad a \quad | \quad \phi \wedge \phi \quad | \quad \neg\phi \quad | \quad \bigcirc \phi \quad | \quad \phi \cup \phi, \quad a \in AP.$$

Let π be a path, the semantics (meanings) of these operators are

$$\begin{aligned}
\pi \models \mathit{true} \quad &\mathit{iff} \quad \mathit{true}, & \pi \models a \quad &\mathit{iff} \quad a \in L(s_0),\\
\pi \models \phi \wedge \psi \quad &\mathit{iff} \quad \pi \models \phi \wedge \pi \models \psi, & \pi \models \neg\phi \quad &\mathit{iff} \quad \pi \not\models \phi,\\
\pi \models \bigcirc\phi \quad &\mathit{iff} \quad \pi[1..] \models \phi, & &\\
\pi \models \phi \cup \psi \quad &\mathit{iff} \quad \exists i \geq 0 : \pi[i..] \models \psi \wedge \forall 0 \leq j < i,\ \pi[j..] \models \phi. & &
\end{aligned}$$

Recursively means that properties like $\phi \cup ((\bigcirc \bigcirc \phi) \wedge \psi)$ are allowed. $\cup$ is called "Until", while $\bigcirc$ "Next". The "Until" operator allows to derive the temporal modalities "Eventually" $\Diamond$ - i.e. sometimes in the future - and "Always" - i.e. for any time period - as follows:

$$\Diamond\phi = \text{true} \cup \phi, \quad \text{and} \quad \Box \quad \phi = \neg\Diamond\neg\phi.$$

Thanks to this logic, it is plausible to declare non-trivial properties, for example concerning the traffic light system of Figure 1.1

- "The light is infinitely often red" is equivalent to $\Box\Diamond red$;
- "Once green, the light cannot become immediately red" is equivalent to $\Box(green \Rightarrow \neg \bigcirc red)$.

To decide if a transition system satisfies an LTL property, Formal Verification uses a set of procedures called Model Checkers. Although we do not focus on the algorithms used by the Model Checkers in this thesis, it is important to recall that the problem is generally NP-hard. Nevertheless, there exist various Model Checkers, for example in the deterministic case we cite SPIN github.com/nimble-code/Spin and NuSMV nusmv.fbk.eu.

Introducing randomness in Definition 1.1 is quite direct using transition probabilities, less obvious is the definition of "probabilistic logics" to study formulae of the type $\mathbb{P}[\Phi]$, i.e. the probability of satisfying Φ for a non-deterministic transition system. Probabilistic logics can be defined, e.g. Probabilistic Computation Tree Logic (PCTL), Baier and Katoen [10], Hansson and Jonsson [76], Kwiatkowska et al. [112] and probabilistic Model Checkers have been developed, e.g. PRISM, Kwiatkowska et al. [111] and MRMC, see mrmc-tool.org.

Probabilistic Model Checkers are "as smartest as possible" in dealing with the explosion of the state space, implementing efficient data structures, Binary Decision Diagrams (BDDs) and multi-terminal BDDs (MTBDDs) to symbolically represent the transition matrices. Nevertheless, we will see that the finiteness of the state space is a limit. However, thanks to formalism, automatism and excellent speed, these tools are attracting increasing attention from the industry.

1.2 Literature review

This Section presents a literature review on the modern procedures used in Formal Verification and the link with the related Mathematical ideas. The results are divided accordingly to the time modelization and the "reference literature": we first discuss literature from the Formal Verification society that assumes the model to be in discrete-time and then we pass to the continuous-time case; later we present the literature from the Mathematical community. The application considered is the reachability problem under safety specification, i.e. computing the distribution of the exit time of a safety region A. This problem has been deeply studied in the Formal Verification literature, as well as in the Mathematical one.

1.2.1 Formal Verification, Discrete-time models

Because of the Probabilistic Model Checkers speed and scalability, several researchers have tried to discretize the continuous-space of the states for discrete-time models. The intent is to prove the convergence of the desired probability, computing explicitly the error done, not "only" the order of convergence.

Optimal Control. In Abate et al. [4] the problem is defined in Stochastic Optimal Control terms using a value function composed of multiplication of indicator functions. The authors prove that a backward recursion algorithm works, similar to the Dynamic Programming Principle. An extension is presented in Summers and Lygeros [173] in the case of reach-avoid problems, i.e. maximizing the probability of hitting a target set, avoiding another one. A re-formulation of reach-avoid problems as martingale problems

can be found in Chatterjee et al. [35].

Abstraction by transition kernel. At a first glance, the "simplest" approach would be to find a "good" grid to discretize the state space. Nonetheless, the challenging part is to find explicitly the error done.

The first step towards this direction has been done in Abate et al. [3, 5]. If the transition probability admits density and is Lipschitz, the authors state the exact approximation error – not "just" the order of convergence – relatively to the reachability problem due to the discretization of the transition probability on a lattice $\delta\mathbb{Z}$. The form of the error is $O(n\delta k)$, where n is the number of time steps and k is the Lipschitz constant.

In [3, 5] the grid is generated instantaneously, instead in Soudjani and Abate [169] the concept of adaptive and sequential grid is introduced, which leads to less memory consumption, a lower number of states, but as a trade-off it is more expensive computationally. All the works from Soudjani and Abate culminate in the tool FAUST^2 [170]. Note that this result is far to be optimal. Although the authors suppose a discrete-time model, let us consider a standard Brownian Motion and a discretization of the time interval [0,1] in n steps. We know that using the grid $n^{-1/2}\mathbb{Z}$ the random walk converges to the Brownian Motion, however using the form of the error $O(n\delta k)$ in this case we would have $O(n^{1/2}k)$ which diverges as $n \to \infty$ since $k \to \infty$ too.

Excursus: Abstraction experiments. We have tested the abstraction techniques presented in [169]. Although we have seen that the convergence is not guaranteed for continuous time models, we have decided to test the scalability relatively to the dimension d. We want to compute the probability that a d-dimensional Brownian Motion W starting in the origin leaves a d-dimensional sphere with radius 1.5 centred in the origin during the time interval [0,1]. The solution of this problem is available analytically, Ciesielski and Taylor [37]: it can be expressed by means of Bessel processes $B_t(\nu) := \|W_t\|_2$, where $2\nu = d - 2$. The first hitting time distribution of the Bessel process is known, Borodin and Salminen [22] and therefore we can say that the probability distribution of the hitting time

is

$$P(\tau_r \leq t) = 1 - \frac{1}{2^{\nu-1}\Gamma(\nu+1)} \sum_{k=1}^{\infty} \frac{j_{\nu,k}^{\nu-1}}{J_{\nu+1}(j_{\nu,k})} e^{-\frac{j_{\nu,k}^2}{2r^2}t}.$$

Here Γ is the gamma function, J_ν is the Bessel function of the first kind of order ν and $\{j_{\nu,k}\}_{k=1}^{\infty}$ is the sequence of positive zeros of J_ν. First, FAUST^2 [170] permits to request an upper-bound of the error on the desired probability, i.e. we can indicate $\varepsilon > 0$ s.t. $|P - P^n| < \varepsilon$, where P indicates the true probability and P^n the approximated one by FAUST^2. Therefore, a balanced request would be an error of 0.01 or lower to have acceptable outcomes. However, it is important to recall that this error is respected for the discrete-time problem, so does not consider the time discretization error. i.e the Euler Scheme. Second, FAUST^2 deals exclusively with linear safety regions, therefore the sphere is not supported, nevertheless in dimension one this problem does not subsist. Before starting the execution, FAUST^2 returns the expected running time. Unfortunately, this method has shown remarkable and unexpected weaknesses in the continuous time case, see Table 1.1.

Dim.	Abs. Err. Requested	Time Steps requested	Time/Expected Time	Returned Probability	Exact Probability	# States of the MC created
1	0.1	10	7 min	0.8073	0.267	1089
	0.5	20	4 min	0.7895	0.267	872
	0.5	50	3 hours	N. E.	0.267	N. E.
	0.01	100	11 years	N. E.	0.267	N. E.
2	1	10	12 years	N. E.	0.5581	N. E.
	0.01	100	Out of memory	N. E.	0.5581	N. E.

Table 1.1: FAUST^2 execution time. N.E. means Not Executed.

1.2.2 Formal Verification, Continuous-time models

The continuous-time case has been studied, and as in the discrete-time case, different procedures have been deepened. The problem is the same, computing the exit time of a safety region, i.e. solve a reachability problem. The main difficulty is to track the error precisely. From a mathematical perspective, it is well known that the solution can be represented via Partial Differential Equations (PDE) techniques, see below.

If the time discretization error can be tolerated, one can immediately use the discretization of the space of the previous subsection, although we have seen that the task is computationally expensive. In the deterministic case, a Hamilton-Jacobi (HJ) PDE is obtained. In Tomlin et al. [179, 180], based on Osher [143], a PDE solver is developed based on level-set methods. They study the problem both with and without controls. In Broucke et al. [29], for the non-stochastic case, an equivalent formulation based on automata and bi-simulations has been explored, therefore the problem is reduced to something closer to the procedures used by Model Checkers.

In the stochastic case, in Wisniewski and Bujorianu [184] is proposed an optimization scheme based on a Dirichlet problem, but no practical implementations are given. The same authors in [188] characterize the set of the safe initial states through another optimization problem, this time linked with optimal measures. An example is then given for piecewise deterministic Markov processes. In Prajna et al. [148], the concept of Barrier certificates is used in a worst-case scenario approach. In the following, we present approaches closer to our point of view.

Bi-simulation. The bi-simulation concept is introduced in Julius and Pappas [93], Zamani et al. [192], which aims to obtain an "abstraction" of the model. Lyapunov-like functions are used to rigorously quantify the distance between a model and its abstraction, under a specified metric. The use of a metric is critical to these approaches, indeed the results depend on the metric chosen. In Abate [1] the author finds sufficient conditions for the existence of bi-simulation functions, by means of "contractiveness" concept for SDE.

Interval Markov chains. Another approach is the one studied in Laurenti et al. [113]. The authors deal with reachability problems under safety specifications with respect to polytopes regions in the special case of linear switching diffusions. They search for an Interval Markov chain (IMC) approximation. The technique uses a Bronwian Bridge to control the error in the time discretization, which implies a strong dependence on the linearity of the diffusion. To obtain an error estimate, they bound the desired probability from above and below using IMCs. From an optimal control perspective, they interpret the

evolution of the IMC as a 2-players adversarial stochastic game, where Player 1 chooses an action and Player 2 chooses a feasible transition probability distribution. Hence, the goal becomes to synthesize a strategy for Player 1 that is robust against all adversaries.

Dynamic Programming. The reachability problem can be characterized as the solution of Hamilton Jacobi Bellman equations. The theory in the case of non-smooth value functions is developed in Bouchard and Touzi [24], where viscosity solutions are used. In Bujorianu and Lygeros [30], Esfahani et al. [54], Mitchell [130], more general reach-avoids problems has been studied through PDE techniques.

1.2.3 Mathematical Literature

We have presented the literature review regarding the methods to approximate the distribution of the exit time from a Formal Verification perspective. Now we explore the literature review from a Mathematical perspective.

Let us consider a h-dimensional Brownian motion W defined on a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t), \mathbb{P})$, a region A in $\mathbb{R}^d$, and the following SDE

$$\begin{aligned} X_t =& \mu(X_t)dt + \sigma(X_t)dW_t, \\ X_0 =& x_0 \in A, \end{aligned} \tag{1.1}$$

where $\mu : \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \to \mathbb{R}^{d\times h}$. In general, $X^{s,x}$ denotes the solution of (1.1) starting from x at time s. When there is no confusion, we denote $X^{s,x}$ as X and we use X^x if $s = 0$. In the following C^n indicates the space of functions n-times differentiable, C^n_b is the space of functions n-times differentiable with bounded derivatives.

For $n > 0$, we call $X^n = (X^n_{t_i})_{i=0,1,2,\dots}$, $t_i = in^{-1}$, a Markov chain and whenever clear from the context we shall indicate $X^n_{t_i}$ as X^n_i. We call $\tau^x_A := \inf\{t \geq 0 : X^x_t \notin A\}, x \in A$ the exit time of X^x from the region A, and similarly we define $\tau^{x,n}_A := \inf\{t_i = in^{-1} : X^n_{t_i}, \notin A\} X^n_0 = x \in A$. If there cannot be confusion we denote τ^x_A (resp. $\tau^{x,n}_A$) as τ_A or τ (resp. τ^n_A or τ^n).

The idea of discretizing the time to approximate the law of X is not new, e.g. Ethier and Kurtz [55], Kloeden and Platen [98], Kushner [108], Milstein [129], Stroock and Varadhan [171], Talay [174], Talay and Tubaro [175]. Two different kind of errors are usually considered: the "strong" and the "weak" error, [98]. If we consider a Markov chain X^n, the strong error often is associated to evaluating the L^p difference $\mathbb{E}|X_t - X_n^n|^p$, whilst the weak error studies $|\mathbb{E}f(X_t) - \mathbb{E}f(X_n^n)|$, $f \in \mathbb{F}$, for a class of functions $\mathbb{F}$. Note that for the strong error X and X^n must be defined on the same probability space. Moreover, we can see that $|\mathbb{E}[X_t] - \mathbb{E}[X_n^n]| \leq \mathbb{E}|X_t - X_n^n|$.

An approximation scheme is said to converge weakly with respect to a class of function $\mathbb{F}$ if

$$\lim_{n\to\infty} |\mathbb{E}f(X_n^n) - \mathbb{E}f(X_t)| = 0, \quad f \in \mathbb{F}$$

and similarly for the strong convergence. It is said that X^n has "weak order" $\beta > 0$ respect to a class of function $\mathbb{F}$ if

$$\mathbb{E}f(X_t) - \mathbb{E}f(X_n^n) = O(n^{-\beta}), \quad \text{as } n \to \infty, \quad f \in \mathbb{F}$$

and similarly can be defined the "strong order" of convergence. The difference between the strong and weak order of convergence implies that the same scheme X^n can have two different orders of convergence, for example the standard Euler Scheme has strong-order 0.5 and weak-order 1 under certain assumptions on μ, σ, f [98, 175]. In the following, we focus on the weak error.

Once the (order of) convergence of a scheme is known, we can approximate $\mathbb{E}f(X_t)$ evaluating $\mathbb{E}f(X_n^n)$. To compute the expectation we consider two alternatives:

(i) to use Monte Carlo techniques. This is the most efficient way, given the level of possible parallelization in modern machines. The drawback is that the error is statistical, i.e.

$$\mathbb{E}f(X_n^n) - \sum_{i=1}^{N} \tfrac{1}{N} f\left(X_n^{n\ (i)}\right) \sim N\left(0, \tfrac{c}{\sqrt{N}}\right),$$

where $N(a,b)$ stands for the normal distribution with mean a and variance b, N

is the number of samples and $X_n^{n\ (i)}$ represents the i-th sample. The most attractive characteristic of Monte Carlo techniques is the low impact of the curse of dimensionality.

(ii) To compute exactly $\mathbb{E}f(X_n^n)$. This is the ambition of the Formal Verification community, indeed if X^n is a discrete space Markov chain we can use the model checkers, if f represents a "logic" formula in PCTL. The dimension of the state space would depend almost surely on the dimensions d and h, therefore this alternative would probably suffer from the curse of dimensionality, although it eliminates the statistical error of Monte Carlo methods.

If the function f and the coefficients μ, σ are C_b^4, and the scheme X^n satisfies the local consistency condition, see Definition 4.4, in [98] it is proved that,

$$\mathbb{E}f(X_t) - \mathbb{E}f(X_n^n) = O(n^{-1}).$$

In [98] the result is extended to higher order schemes, assuming more smoothness for f, μ, σ and modifying accordingly the local consistency condition. In Talay and Tubaro [175] an expansion of the error in terms of $n^{-1}, n^{-2}, ...$ is stated, using the Euler Scheme

$$X_{i+1}^n = X_i^n + n^{-1}\mu(X_i^n) + n^{-1/2}\sigma(X_i^n) \cdot W_i, \qquad W_i \sim N(0,1).$$

The expansion allows the possibility to use Romberg extrapolation techniques to obtain higher order schemes.

More generally, a similar expansion of the approximation error given in Talay and Tubaro [175] exists for the Euler Scheme, Bally and Talay [12], if f is only measurable and bounded and the coefficients μ, σ are bounded and in C_b^∞. The proof is based on Malliavin calculus, indeed the vector fields μ and σ are required to satisfy the Hörmander's condition, see e.g. Bally [11], Hörmander [84]. Since the proof is strictly related to Malliavin calculus, it seems difficult to us to extend the idea to discrete-space Markov chains. During the confirmation exam, Prof. Samuel Cohen pointed us to Bender and Parczewski [15], where the authors obtain (strong and weak) L^2 convergence results for relevant functionals in

Malliavin Calculus. In particular, they do not require the absolute continuity of the measures of the random variables approximating the desired stochastic processes. However, because of the strong convergence and the absence of a rate of convergence we do not see an immediate application to the case considered by us.

A complete different procedure is used in Konakov and Mammen [100, 102, 103] where a parametrix method mixed with an edgeworth type expansion is used to prove that

$$|\mathbb{P}_x\left(X_t \in dx\right) - \mathbb{P}_x\left(X_n^n \in dx\right) + c_1 n^{-1/2} + c_2 n^{-1}| = o(n^{-1})$$

for some specified functions c_1, c_2, under assumptions similar to [98] (see Chapter 4, Definition 4.4) and if the measure $\mathbb{P}\left(X_i^n - X_{i-1}^n \in dx\right)$ is absolutely continuous respect to the Lebesgue measure. In Konakov and Mammen [101] the authors prove a similar statement - up to the order $n^{-1/2}$ - if the measure $\mathbb{P}\left(X_i^n - X_{i-1}^n \in dx\right)$ has support on a Lattice.

For the reader interested in higher order scheme we refer to Kloeden and Platen [98], Milstein [129], Talay and Tubaro [175] and for "different" approaches Kusuoka [110], Lyons and Victoir [125], Ninomiya and Ninomiya [140]. If the interest is in weaker assumptions on f, μ, σ, we refer to Ethier and Kurtz [55], Gyöngy and Rásonyi [74], Kohatsu-Higa et al. [99], Konakov and Menozzi [104], Kushner [108], Stroock and Varadhan [171].

In Kushner [108] the convergence of controlled Markov chains to controlled SDEs is proved, without the rate, using "minimal assumptions". Let U be a compact set of possible controls and v be a continuous and bounded function. If for some $\beta > 0$

$$\mathbb{E}\left[\left(X_{t_{i+1}}^n - X_{t_i}^n\right)^{\otimes m} \middle| X_{t_i}^n = x\right] - \mathbb{E}\left[\left(X_{t_{i+1}} - X_{t_i}\right)^{\otimes m} \middle| X_{t_i} = x\right] = O(c_x n^{-1-\beta}), \quad m = 1,2$$

$$\left|X_{t_{i+1}}^n - X_{t_i}^n\right| \to 0, \quad a.s.$$

uniformly in $i \leq n$, where $t_i = in^{-1}$, then [108]

$$\lim_{n\to\infty} \min_{u\in U} \mathbb{E}\left[\sum_{i=0}^{\tau_n} n^{-1} v\left(X_{t_i}(u)\right) \Big| X_0^n = x\right] = \min_{u\in U} \mathbb{E}\left[\int_0^{\tau} v\left(X_s(u)\right) ds \Big| X_0 = x\right].$$

τ and τ^n represents the exit times from a compact region A with piecewise smooth boundary of X and X^n. Quantifying precisely the error done is difficult: to prove the results in [108] is used the tightness of sequences of measures plus the uniqueness of weak solutions of SDEs.

In Stroock and Varadhan [171, Section 11.2], Ethier and Kurtz [55, Section 7.4] it is proved the convergence in measure of Markov chains to the "corresponding" SDE solutions. The proof is done using the martingale problem characterization, together with the "pre-compactness" of the approximating measures sequence. As per [108], the error seems difficult to be quantified, given the use of "pre-compactness" results, strictly related to the tightness concept.

Exit times. In all the previously cited works the interest is in approximating a function of the solution X of (1.1) at a final deterministic time point t, which does not include the possibility to approximate the distribution of the exit time τ. In the one dimensional case, there exists an expansion of $\mathbb{P}(\tau < t)$ for the Brownian motion in terms of τ^n, $n^{-1/2}$ and $o(n^{-1/2})$, Siegmund and Yuh [166]. The proof is based on edgeworth type expansions and could be extended to the case of discrete random variables, but we do not think the ideas can be extended to general diffusions and regions A in the multidimensional case.
Under ellipticity conditions on σ, for functionals of the type

$$\mathbb{E}[f(X_t)\mathbb{1}_{\tau<t}], \tag{1.2}$$

the discrete Euler scheme has rate of convergence 1/2, Gobet [70] if either *(i)* f is smooth and the derivatives at border of the region A vanish[1] or *(ii)* f has support strictly included in A. In Gobet [70] is proved also that the rate of convergence for the functionals (1.2) is 1

[1]The precise statement is more general.

if the "continuous-time" Euler scheme is considered

$$X_t^n = x + \int_0^t \mu(X_{\varphi(s)}^n)ds + \int_0^t \sigma(X_{\varphi(s)}^n)dW_s,$$

where $\varphi(s) = \sup\{in^{-1} : in^{-1} \leq s\}$, i.e. if one allows the scheme to move between two different discretization times. This scheme is interesting theoretically, but generally it is not feasible to simulate the stochastic integral term unless the considered region is particular, e.g. half-space or the dimension is 1. Under the more general Hörmander's condition we refer to Gobet and Menozzi [71], although f in this case must have support strictly included in A, which does not include the possibility to study $\mathbb{P}(\tau < t)$. The authors in [70, 71] use the well-known fact, see e.g. Friedman [62], that $\mathbb{E}_x[f(X_T)\mathbb{1}_{\tau<T}]$ can be represented as the solution to the following PDE with Cauchy and Dirichlet conditions

$$\begin{aligned} \partial_t v + Lv &= 0, && \text{for } (t,x) \in [0,T) \times A \\ v(t,x) &= 0, && \text{for } (t,x) \in [0,T) \times A^c \\ v(T,x) &= f(x), && \text{for } x \in A \end{aligned}$$

and L is the infinitesimal generator of the SDE (1.1). In [70] it is emphasized that the rate $O(n^{-1/2})$ is intrinsic in the discretization time procedure, thus using discrete time steps we cannot hope in a better order of convergence, see also Gobet and Menozzi [72]. The proof in [70] is based on Malliavin Calculus and uses properties of the probability density of the Euler scheme. We are not aware of general theories to extend the Malliavin Calculs to discrete random variables, except the previous mentioned work [15]. In Gobet and Menozzi [73], the same authors prove that, shrinking ("overshooting") the region A in a particular way, the order of convergence can be improved to $o(n^{-1/2})$ if the functional is smooth and they conjecture from experiments that this procedure can lead to an order of convergence $O(n^{-1})$.

A completely different approach, namely an "Exact Simulation" algorithm, is presented in Beskos and Roberts [16]. We do not deepen this approach because it is not based on standard Markov chains approximations. The authors assume an ellipticity condition on

σ and, given its dependence on Brownian bridges, it is limited to the 1 dimensional case. Nevertheless, in dimension 1 the experiments show that it is to be preferred to the classical Euler Scheme. It is based on a rejection-sampling strategy and other algorithms to simulate exit times have been developed in dimension 1 using a similar approach, e.g. Herrmann and Zucca [79]. The same algorithm is studied if the dimension is greater than 1, Casella and Roberts [34], under special structures of the SDE (1.1).
Another technique used is to adapt the time-step strategically during the simulations with Monte Carlo approaches, e.g. Dzougoutov et al. [52]. We discard these techniques because the adaptive strategy must be done "online" and therefore we do not obtain a Markov chain.

Another approach is the one presented in Frikha et al. [63, 64], where the authors obtain unbiased estimates for functionals $\mathbb{E}[f(X_t)\mathbb{1}_{\tau<t}]$ when the dimension is 1.

2 A Randomized Algorithm to Reduce the Support of Discrete Measures

Keywords: Recombination, Randomized algorithms, Discrete Measures, Tchakaloff's Theorem, Carathéodory's Theorem

Given a discrete probability measure supported on N atoms and a set of n real-valued functions, there exists a probability measure that is supported on a subset of $n+1$ of the original N atoms and has the same mean when integrated against each of the n functions. If $N \gg n$ this results in a huge reduction of complexity. We give a simple geometric characterization of barycenters via negative cones and derive a randomized algorithm that computes this new measure by "greedy geometric sampling". We then study its properties, and benchmark it on synthetic and real-world data to show that it can be very beneficial in the $N \gg n$ regime. A Python implementation is available at github.com/FraCose/Recombination. This Chapter contains the material presented in Cosentino et al. [40].

2.1 Introduction

Discrete probability measures are central to many inference tasks, for example as empirical measures. In the "big data" regime, where the number N of samples is huge, this often requires to construct a reduced summary of the original measure. Often this summary is constructed by sampling n points at random out of the N points, but Tchakaloff's theorem suggests that there is another way.

Theorem 2.1. *[Tchakaloff [176]] Let μ be a discrete probability measure that is supported on N points in a space $\mathcal{X}$. Let $\{f_1, \ldots, f_n\}$ be a set of n real-valued functions $f_i : \mathcal{X} \to \mathbb{R}$, $n < N$. There exists a discrete probability measure $\hat{\mu}$ such that $\operatorname{supp} \hat{\mu} \subset \operatorname{supp} \mu$, $|\operatorname{supp} \hat{\mu}| \leq n + 1$, and*

$$\mathbb{E}_{X \sim \mu}[f_i(X)] = \mathbb{E}_{X \sim \hat{\mu}}[f_i(X)] \text{ for all } i \in \{1, \ldots, n\}. \tag{2.1}$$

We introduce a randomized algorithm that computes $\hat{\mu}$ efficiently in the $n \ll N$ regime. Reducing the support of a (not necessarily discrete) measure subject to matching the mean on a set of functions is a classical problem, which goes back at least to Gauss' famous quadrature formula that matches the mean of monomials up to a given degree when μ is the Lebesgue measure on $\mathbb{R}$. In multi-dimensions this is known as cubature and Tchakaloff [176] showed the existence of a reduced measure for compactly supported, not necessarily discrete, measures, see Bayer and Teichmann [14]. When μ is discrete, the problem of computing $\hat{\mu}$ also runs under the name recombination. Algorithms to compute $\hat{\mu}$ for discrete measures μ go back at least to Davis [50] and have been an intensive research topic ever since; we refer to Litterer and Lyons [119], Maalouf et al. [126], Tchernychova [177] for an overview of the different approaches, and for recent, state of the art algorithms and applications. Using randomness for Tchakaloff's Theorem has been suggested before in Hayakawa [77], Piazzon et al. [145] but the focus there is to show that the barycenter lies in the convex hull when enough points from a continuous measure are sampled so that subsequently any of the algorithms in [119, 126, 177] can be applied; in contrast, our randomness stems from the reduction algorithm itself. More generally, the topic of replacing a large data set by a small, carefully weighted subset is a vast field that has attracted many different communities and we mention, pars pro toto, coresets in computer science [144], scalable Bayesian statistics [88], clustering and optimisation [56].

The above mentioned algorithms [119, 126, 177] use a divide and conquer approach that splits up points into groups, computes a barycenter for each group, and solves a constrained linear system several times. This leads to a deterministic complexity that is determined by N and n. In contrast, our approach uses the geometry of cones to "greedy" sample for

candidates in the support of μ that are atoms for $\hat{\mu}$ and tries to construct the reduced measure in one go. Further, it can be optimized with classical black box reset strategies to reduce the variance of the run time. Our results show that this can be very efficient in the big data regime $N \gg n$ that is common in machine learning applications, such as least square solvers when the number of samples N is very large. Moreover, our approach is complementary to previous works since it can be combined with it: by limiting the iterations for our randomized algorithm and subsequently running any of the deterministic algorithms above if a solution by "greedy geometric sampling" was not found, one gets a hybrid algorithm that is of the same order as the deterministic one but that has a good chance of being faster; we give full details in Appendix 2.B.

We introduce the basic ideas in Section 2.2, where we derive a simple version of the greedy sampling algorithm, and study its theoretical properties. In Section 2.3 we optimize the algorithm to better use the cone geometry, combine with reset strategies to reduce the running time variance, and use the Woodbury formula to obtain a robustness result. In Section 5.5 we discuss numerical experiments that study the properties of the algorithms on two problems: reducing the support of empirical measures and least square solvers for large samples.

2.2 Negative cones and a naive algorithm

As is well-known, Tchakalof Theorem 2.1 follows from Carathéodory's convex hull theorem

Theorem 2.2. *[Carathéodory [51]] Given a set of N points in $\mathbb{R}^n$ and a point x that lies in the convex hull of these N points. Then x is a linear combination of at most $n+1$ points from the N points.*

It is instructive to recall how Theorem 2.2 implies Tchakalof Theorem 2.1. Therefore define a $\mathbb{R}^n$-valued random variable $F : \Omega = \mathcal{X} \to \mathbb{R}^n$ as $F(\omega) := (f_1(\omega), \ldots, f_n(\omega))$ and note that Equation (2.1) is equivalent to

$$\int_\Omega F(\omega)\mu(d\omega) = \int_\Omega F(\omega)\hat{\mu}(d\omega).$$

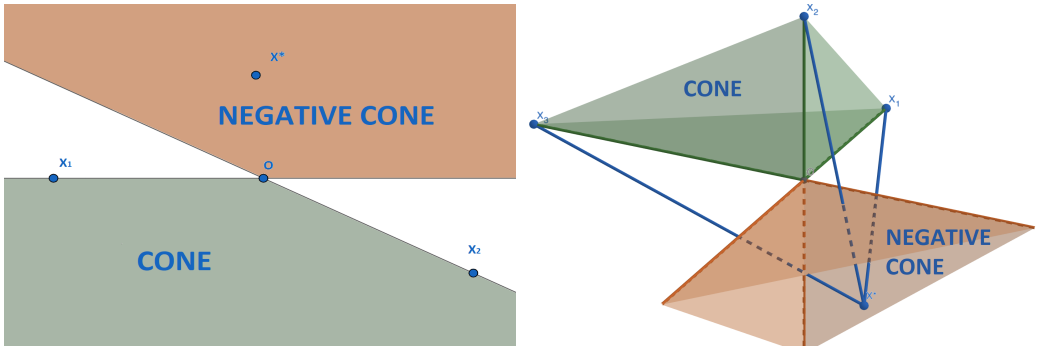


Figure 2.1: Cones and Negative Cones spanned by two (left) and three points (right).

Since μ has finite support, the left-hand side can be written as a sum $\sum_{\omega \in \operatorname{supp} \mu} F(\omega)\mu(\omega)$. This sum gives a point in the convex hull of the set of N points $\mathbf{x} := \{F(\omega) : \omega \in \operatorname{supp} \mu\}$ in $\mathbb{R}^n$. But by Carathéodory's theorem, this point must be a convex combination of a subset $\hat{\mathbf{x}}$ of only $n+1$ (or less) points of $\mathbf{x}$ and Tchakalof Theorem 2.1 follows. This proof of Tchakalof Theorem 2.1 is also constructive in the sense that it shows that computing $\hat{\mu}$ reduces to constructing the linear combination guaranteed by Carathéodory's theorem; e.g. by solving N times a constrained linear system, see Davis [50].

Key to Tchakaloff's theorem is to verify if two measures have the same mean. We now give a simple geometric characterization in terms of negative cones, Theorem 2.4.

Definition 2.3. *Let $\mathbf{x} \subset \mathbb{R}^n$ be a finite set of points in $\mathbb{R}^n$. We call the set $C(\mathbf{x}) := \{c \in \mathbb{R}^n : c = \sum_{x \in \mathbf{x}} \lambda_x x,$ where $\lambda_x \geq 0\}$ the cone generated by $\mathbf{x}$ and we also refer to $\mathbf{x}$ as its basis. We call the set $C^-(\mathbf{x}) := \{c \in \mathbb{R}^n : c = \sum_{x \in \mathbf{x}} \lambda_x x,$ where $\lambda_x \leq 0\}$ the negative cone generated by $\mathbf{x}$.*

For example, $C(x_1,x_2)$ is the "infinite" triangle created by the half-lines $\overline{0x_1}$, $\overline{0x_2}$ with origin in 0; $C(\{x_1,x_2,x_3\})$, is the infinite pyramid formed by the edges $\overline{0x_1}$, $\overline{0x_2}$, $\overline{0x_3}$, with vertex 0; see Figure 2.1.

To prepare the proof of Theorem 2.4 we want to recall that a well-known tool from discrete geometry, is that polygons and polyhedral descriptions are equivalent. That is $C(\mathbf{x})$ is an affine span of the vectors $\mathbf{x}$ as in Definition 2.3, but equivalently $C(\mathbf{x})$ is the intersection of half-spaces. In general, this is the content of the celebrated Weyl–Minkowksi

theorem and computing one representation from the other is non-trivial, see Ziegler [194]. However, when restricted to n generic vectors in $\mathbb{R}^n$ (as is the case required in Theorem 2.4), one can immediately switch from one to the other, see item 2 of Theorem 2.4.

Theorem 2.4. *Let $\mathbf{x} = \{x_1, \ldots, x_{n+1}\}$ be a set of $n+1$ points in $\mathbb{R}^n$ such that $\mathbf{x} \setminus \{x_{n+1}\}$ spans $\mathbb{R}^n$. Let A be the matrix that transforms $\mathbf{x} \setminus \{x_{n+1}\} = \{x_1, \ldots, x_n\}$ to the orthonormal basis $\{e_1, \ldots, e_n\}$ of $\mathbb{R}^n$, i.e. $Ax_i = e_i$. Further, let h_i be the unit vector such that $\langle h_i, x\rangle = 0$ for all $x \in \mathbf{x} \setminus \{x_i, x_{n+1}\}$ and $\langle h_i, x_i\rangle < 0$ and denote with $H_{\mathbf{x}\setminus\{x_{n+1}\}}$ a $n \times n$ matrix that has $h_1, \ldots, h_n$ as row vectors. It holds that*

1. *$C(\mathbf{x} \setminus \{x_{n+1}\}) = \{c : H_{\mathbf{x}\setminus\{x_{n+1}\}} c \leq 0\}$ and $C^-(\mathbf{x} \setminus \{x_{n+1}\}) = \{c : H_{\mathbf{x}\setminus\{x_{n+1}\}} c \geq 0\}$.*
2. *$Ax \geq 0$ if and only if $H_{\mathbf{x}\setminus\{x_{n+1}\}} x \leq 0$ and $Ax \leq 0$ if and only if $H_{\mathbf{x}\setminus\{x_{n+1}\}} x \geq 0$.*
3. *There exists a convex combination of $\mathbf{x}$ with 0 as barycentre, $\sum_{i=1}^{n+1} w_i x_i = 0$ for some $w_i > 0$, and $\sum_{i=1}^{n+1} w_i = 1$ if and only if $x_{n+1} \in C^-(\mathbf{x} \setminus \{x_{n+1}\})$.*

The above result could be formulated without the matrix A, only in terms of $H_{\mathbf{x}\setminus\{x_{n+1}\}}$. However, A is the inverse of the matrix with columns equal to the vectors $\{\mathbf{x} \setminus \{x_{n+1}\}\}$, hence computing A is more efficient than computing $H_{\mathbf{x}\setminus\{x_{n+1}\}}$, since matrix inversion is optimized in standard libraries.

Proof. For item 1 first note that the h_i are well-defined since any set of $n-1$ independent points determines a hyperplane that includes 0 and that divides $\mathbb{R}^n$ into two parts. Each of these two parts is of the form $\{x : \langle h,x\rangle \leq 0\}$ or $\{x : \langle h,x\rangle > 0\}$ and the additional condition $\langle h,x_i\rangle < 0$ selects one of the two parts. Now let $c = \sum_{x\in\mathbf{x}\setminus\{x_{n+1}\}} w_x x$ be a general vector. Since $H_{\mathbf{x}\setminus\{x_{n+1}\}} x \leq 0$, for $x \in \mathbf{x} \setminus \{x_{n+1}\}$, it follows that $H_{\mathbf{x}\setminus\{x_{n+1}\}} c = \sum_{x\in\mathbf{x}\setminus\{x_{n+1}\}} w_x H_{\mathbf{x}\setminus\{x_{n+1}\}} x$ satisfies $H_{\mathbf{x}\setminus\{x_{n+1}\}} c \leq 0$ if and only if the w_x, for $x \in \mathbf{x} \setminus \{x_{n+1}\}$, are positive.

For item 2, we can write $x = \sum_{i=1}^{n} w_i x_i$, for some $w_i \in \mathbb{R}$, hence $Ax = (w_1, \ldots, w_n)^\top$. By definition

$$\begin{aligned} H_{\mathbf{x}\setminus\{x_{n+1}\}} x =& \sum w_i H_{\mathbf{x}\setminus\{x_{n+1}\}} x_i = \sum w_i (\langle h_1,x_i\rangle, \ldots, \langle h_n,x_i\rangle)^\top \\ =& (w_1\langle h_1,x_1\rangle, \ldots, w_n\langle h_n,x_n\rangle)^\top. \end{aligned}$$

Note that $\langle h_i, x_i \rangle \leq 0$ by definition, therefore sign $\{-Ax\}$ = sign $\{H_{\mathbf{x}\setminus\{x_{n+1}\}}x\}$. The statement with the reversed inequalities follows similarly.

For item 3($\Rightarrow$) assume there exists a convex combination of $\mathbf{x}$, this means that $x_{n+1} = -\frac{1}{w_{n+1}} \sum_{i=1}^{n} w_i x_i$, and

$$Ax_{n+1} = -\frac{1}{w_{n+1}} \sum_{i=1}^{n} w_i Ax_i = \sum_{i=1}^{n} -\frac{w_i}{w_{n+1}} e_i.$$

Therefore, $Ax_{n+1} \leq 0$ which is by item 2 equivalent to $H_{\mathbf{x}\setminus\{x_{n+1}\}}x \geq 0$. Thus, $x_{n+1} \in C^-(\mathbf{x} \setminus \{x_{n+1}\})$. Finally, for item 3($\Leftarrow$) assume that $x_{n+1} \in C^-(\mathbf{x} \setminus \{x_{n+1}\})$. The by item 2 $Ax_{n+1} \leq 0$. Moreover, there exits $\lambda_i \in \mathbb{R}$ such that $x_{n+1} = \sum_{i=1}^{n} \lambda_i x_i$, therefore $Ax_{n+1} = \sum_{i=1}^{n} \lambda_i e_i = (\lambda_1, \ldots, \lambda_n)^\top \leq 0$. Let us call $\lambda^* := 1 - \sum_{i=1}^{n} \lambda_i$, by the decomposition of x_{n+1} we know that

$$\frac{1}{\lambda^*} x_{n+1} + \sum_{i=1}^{n} \frac{-\lambda_i}{\lambda^*} x_i = 0 \text{ and } \frac{1}{\lambda^*} + \sum_{i=1}^{n} \frac{-\lambda_i}{\lambda^*} = 1 \text{ and } \frac{1}{\lambda^*}, \frac{-\lambda_i}{\lambda^*} \geq 0.$$

□

Remark 2.5. *The assumption that $\{\mathbf{x}\setminus\{x_{n+1}\}\}$ spans $\mathbb{R}^n$ can be relaxed, indeed item 1 is a particular case of the Weyl's Theorem, which briefly does not require the independence of the cone basis. From an implementation point of view, however, item 2 gives an important boost, indeed the computation of $H_{\mathbf{x}\setminus\{x_{n+1}\}}$ is heavier than inverting a matrix, i.e. computing A, since it requires the computation of the coefficients of n different hyperplanes in $\mathbb{R}^n$. Moreover, speaking about the greedy searching strategy of Algorithm 2, see below Section 2.3, using $H_{\mathbf{x}\setminus\{x_{n+1}\}}$ in place of A does not allow the use of the Sherman–Morrison formula weighing even more on the total computational cost.*

A Naïve Algorithm. Theorem 2.4 implies a simple randomized algorithm: sample n points at random until the negative cone spanned by the n points is not empty. Then item 3 of Theorem 2.4 implies that the n points in the cone and any point in the negative cone form the support of a reduced measure. If $\mathbb{E}_\mu[X] \neq 0$ we can always study the points $\mathbf{x} - \mathbb{E}_\mu[X]$, this is equivalent to work with cones whose vertex is not 0.

Algorithm 1 Basic measure reduction algorithm

procedure REDUCE(A set $\mathbf{x}$ of N points in $\mathbb{R}^n$)
  Choose n points $\mathbf{x}^\star$ from $\mathbf{x}$
  while $C^-(\mathbf{x}^\star) \cap \mathbf{x} = \emptyset$ **do**
    Replace $\mathbf{x}^\star$ with n new random points $\mathbf{x}^\star$ from $\mathbf{x}$
  end while
  $\mathbf{x}^\star \leftarrow \mathbf{x}^\star \cup x^\star$ with an arbitrary $x^\star \in C^-(\mathbf{x}^\star) \cap \mathbf{x}$
  Solve the linear system $\sum_{x \in \mathbf{x}^\star} w_x^\star x = 0$ for $w^\star = (w_x^\star)_{x \in \mathbf{x}^\star}$
  return ($\mathbf{x}^\star$,$w^\star$)
end procedure

Corollary 2.6. *Algorithm 1 computes a reduced measure $\hat{\mu}$ as required by Tchakalof Theorem 2.1 in $\tau \cdot O(n^3 + Nn^2)$ computational steps. Here $\tau = \inf\{i \geq 1 : C^-(\mathbf{X}_i) \cap \mathbf{x} \neq \emptyset\}$, where $\mathbf{X}_1, \mathbf{X}_2, \ldots$ are random sets of n points sampled uniformly at random from $\mathbf{x}$.*

The complexity of Algorithm 1 in a single loop iteration is dominated by

(i) computing the matrix A that defines the cones $C^-(\mathbf{x}^\star)$ and $C(\mathbf{x}^\star)$,

(ii) checking if there are points inside the cones,

(iii) solving a linear system to compute the weights $w_i^\star$.

Respectively, the worst case complexities are $O(n^3)$, $O(Nn^2)$ and $O(n^3)$, since to check if there are points inside the cones we have to multiply A and $\mathbf{X}$, where $\mathbf{X}$ is the matrix whose rows are the vector in $\mathbf{x}$.

Proposition 2.7. *Let $N > n + 1$ and μ be a discrete probability with finite support and $f_1, \ldots, f_n$ be as in Tchakalof Theorem 2.1. Moreover wlog assume $\mathbb{E}_{X \sim \mu}[f_i(X)] = 0$ for $i = 1, \ldots, n$. With $p := \frac{n \cdot n!(N-n)!}{N!}$ it holds that $\mathbb{E}[\tau] \leq \frac{1}{p}$ and $Var(\tau) \leq \frac{1-p}{p^2}$ and, for fixed n, $\lim_{N \to \infty} \mathbb{E}[\tau] = 1$.*

Proof. Note that in every run through the loop, n points are randomly selected. However, by Theorem 2.4, Algorithm 1 finishes when the event

$$A := \{\text{for } n \text{ uniformly at random chosen points } \mathbf{x}^\star \text{ from } \mathbf{x},\ \exists x \in \mathbf{x}\ s.t.\ x \in C^-(\mathbf{x}^\star)\}.$$

occurs. Combined with Tchakaloff's Theorem guarantees this shows that there exists at

least one set of n points $\mathbf{x}^\star$ such that $C^-(\mathbf{x}^\star) \neq \emptyset$ and therefore

$$\mathbb{P}(A) \geq \frac{\binom{n+1}{n}}{\binom{N}{n}} = \frac{n \cdot n!(N-n)!}{N!},$$

By independence, τ can be modelled by a geometric distribution with parameter $p = \mathbb{P}(A)$

$$\mathbb{P}(\tau = k) = (1-p)^{k-1}p$$

and the bounds for mean and variance follow. □

Not surprisingly, the worst case bound for $\mathbb{E}[\tau]$ is not practical, and it is easy to come up with examples where τ will be very large with high probability, e.g. a cluster of points plus one point far apart would result in wasteful oversampling of points from the cluster. Such examples suggest that one should be able to do better by taking the geometry of points into account which we will do in Section 2.3. However, before that, it is instructive to better understand the properties of Algorithm 1 when applied to empirical measures.

Application to empirical measures. Consider a random probability measure $\mu = \frac{1}{N}\sum_{i=1}^N \delta_{\left(f_1(X_i),\ldots,f_n(X_i)\right)}$ where the X_1,X_2,... are iid.

Proposition 2.8. *Let $N > n+1$ and let $f_1,\ldots,f_n$ be n real-valued functions and $X_1,\ldots,X_N$ be N i.i.d. copies of a random variable X. Set $F(X) = (f_1(X),\ldots,f_n(X))$, assume $\mathbb{E}[F(X)] = 0$ and denote*

$$E := \{0 \in \mathrm{Conv}\{F(X_i), i \in \{1,\ldots,N\}\}\}. \tag{2.2}$$

1. *$\mathbb{E}[\tau|E] \leq \frac{1}{p}$ and $Var(\tau|E) \leq \frac{1-p}{p^2}$, where*

$$p = \max\left\{\frac{n \cdot n!(N-n)!}{N!}, 1 - \mathbb{P}\left(0 \notin \mathrm{Conv}\{F(X_1),\ldots,F(X_{n+1})\}\right)^{N-n}\right\}, \tag{2.3}$$

2. *If the law of $F(X)$ is invariant under reflection in the origin, then*

$$\mathbb{P}\left(0 \notin \mathrm{Conv}\{F(X_1),\ldots,F(X_{n+1})\}\right) = 1 - 2^{-n},$$

3. For fixed n, as $N \to \infty$

$$\mathbb{P}\left(\begin{array}{c} \textit{for } n \textit{ uniformly at random chosen points } \mathbf{x}^\star \textit{ from } \mathbf{x}, \\ \exists x \in \mathbf{x} \ s.t. \ x \in C^-(\mathbf{x}^\star) \end{array}\right) \to 1,$$

where $\mathbf{x} = \{F(X_1), F(X_2), \ldots, F(X_N)\}$.

Proof. As in Proposition 2.7, the algorithm terminates when the event

$$A := \{\text{for } n \text{ uniformly at random chosen points } \mathbf{x}^\star \text{ from } \mathbf{x}, \exists x \in \mathbf{x} \ s.t. \ x \in C^-(\mathbf{x}^\star)\}$$

happens, where $\mathbf{x} = \{F(X_1), F(X_2), \ldots, F(X_N)\}$.
For item 1 denote $F_i := (f_1(X_{I_i}), \ldots, f_n(X_{I_i}))$ where $\{I_1, \ldots, I_N\}$ is a uniform shuffle of $\{1, \ldots, N\}$, i.e. a random permutation of its elements that makes every rearrangement equally probable, then

$$A = \{\exists i \in \{n+1, \ldots, N\} \text{ s.t. } F_i \in C^-(F_1, \ldots, F_n)\}$$

and note that

$$\mathbb{P}(A|E) = \frac{\mathbb{P}(E|A)\mathbb{P}(A)}{\mathbb{P}(E)} \geq \mathbb{P}(A)$$

since by Theorem 2.4 $\mathbb{P}(E|A) = 1$ and $\mathbb{P}(E) > 0$ since $N \geq n+1$ and $\mathbb{E}F(X) = 0$. The estimate of Proposition 2.7 $\mathbb{P}(E|0 \in \text{Conv}\{F_i\}) \geq \frac{n \cdot n!(N-n)!}{N!}$ is still valid, moreover

$$\begin{aligned} \mathbb{P}(A|E) \geq \mathbb{P}(A) =& \mathbb{P}(\exists i \in \{n+1, \ldots, N\} \text{ such that } F_i \in C^-(F_1, \ldots, F_n)) \\ =& 1 - \prod_{j=n+1}^{N} \mathbb{P}(F_j \notin C^-(F_1, \ldots, F_n)) \\ =& 1 - \mathbb{P}(F_{n+1} \notin C^-(F_1, \ldots, F_n))^{N-n} \\ =& 1 - \mathbb{P}(0 \notin \text{Conv}(F_1, \ldots, F_{n+1}))^{N-n} \end{aligned}$$

where the last equality follows from Theorem 2.4. We have therefore two different bounds

for $\mathbb{P}(A|E)$, so we can take the maximum, i.e.

$$\mathbb{P}(A|E) \geq \max\left\{\frac{n\cdot n!(N-n)!}{N!}, 1-\mathbb{P}\left(0\notin \text{Conv}\{F_1,\ldots,F_{n+1}\}\right)^{N-n}\right\}$$

Item 2. In Wendel [182] the author shows that when the F_i are distributed uniformly randomly on the unit sphere, then $\mathbb{P}(0\notin \text{Conv}(F_1,\ldots,F_{n+1}))^{N-n} = \left(1-2^{-n}\right)^{N-n}$. In Schneider and Weil [164, Theorem 8.2.1] it is shown the same result, for all the symmetric distributions with respect to 0.

Now τ can be modelled by a geometric distribution with parameter $p = \mathbb{P}(A|E)$, i.e. $\mathbb{P}(\tau = i) \geq (1-p)^{i-1}p$ and the mean and variance follows.

For item 3, it is enough to show $\mathbb{P}(A)\to 1$:

$$\mathbb{P}(A) = \mathbb{P}(A\Big|E)\times\mathbb{P}(E) + \mathbb{P}\left(A\Big|E^C\right)\times\mathbb{P}\left(E^C\right) = \mathbb{P}(A\Big|E)\times\mathbb{P}(E) + 0\times\mathbb{P}\left(E^C\right)\to 1\times 1,$$

as $N\to\infty$, where $\mathbb{P}(A|E^C)=0$ is due to Theorem 2.4, the convergence $\mathbb{P}(E)\to 1$ is guaranteed by Hayakawa [77] and the convergence $\mathbb{P}(A|E)\to 1$, is guaranteed by the proof of item 1 for fixed n. □

Proposition 2.8 shows us a "universal strategy" to explore the space of all the combination of points more efficiently, i.e. choosing the basis of the cone to maximize the probability placed on its inverse. In other words, ideally we should try to maximize

$$\max_{F_i\in\mathbf{x}} \mathbb{P}\left(F(X)\in C^-(F_1,\ldots,F_n)\right).$$

Not surprisingly, the worst case bounds of Algorithm 1 can be inconvenient, as Equation (2.3) shows. Nevertheless, item 2 of Proposition 2.8 shows an interesting trade-off in computational complexity, since the total cost

$$\mathbb{E}[\tau]O(n^3+Nn^2) \leq C(n^3+Nn^2)\min\left\{\frac{N!}{n\cdot n!(N-n)!}, \frac{1}{1-(1-2^{-n})^{N-n}}\right\} \quad (2.4)$$

has a local minimum in N, see Figure 2.2.

Section 5.5 shows that this is observed in experiments and this minimum also motivates

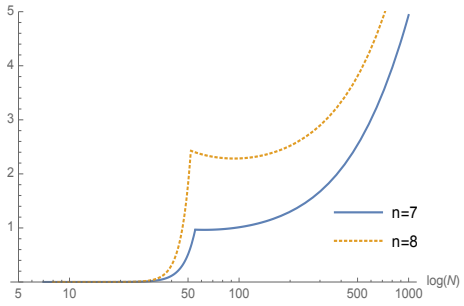


Figure 2.2: The plots shows the logplot of Inequality (2.4), i.e. the complexity of Algorithm 1 in case of symmetric distributions.

the divide and conquer strategy we use in the next Section.

Proposition 2.8 conditions on the event (2.2) so that the recombination problem is well-posed, but as intuition suggests this happens with probability one for large enough N, see Hayakawa [77].

Sampling from empirical measures. Often we do not know the distribution ϕ of the points or $\mathbb{E}[F(X)]$, moreover it could be that for the realized samples $\{F_i\}_{i=1}^N$, $\mathbb{E}[F(X)] \notin \text{Conv}\{F_i\}_{i=1}^N$. In these cases, due to Theorem 2.2, and since Algorithm 1 is based on Theorem 2.4, which assumes that the barycentre of the points given is 0, the input of the Algorithm is not the collection $\{F_i\}_{i=1}^N$, but

$$\hat{F}_i = (f_1(X_i), \dots, f_n(X_i)) - \left(\sum_{j=1}^N f_1(X_j) w_j, \dots, \sum_{j=1}^N f_n(X_j) w_j \right),$$

in this way we are sure that the barycentre is 0 and $0 \in \text{Conv}\{\hat{F}_i\}_{i=1}^N$, and the hypothesis of both Theorem 2.2 and 2.4 are satisfied. Unfortunately, the $\{\hat{F}_i\}_{i=1}^N$ are not independent, which leads to an impossible analysis, even though the correlation between the $\hat{F}_i$ decreases when N becomes bigger and tends to 0. Thus, we can believe that the analysis of the proof of Proposition 2.8 is a good approximation of the complexity of the Algorithm 1, when the $\{\hat{F}_i\}_{i=1}^N$ are given as input and N is big "enough", as it is shown in Figure 2.2 and Figure 2.4 in case of symmetric distribution.

It is relevant to note that we can always consider uniform measures, i.e. $\mu = \frac{1}{N} \sum_{i=1}^N \delta_{x_i}$,

modifying the support of the measure, and then eventually go back to the original (not-uniform) measure.

Lemma 2.9. *Let us consider a set* $\mathbf{x} = \{x_i\}_{i=1}^N$ *in* $\mathbb{R}^n$ *and a sequence* $\{\kappa_i\}_{i=1}^N$ *of strictly positive numbers. There exists a measure* μ *on* $\mathbf{x}$ *such that* $\mu(\mathbf{x}) = 0$ *if and only if there exists a measure* $\mu^\star$ *on* $\{\frac{x_i}{\kappa_i}\}_{i=1}^N$ *such that* $\mu^*(\{\frac{x_i}{\kappa_i}\}_{i=1}^N) = 0$.

Proof. Let us assume that there exists μ on $\mathbf{x}$ such that $\mu(\mathbf{x}) = 0$, and let us call $\mu_i := \mu(x_i)$. It is enough to define $\mu^\star = \mu_i\kappa_i$. The other side of the equivalence is proved in the same way. □

Remark 2.10. *Lemma 2.9 is a consequence of the fact that a cone is defined only by the directions of the vectors of the "basis", and not from their length.*

2.3 A geometrically greedy Algorithm

Algorithm 1 produces candidates for the reduced measure by random sampling and then accepts or rejects them via the characterization given in Theorem 2.4. We now optimize the first part of it, namely the selection of candidates, by exploiting the geometry of cones.

Motivation in two dimensions. Having chosen $\mathbf{x}^\star$ points we know that we have found a solution if $C^-(\mathbf{x}^\star) \cap \mathbf{x} \neq \emptyset$. Hence, maximizing the volume of the cone increases the chance that this intersection is not empty. Indeed, when $n = 2$ it is easy to show the following result.

Theorem 2.11. *Let* $\mathbf{x}$ *be a set of* $N \geq 3$ *points in* $\mathbb{R}^2$ *and* $x_1 \in \mathbf{x}$. *Define* $\mathbf{x}^\star = (x_1^\star, x_2^\star)$, *where*

$$x_2^\star := \arg\max_{x \in \mathbf{x} \setminus \{x_1^\star\}} \left| \frac{\langle x_1^\star, x \rangle}{\|x_1^\star\| \|x\|} - 1 \right|. \tag{2.5}$$

There exists a convex combination $\sum_{x \in \mathbf{x}} w_x x$ *of* $\mathbf{x}$ *that equals* 0 *if and only if* $\mathbf{x} \cap C^-(\mathbf{x}^\star) \neq \emptyset$.

Theorem 2.11 follows immediately from the cone geometry, but it is instructive to spell it out since it motivates the case of general n.

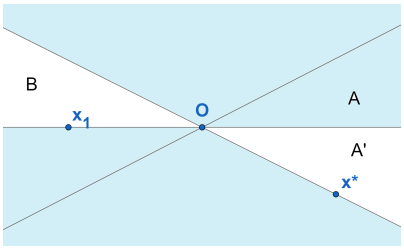


Figure 2.3: Proof of Theorem 2.11.

Proof. The shaded areas in Figure 2.3 indicate $C^-(x_1,x^\star)$ (on the top) and $C(x_1,x^\star)$ (on the bottom). Moreover, by definition of $x^\star$ in the region A and A' there are no points.
($\Rightarrow$) If there exists a point x_2 in $C^-(x_1,x^\star)$, then by convexity it follows that there exists a convex combination of 0 for $x_1, x_2, x^\star$.
($\Leftarrow$) If there does not exist a point in $C^-(x_1,x^\star)$, then there are points only in $B \cup C(x_1,x^\star)$, therefore, again for simple convex geometry arguments, it is impossible that there exists a convex combination of 0 for $x_1, x_2, x^\star$. □

If we modify Algorithm 1 by selecting in step 4 the new point by maximizing the angle according to Equation (2.5) then for $n = 2$, Theorem 2.11 guarantees that $n+1 = 3$ points out of N are found that constitute a reduced measure $\hat{\mu}$ in $\tau \leq 2$ computational steps.

A geometrically greedy Algorithm. For general dimensions n, the intuition remains that a good (but not perfect) proxy to maximize the likelihood that the negative cone is non-empty, is given by maximizing the volume of the cone, see [164, Chapter 8]. Such a volume maximization is a long-standing open question and goes at least back to Sommerville [168]; see Bollobás [21] for an overview. All this motivates the "*geometrically greedy*" Algorithm 2 that applies for any n. First note that the deletion of points in step 5 in Algorithm 2 does not throw away potential solutions: suppose we have deleted a point $\hat{x}$ in the previous step that was part of a solution, i.e. there exists a set of $n+1$ points $\hat{\mathbf{x}}^\star$ in $\mathbf{x}$ such that $\hat{x} \in \hat{\mathbf{x}}^\star$, and there exist $n+1$ weights $\hat{w}_i$, $i \in [0,\ldots,n+1]$, $\hat{w}_i \in [0,1]$, $\sum w_i = 1$ such that $\sum_{i=1}^{n+1} \hat{w}_i \hat{x}_i^* = 0$, $\hat{x}_i^* \in \hat{\mathbf{x}}^\star$. If we indicate with $\mathbf{x}_c$ the n vectors of the basis of the cone of the previous iteration, we know that $\hat{x} \in \text{interior}\{C(\mathbf{x}_c)\}$, which means there exist strictly

Algorithm 2 Optimized measure reduction algorithm

1: **procedure** REDUCE-OPTIMIZED(A set $\mathbf{x}$ of N points in $\mathbb{R}^n$)
2: Choose n points $\mathbf{x}^\star = \{x_1^\star, \ldots, x_n^\star\}$ from $\mathbf{x}$
3: $i \leftarrow 0$
4: **while** $C^-(\mathbf{x}^\star) \cap \mathbf{x} = \emptyset$ **do**
5: $\mathbf{x} \leftarrow \mathbf{x} \setminus \text{interior}\{C(\mathbf{x}^\star)\}$
6: **if** $i = 0$ **then**
7: $x_{i+1}^\star \leftarrow \arg\max_{x \in \mathbf{x} \setminus \mathbf{x}^\star} \left| \langle x, \sum_{j=2}^n x_j^\star \rangle - 1 \right|$
8: **else**
9: $x_{i+1}^\star \leftarrow \arg\max_{x \in \mathbf{x} \setminus \mathbf{x}^\star} \left| \langle x, \sum_{j=1}^i x_j^\star \rangle - 1 \right|$
10: **end if**
11: $i \leftarrow ((i+1) \mod n)$
12: **end while**
13: $\mathbf{x}^\star \leftarrow \mathbf{x}^\star \cup x^\star$ with an arbitrary $x^\star \in C^-(\mathbf{x}^\star) \cap \mathbf{x}$
14: Solve the linear system $\sum_{x \in \mathbf{x}^\star} w_x^\star x = 0$ for $w^\star = (w_x^\star)_{x \in \mathbf{x}^\star}$
15: **return** $(\mathbf{x}^\star, w^\star)$
16: **end procedure**

positive values c_i such that $\hat{x} = \sum_{i=1}^n c_i x_i^c$, $x_i^c \in \mathbf{x}^c$. Therefore,

$$\sum_{i=1}^{n+1} \hat{w}_i \hat{x}_i^* = \hat{w}_{n+1}\hat{x} + \sum_{i=1}^{n} \hat{w}_i \hat{x}_i^* = \hat{w}_{n+1} \sum_{i=1}^{n} c_i x_i^c + \sum_{i=1}^{n} \hat{w}_i \hat{x}_i^*, \ x_i^c \in \mathbf{x}^c \text{ and } \hat{x}_i^* \in \hat{\mathbf{x}}^\star.$$

Given that w_i and c_i are positive $0 \in \text{Conv}\{\hat{\mathbf{x}}^\star \cup \mathbf{x}_c\}$ and we can apply again the Carathéodory's Theorem 2.2, which implies that the deleted point $\hat{x}$ was not essential. The reason for the if clause in step 6 is simply that the first time the loop is entered we optimize using the randomly selected bases, but in subsequent runs it is intuitive that we should only optimize over the base points that were optimized in previous loops. We now discuss the complexity of Algorithm 2.

Proposition 2.12. *The complexity of Algorithm 2 to compute a reduced measure $\hat{\mu}$,as in Tchakalof Theorem 2.1, is*

$$O(n^3 + n^2 N) + (\tau - 1)O(n^2 + nN),$$

here $\tau = \inf\{i \geq 1 : C^-(\mathbf{X}_i) \cap \mathbf{x} \neq \emptyset\}$ where $\mathbf{X}_1, \mathbf{X}_2, \ldots$ are obtained as in Algorithm 2.

Proof. The most expensive steps are the same ones as in Algorithm 1 and in addition the maximization problem. After the first step, given that we update one point of the basis

at time, we can be more efficient using the Sherman–Morrison formula. Let us call $\mathbf{X}_t^\star$ the matrix whose rows are the vectors of the basis at the step t $\mathbf{x}_t^\star$, therefore we have $A_t = ((\mathbf{X}_t^\star)^\top)^{-1}$, thus (shifting properly the vectors)

$$A_{t+1} = \left((\mathbf{X}_t^\star)^\top + (X^\star - X_1^\star) \cdot e_1^\top\right)^{-1} = A_t - \frac{A_t(X^\star - X_1^\star)e_1^\top A_t}{1 + e_1^\top A_t(X^\star - X_1^\star)},$$

in the case we want to substitute the "first" vector of the basis $X_1^\star$ with the vector $X^\star$. Let us note that the only multiplications to be computed are $A_t(X^\star - X_1^\star)$ and $\left[A_t(X^\star - X_1^\star)\right] \cdot \left[e_1^\top A_t\right]$, which are done in $O(n^2)$ operations. To check if there are points inside the cone (or the inverse cone), we multiply the matrix A times the matrix $\mathbf{X}$ (of all the remaining vectors $\mathbf{X}$), and again after the first step costs $O(Nn^2)$, we can use the Sherman–Morrison formula as before and obtain

$$A_{t+1}\mathbf{X}^\top = A_t\mathbf{X}^\top - \frac{A_t(X^\star - X_1^\star)e_1^\top A_t\mathbf{X}^\top}{1 + e_1^\top A_t(X^\star - X_1^\star)}.$$

Let us note that we have already computed $A_t\mathbf{X}^\top$ at the previous step, $A_t(X^\star - X_1^\star)$ to compute A_{t+1}, therefore the only cost is to compute $\left[A_t(X^\star - X_1^\star)\right] \cdot \left[e_1^\top A_t\mathbf{X}^\top\right]$, which is done in $O(nN)$ operations.

The previous computations show us that after the first step, updating one element at a time, improves the computational efficiency of the successive steps of a factor n. Let us now tackle the maximization problem, it requires to compute the norm, i.e. $O(Nn^2)$ operations, which could be done only once. Moreover, the maximization problem requires to compute (part of) the sum of the vectors in $\mathbf{x}^\star$ and then the scalar product, which require $O(Nn)$. The last expensive operation we should consider is due to solve the last system to find the weights. The system we want to solve is

$$\begin{pmatrix} (\mathbf{X}^\star)^\top & X^\star \\ \mathbf{1} & 1 \end{pmatrix} w = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix},$$

where $\mathbf{0}$ is a $n \times 1$ vector of 0, and $\mathbf{1}$ is a $1 \times n$ vector of 1. Using again the Sherman–Morrison

formula, since we have already computed $A = (\mathbf{X}^\star)^{-1}$ the weights w_i can be computed as

$$w = \begin{pmatrix} (\mathbf{X}^\star)^\top & X^\star \\ \mathbf{1} & 1 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} = \begin{pmatrix} A^\top + A^\top X^\star c^{-1} \mathbf{1} A^\top, & -A^\top X^\star c^{-1} \\ -c^{-1} \mathbf{1} A^\top, & c^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}$$
$$= -\frac{1}{c} \begin{pmatrix} A^\top X^\star \\ 1 \end{pmatrix},$$

where $c = 1 - \mathbf{1} A^\top X^\star$ is a number. In this way we need $O(n^2)$ operations, not $O(n^3)$, i.e. the complexity of solving a linear system.
The total cost therefore is $O(n^3 + n^2 N) + (\kappa - 1) O(n^2 + nN)$. □

Remark 2.13. *The gain in the computational cost we obtain using the Sherman–Morrison formula has a cost in term of numerical stability.*

In contrast, to the complexity of Algorithm 1, Corollary 2.6, the n^3 term that results from a matrix inversion is no longer proportional to τ, and the random runtime τ only affects the complexity proportional to $n^2 + nN$. For a generalization of Theorem 2.11 from $n = 2$ to general n, that is a statement of the form "in n dimensions the algorithm terminates after at most $\tau \leq f(n)$", one ultimately runs into the result of a "positive basis" from discrete geometry, see for example Regis [153], that says if $n \geq 3$ it is possible to build positive independent set of vectors of any cardinality. Characterizing the probability of the occurrences of such sets is an ongoing discrete geometry research topic and we have nothing new to contribute to this. However, despite the existence of such "positive independent sets" for $n \geq 3$, the experiments in Section 5.5 underlines the intuition that in the generic case, maximizing the angles is hugely beneficial also in higher dimension. If a deterministic bound on the runtime is crucial, one can combine the strengths of Algorithm 2 (a good chance of finding $\mathbf{x}^\star$ quickly by repeatedly smart guessing) with the strength of deterministic algorithms such as Litterer and Lyons [119], Maalouf et al. [126], Tchernychova [177] by running Algorithm 2 for at most k steps and if a solution is not found run a deterministic algorithm. Indeed, our experiments show that this is on average a very good trade-off since most of the probability mass of τ is concentrated at small k, see also Appendix 2.B.

Robustness. An interesting question is how robust the measure reduction procedure is to the initial points. Therefore assume we know the solution of the recombination problem (RP) for $\mathbf{x} \subset \mathbb{R}^n$, i.e. a subset of $n+1$ points $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{n+1}) \subset \mathbf{x}$ and a discrete measure $\hat{\mu}$ on $\hat{\mathbf{x}}$ such that $\hat{\mu}(\hat{\mathbf{x}}) = 0$. If a set of points $\mathbf{y}$ is "close" to $\mathbf{x}$, one would expect that, to find the solution $\hat{\mathbf{y}} \subset \mathbf{y}$ of the RP for $\mathbf{y}$, looking for points "close" to $\hat{\mathbf{x}}$ would be useful. The theorem below uses the Woodbury matrix identity to make this precise.

Proposition 2.14. *Assume that* $\operatorname{span}(\hat{\mathbf{x}}) = \operatorname{span}(\hat{\mathbf{x}}_{-1}) = \mathbb{R}^n$*, where* $\hat{\mathbf{x}}_{-i} := \hat{\mathbf{x}} \setminus \hat{x}_i$*. Denote with* $\mathbf{X}$ *and* $\mathbf{Y}$ *the matrices having as rows the vectors respectively in* $\mathbf{x}$ *and* $\mathbf{y}$*. Let* R *be an invertible matrix and* $E := \mathbf{X} - \mathbf{Y}R$*, such that* $\mathbf{X} = \mathbf{Y}R + E$*. Denote* $\gamma_1 := A_1 \hat{X}_1^\top$*, where* $\hat{\mathbf{x}}$ *is a solution to the RP* $\mathbf{x}$ *and* $A_1 = (\hat{\mathbf{X}}_{-1}^\top)^{-1}$*. Assuming that the inverse matrices exist,* $\hat{\mathbf{Y}} := \hat{\mathbf{X}}R + E_{\hat{\mathbf{x}}}$ *is a solution to the RP* $\mathbf{y}$ *if and only if*

$$\gamma_1^\top + E_{\hat{x}_1} R^{-1} A_1^\top \leq \left(\gamma_1^\top + E_{\hat{x}_1} R^{-1} A_1^\top\right) E_{\hat{\mathbf{x}}_{-1}} \left(I + R^{-1} A_1^\top E_{\hat{\mathbf{x}}_{-1}}\right) R^{-1} A_1^T \tag{2.6}$$

where E_a *indicates the part of the matrix* E *related to the set of vectors* $a \subset \mathbf{x}$*.*

Note that if $E = 0$ the previous inequality is true due to Theorem 2.4, which implies the "robustness" with respect to rotations, as one would expect from geometrical reasoning. Thus, heuristically we would prefer R to be (possibly one of) the invertible matrix that minimizes the norm of $E_{\hat{\mathbf{x}}}$. Also, it is important to remark that γ_1 is independent of the relation between $\mathbf{Y}$ and $\mathbf{X}$.

For example, considering the dimension equal to 2, $R = I$ and $A_1 = I$ - which implies $\{\hat{X}_2, \hat{X}_3\} = \{e_1, e_2\}$ and that $\hat{X}_1$ is in the third quadrant - Equation (2.6) becomes $\hat{X}_1 + E_{\hat{x}_1} \leq \left(\hat{X}_1 + E_{\hat{x}_1}\right) E_{\hat{\mathbf{x}}_{-1}} \left(I + E_{\hat{\mathbf{x}}_{-1}}\right)$, which quantifies how much we can perturbate (by means of $E_{\hat{\mathbf{x}}}$) the points $\{e_1, e_2, \hat{X}_1\}$ to still have the origin inside the convex hull of $\hat{\mathbf{X}} + E_{\hat{\mathbf{x}}}$.

Proof. From Theorem 2.4, we know that $\hat{\mathbf{X}}R + E_{\hat{\mathbf{x}}}$ is a solution if and only if

$$\left(\left(\hat{\mathbf{X}}_{-1}R + E_{\hat{\mathbf{x}}_{-1}}\right)^\top\right)^{-1} \left(\hat{X}_1 R + E_{\hat{x}_1}\right)^\top \leq 0.$$

Let us note that the last product is a vector, therefore we can study the transpose and

using the Woodbury matrix identity we have that

$$\begin{aligned}\left(\hat{X}_1 R + E_{\hat{x}_1}\right)\left(\hat{\mathbf{X}}_{-1} R + E_{\hat{\mathbf{x}}_{-1}}\right)^{-1} &= \\ &= \left(\hat{X}_1 + E_{\hat{x}_1} R^{-1}\right)\left(\hat{\mathbf{X}}_{-1} + E_{\hat{\mathbf{x}}_{-1}} R^{-1}\right)^{-1} \\ &= \left(\hat{X}_1 + E_{\hat{x}_1} R^{-1}\right)\left(I - A_1^T E_{\hat{\mathbf{x}}_{-1}}\left(I + R^{-1} A_1^T E_{\hat{\mathbf{x}}_{-1}}\right) R^{-1}\right) A_1^T.\end{aligned}$$

Setting the last equation less or equal than 0 shows the result. □

This is not only of theoretical interest, since the initial choice of a cone basis in Algorithm 2 can be decisive. For example, if we repeatedly solve a "similar" RP, e.g. N points are sampled from the same distribution, then Proposition 2.14 suggests that after the first RP solution, we should use the points in the new set of points that are closest to the previous solution as initial choice of the cone basis.

Las Vegas resets. The only source of randomness in Algorithm 2 is the choice of $\mathbf{x}^\star$ in step 2. If this happens to be a bad choice, much computational time is wasted, or even worse, the Algorithm might not even terminate. However, like any other random "black box" algorithm one can stop Algorithm 2 if τ becomes large and then restart with a random basis $\mathbf{x}^\star$ sampled independently from the previous one. We call a sequence $\mathcal{S} = (t_1, t_2, \ldots)$ of integers a reset strategy where the ith entry t_i denotes the number of iterations we allow after the ith time the algorithm was restarted, e.g. $\mathcal{S} = (10,3,6,\ldots)$ means that if a solution is not found after 10 loops, we stop and restart; then wait for at most 3 loop iterations before restarting; then 6, etc. Surprisingly, the question which strategy $\mathcal{S}$ minimises the expected run time of Algorithm 2 has an explicit answer. A direct consquence of the main result in Luby et al. [123] is that $\mathcal{S}^\star = c \times (1,1,2,1,1,2,4,1,1,2,1,1,2,4,8,1\ldots)$ achieves the best expected running time up to a logarithmic factor, i.e. by following $\mathcal{S}^\star$ the expected running time is $O(\mathbb{E}[\tau^\star]\log\mathbb{E}[\tau^\star])$ where $\tau^\star$ denotes the minimal expected run time under *any* reset strategy. Thus, although in general we do not know the optimal reset strategy $\tau^\star$ (which will be highly dependent on the points $\mathbf{x}$), we can still follow $\mathcal{S}^\star$ which guarantees that Algorithm 2 terminates and that its expected running time is within a logarithmic factor of the best reset strategy. Since Algorithm 2 uses n updates of a cone basis, it is

natural to take c proportional to n (in our experiments we fixed throughout $c = 2n$).

Divide and conquer. A strategy used in existing algorithms Litterer and Lyons [119], Maalouf et al. [126], Tchernychova [177] is for a given $\mu = \sum_{i=1}^{N} w_i x_i$ to partition the N points into $2(n+1)$ groups $I_1, \ldots, I_{2(n+1)}$ of approximately equal size, compute the barycenter b_i for each group I_i, and then carry out any measure reduction algorithm to reduce the support of $\sum_{i=1}^{2(n+1)} \frac{w_i}{\sum_{j \in I_i} w_j} \delta_{b_i}$ to $n+1$ points. One can then repeat this procedure, see [119, Algorithm 2 on p1306] for details, and since each iteration halves the support, this terminates after $\log(N/n)$ iterations. Hence the total cost is $O(Nn + \log(N/n))C(2(n+1),n+1))$, where $C(2(n+1),n+1)$ denotes the cost to reduce the measure from $2(n+1)$ points to $n+1$ points. For the algorithm in [119] $C(2(n+1),n+1) = O(n^4)$, similarly to [126]; for the algorithm in [177] $C(2(n+1),n+1) = O(n^3)$. Similarly, we can also run our randomized Algorithm 2 on the $2(n+1)$ points. However, the situation is more subtle since the regime where Algorithm 2 excels is the big data regime $N \gg n$, so running the algorithm on smaller groups of points could reduce the effectiveness. Already for simple distributions and Algorithm 1, as in Proposition 2.8 item 2, we see that for the optimal choice we should divide the points in N_n^* groups with N_n^* denoting the argmin of the complexity in N. This decreases the computational cost to $O(Nn + \log_{N_n^*/n}(N)\bar{C}(N_n^*,n+1))$, where $\bar{C}$ denotes the computational cost of Algorithm 1 to reduce from N_n^* points to $(n+1)$ points. In general, the optimal N_n^* will depend on the distribution of the points, but an informal calculation shows that $N_n^* = 50(n+1)$ achieves the best trade-off; see Appendix 2.A.

2.4 Experiments

We give two sets of experiments[1] to demonstrate the properties of Algorithm 1 and 2:

(i) using synthetic data allows to study the behaviour in various regimes of N and n;

(ii) on real-world data, following Maalouf et al. [126], for fast least square solvers.

As baselines we consider two recent algorithms, Tchernychova [177] (*det3*) and Litterer

[1] All experiments have been run on a MacBook Pro, CPU: i7-7920HQ, RAM: 16 GB, 2133 MHz LPDDR3.

and Lyons [119] resp. Maalouf et al. [126] (*det4*)[2].

Reducing empirical measures. We sampled $N \in \{2n, 50, \ldots, 10^6\}$ points from

(i) a $n = 15$-dimensional standard normal random variable (*symmetric1*),

(ii) a $n = 20$-dimensional standard normal random variable (*symmetric2*),

(iii) $n = 20$ dimensional mixture of exponential (*non symmetric*).

We then ran Algorithm 1 (*basic*), Algorithm 2 (*optimized*), as well as Algorithm 2 with the optimal Las Vegas reset (*optimized-reset*) and the divide and conquer strategy (*log-opt*); the results are shown in Figure 2.4. The first row clearly shows that the performance gets best in the big data regime $N \gg n$. The biggest effect is how the angle/volume optimization of Algorithm 2 drastically reduces the number of iterations compared to Algorithm 1, and therefore the running time and the variance. From a theoretical perspective, it is interesting that the shape predicted in Proposition 2.8, for symmetric distributions such as the normal distribution (the two columns on the left) also manifests itself for the non-symmetric mixture (the right column); see also Figure 2.2.The Las Vegas reset strategy is only noticeable in the regime when simultaneously N and n are close and relatively large; e.g. Figure 2.4 falls not in this regime which is the reason why the plots are indistinguishable. Nevertheless, even in regimes such as in Figure 2.4, the reset strategy is at least on a theoretical level useful: it guarantees the convergence of Algorithm 2 by excluding pathological cases of cycling through a "sequence" of cone bases (although we have not witnessed such pathological cases in our experiments).

As expected, the regime when the number of points N is much higher than the number of dimensions n, yields the best performance for the randomized algorithm; moreover, Figure 2.5 shows that the run time is approximately $O(n)$ (in contrast to the runtime of *det4* and *det3* that is $O(n^4)$ resp. $O(n^3)$).

[2]Although the derivation is different, the resulting algorithms in Litterer and Lyons [119], Maalouf et al. [126] are essentially identical; we use the implementation of Maalouf et al. [126] since we do the same least mean square experiment as in Maalouf et al. [126].

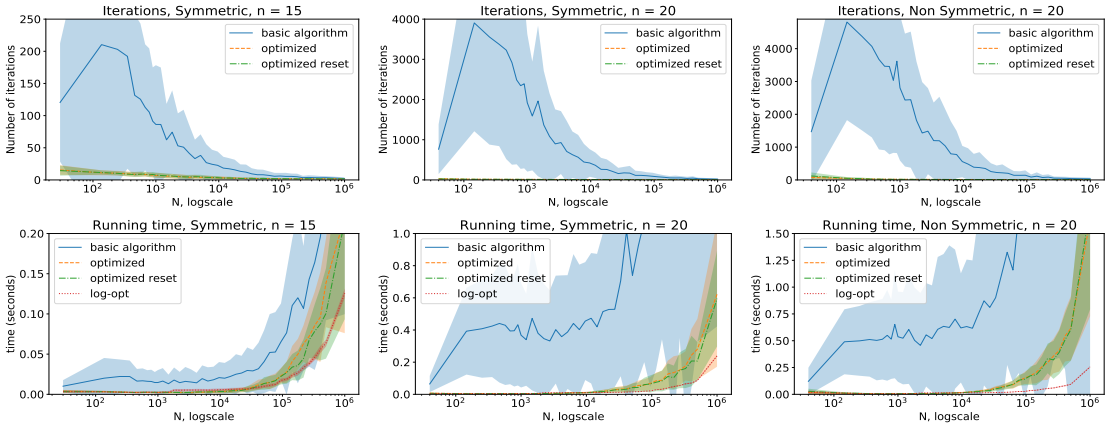


Figure 2.4: Running time and number of iterations of the randomized algorithms as N varies. The first two columns show the results for *symmetric1* and *symmetric2* , the right column for *non-symmetric*. The shaded area represents the standard deviation (from 70 repetitions of the experiment).

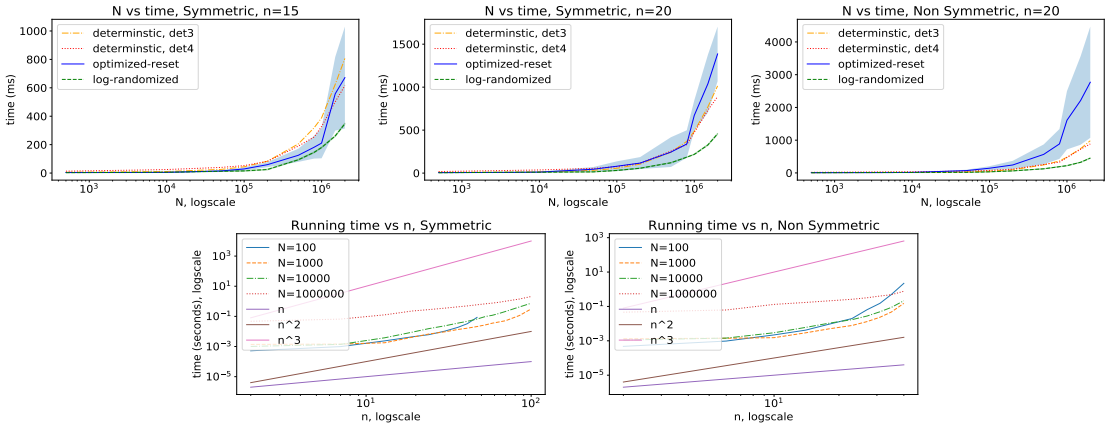


Figure 2.5: The top row compares the running time of randomized Algorithms against deterministic algorithms as N varies for *symmetric1* (left), *symmetric2* (middle) and *non-symmetric* (right). The shaded area represents the standard deviation (from 70 repetitions of the experiment). The bottom row running time of the *log-optimized* Algorithm as n varies for different N (average of 70 samples).

Fast mean-square solvers. In Maalouf et al. [126] a measure reduction was used to accelerate the least squares method, i.e. the solution of the minimization problem $\min_w \|\mathbf{X}w - \mathbf{Y}\|^2$ where $\mathbf{X} \in \mathbb{R}^{N\times d}$ and $\mathbf{Y} \in \mathbb{R}^N$; as in Proposition 2.14, $\mathbf{X}$ denotes a matrix which has as row vectors the elements of $\mathbf{x}$, similarly for $\mathbf{Y}$. Sometimes precise solutions are required and in this case, the measure reduction approach yields a scalable method: Theorem 2.2 guarantees the existence of a subset of $n+1 = (d+1)(d+2)/2+1$ points $(\mathbf{x}^\star,\mathbf{y}^\star)$ of $\mathbf{x}$ and $\mathbf{y}$ such that $(\mathbf{X}|\mathbf{Y})^\top \cdot (\mathbf{X}|\mathbf{Y}) = (\mathbf{X}^\star|\mathbf{Y}^\star)^\top \cdot (\mathbf{X}^\star|\mathbf{Y}^\star)$ where we

denote with $(\mathbf{X}|\mathbf{Y})$ the element of $\mathbb{R}^{N\times(d+1)}$ formed by adding $\mathbf{Y}$ as a column. However, this implies that $\|\mathbf{X}w - \mathbf{Y}\|^2 = \|\mathbf{X}^\star w - \mathbf{Y}^\star\|^2$ for every w, hence it is sufficient to solve the least square problem in much lower dimensions once $\mathbf{X}^\star$ and $\mathbf{Y}^\star$ have been found. We use the following datasets from Maalouf et al. [126]

(i) 3D Road Network [8] that contains 434 874 records and use the two attributes longitude and latitude to predict height,

(ii) Household power consumption [9] that contains 2 075 259 records and use the two 2 attributes active and reactive power to predict voltage.

(iii) We also add a synthetic dataset, where $\mathbf{X} \in \mathbb{R}^{N\times n}$, $\theta \in \mathbb{R}^n$ and $\epsilon \in \mathbb{R}^N$ are normal random variables, and $\mathbf{Y} = \mathbf{X}\theta + \epsilon$ which allows to study various regimes of N and n.

Figure 2.6 shows the performance of Algorithm 2 with the Las Vegas reset, with the Las Vegas reset and the divide and conquer optimization on the datasets *(i),(ii)*. We observe that already Algorithm 2 with Las Vegas resets is on average faster but the running time distribution has outliers where the algorithm takes longer than for the deterministic run time algorithms; combined with divide and conquer the variance is reduced by a lot. Figure 2.7 shows the results on the synthetic dataset *(iii)* for various values of N and $n = (d+1)(d+2)/2$.

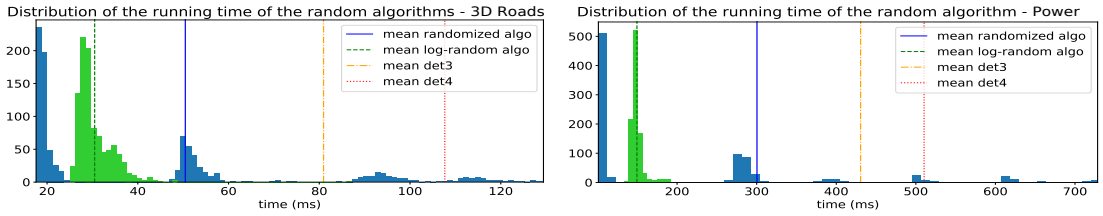


Figure 2.6: Histogram of the running time of Algorithm 2 with a reset strategy and of the "divide and conquer" variation algorithm (*log-random*). The vast majority of probability mass of the random runtime is below any of runtimes of the deterministic algorithms although with small probability it can take longer than the deterministic runtimes.

Breaking the randomized Algorithm. As the above experiments show, Algorithm 2 can lead to big speed ups. However, it is not uniformly better than the deterministic algorithms, and there are situations where one should not use it: firstly, Algorithm 2 was

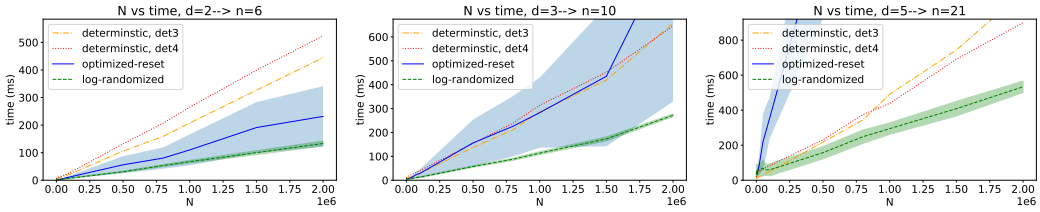


Figure 2.7: Performance of the different algorithms on the synthetic data set *(iii)* for various values of N and $n = (d+1)(d+2)/2$.

optimized to work well in the $N \gg n$ regime and while this is an important regime for data science, the recombination problem itself is of interest also in other regimes, [119]. Secondly, the Las Vegas resets give a finite running time, but it is easy to construct examples where this can be much worse than the deterministic algorithms. Arguably the most practically relevant issue is when the independence hypothesis of Theorem 2.4 is not satisfied. This can appear in data sets with a high number of highly correlated categorical features, such as [94]. This can be overcome by using the Weyl Theorem, see Remark 2.5 but the computational cost is higher than computing the inverse of the cone basis (A in Theorem 2.4) and the benefits would be marginal, if not annulled, compared to the deterministic algorithms. More relevant is that Algorithm 2 can be easily combined with any of the deterministic algorithms to build an algorithm that has a worst case run time of the same order as a deterministic one but has a good chance of being faster; see Appendix 2.B for details and experiments.

Implementation and benchmarking. For *det4* we have used the code provided by the authors of Maalouf et al. [126] available at the repository[3] in Python; for *det3* the code of Tchernychova [177] has not been written in Python, therefore we have implemented it to allow for a fair comparison using standard Numpy libraries. We used throughout the same codeblocks for the Divide and Conquer strategy in all the implementations. We did not use the tree data-structures presented in Tchernychova [177] since it does not change complexity bounds and is independent of the reduction procedure itself; however, it could be also used for *det4* and our log-randomized algorithms. The code is available in the

[3] github.com/ibramjub/Fast-and-Accurate-Least-Mean-Squares-Solvers

public repository[4].

2.5 Summary

We introduced a randomized algorithm that reduces the support of a discrete measure supported on N atoms down to $n+1$ atoms while preserving the statistics as captured with n functions. The key was a characterization of the barycenter in terms of negative cones, that inspired a greedy sampling. Motivated by the geometry of cones this greedy sampling can be optimized, and finally combined with optimization methods for randomized algorithms. This yields a "greedy geometric sampling" that follows a very different strategy than the previous deterministic algorithms, and that performs very well in the big data regime when $N \gg n$ as is often the case for large sample sizes common in inference tasks such as least square solvers.

[4] github.com/FraCose/Recombination

2.A Appendix - Divide and conquer, choice of the subgroup size

As mentioned in Section 2.3, to apply a divide and conquer strategy requires to balance the size of subgroups against the property of Algorithm 2 to exploit a large number of points as to maximize the likelihood of points being in the (inverse) cone. Let us explain how we have chosen $N_n^* = 50(n+1)$, which should be thought as linear approximation of the exact minimum for the complexity of Algorithm 2. Therefore first note, that as Figure 2.8 shows,

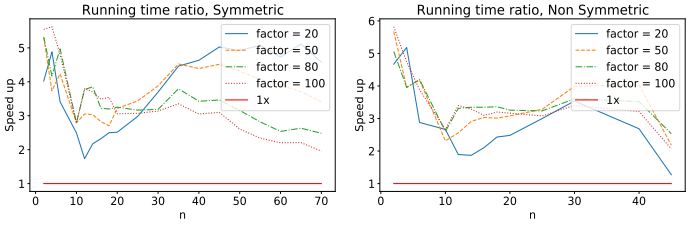


Figure 2.8: Bottom-right: it is shown how much the *optimized-reset* algorithm is faster than *det4* in case $N = (n+1) \times \text{factor}$.

choosing any number between 20 and 80, in place of 50, has similar effects if $n < 70$ in the case of symmetric distribution, whilst the same holds in the case of mixture of exponentials (non symmetric) if $n \leq 40$. We think that this effect is due to the fact that "experimentally" there exists a long plateau in the running time of the optimized Algorithms with and without reset, see Figure 2.4. Therefore, let us suppose that there exist k,K s.t. for any $k(n+1) \leq N^{(1)}$,$N^{(2)} \leq K(n+1)$ and $n < 40$, then $\bar{C}(N^{(1)},n+1) \approx \bar{C}(N^{(2)},n+1)$, where $\bar{C}(\cdot,n+1)$ is the computational cost to reduce $\cdot$ number of points in $\mathbb{R}^n$ using Algorithm 2. Moreover, we suppose that the $\arg\min_x \bar{C}(x,n+1) \in [k(n+1),K(n+1)]$. The previous two conditions are equivalent to the presence of the plateau in Figure 2.4, in correspondence with the minimum value of the running time, for the optimized Algorithms with and without reset. Under these assumptions, the best choice would be $N_n^* = K(n+1)$. Without knowing the value of K, however we can estimate the difference into the complexity for different group subdivisions: if we have $N \gg K(n+1)$ number of points, using the "divide and conquer" paradigm with $N^{(i)}$ groups, we can build two algorithms s.t. the difference

of the computational costs is

$$O\Big(Nn+\log_{N^{(1)}/n}(N/n)C(N^{(1)},n+1)\Big)-O\left(Nn+\log_{N^{(2)}/n}(N/n)C(N^{(2)},n+1)\right)\approx$$
$$\approx \bar{C}(Kn,n+1)\log_{N^{(2)}/n}(N/n)\left(\frac{1}{\log_{N^{(2)}/n}N^{(1)}/n}-1\right). \tag{2.7}$$

Therefore, we have that the difference depends on a factor $|1/\log_{N^{(2)}/n}(N^{(1)}/n)-1|$. Given Figure 2.8, we have estimated approximately $k=20$, $K=80$ for the symmetric case, thus as a rule of thumb we assume that $N_n^*=50(n+1)$ is a reasonable value, and in view of Equation (2.7) we can say that changing slightly 50 the running time would remain stable. The analogous argument can be made for the mixture of exponentials.

2.B Appendix - A hybrid algorithm

As mentioned in the introduction, the strategy of our randomized Algorithm 2 is very different to the deterministic ones, and one can combine both to form a new algorithm. The randomized Algorithm 2 runs into trouble when the independence assumption for the cone basis in Theorem 2.4 is not met which can happen in datasets with highly correlated features; on the other hand, the deterministic algorithms have the disadvantage that they need to complete a full run over the whole dataset even when geometric greedy sampling could have finished much earlier. We give the details for this hybrid Algorithm 3 below; it has a worst case running time of the same order as the deterministic Algorithms, Litterer and Lyons [119], Maalouf et al. [126], Tchernychova [177] but in return has a very good chance of terminating faster. We demonstrate this by benchmarking it against the same datasets for fast least square solvers that were used in Maalouf et al. [126].

Some observations:

(i) Algorithm 3 always finds a solution;

(ii) in the case of dataset "with linear dependence", e.g. [94], if PCA reductions are not allowed as in the case of the application of Maalouf et al. [126], the basis $\mathbf{b}$ won't be invertible and therefore the complexity of the Algorithms in Litterer and Lyons

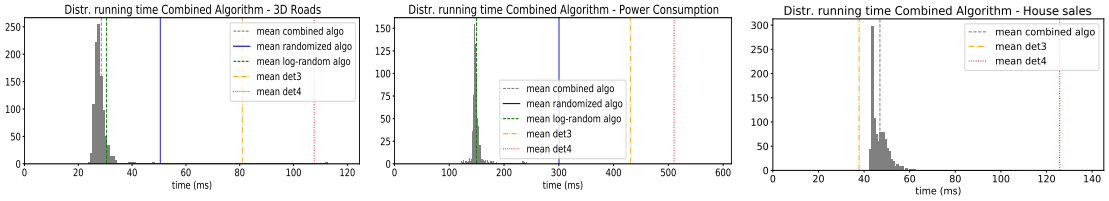


Figure 2.9: The running time of the Combined Algorithm 3. Note that the mean of the log-random Algorithm is essentially equal to the one of the Combined Algorithm, except for the dataset [94] where the random-algo does not work (cf. discussion above). For [94] the features are highly correlated, so Algorithm 2 fails, but the additional time necessary to Algorithm 3 is only due to checking if the sampled basis are invertible; hence, in this case Algorithm 3 behaves as it was deterministic. We have chosen *#_trials*= 10 and $G = 50(n+1)$.

[119], Maalouf et al. [126], Tchernychova [177] would worsen of "only" the complexity to check *#_trials* times if a $n \times n$ matrix is invertible, i.e. *#_trials*$\times O(n^3)$, see Figure 2.9;

(iii) Algorithm 2 must be run without reset, however note that we can add a reset strategy changing the number of iterations allowed to Algorithm 2 at step 8;

(iv) following the guidelines of Section 2.A, $G = 50(n+1)$ for "small n".

To sum up, as Figure 2.9[5], the hybrid Algorithm 3 takes advantage of both the greedy geometric sampling of Algorithm 2 and robustness of the deterministic Algorithms for highly correlated datasets. Totalled over all the datasets Algorithm 3 is the fastest. Many variations of the above hybrid algorithm are of course possible.

[5]For practical reasons in our implementation of Algorithm 3 we have used our implementation of *det3*, see Section 5.5.

Algorithm 3 Combined measure reduction algorithm

procedure REDUCE-COMBINED(A set $\mathbf{x}$ of N points in $\mathbb{R}^n$, $\mu = \{w_i\}$)
  rem_points $\leftarrow N$
  while rem_points$> n+1$ **do**
    Subdivide the points $\mathbf{x}$ in $G\wedge$ rem_points groups $\{\mathbf{x}_j\}_{j=1}^{G\wedge\text{rem_points}}$
    Compute $\bar{\mathbf{w}}_j = \sum_{w_i\,:\,x_i\in\mathbf{x}_j} w_i$, $\bar{\mathbf{x}}_j = \sum_{x_i\in g_j} w_i x_i/\bar{\mathbf{w}}_j$
    for *#_trials* times **do**
      $\mathbf{b} \leftarrow n$ random vectors from $\{\bar{\mathbf{x}}_j - \sum_j \bar{\mathbf{w}}_j\mathbf{x}_j\}_{j=1}^{G}$
      $\mathbf{x}^\star, w^\star \leftarrow$ Algorithm 2 with the points $\{\bar{\mathbf{x}}_j - \sum_j \bar{\mathbf{w}}_j\mathbf{x}_j\}$ using $\mathbf{b}$ as cone basis
      if Algorithm 2 has found a solution **then**
        Exit for
      end if
    end for
    if Algorithm 2 has not found a solution **then**
      $\mathbf{x}^\star, w^\star \leftarrow$ Deterministic Algorithm (e.g. [119, 126, 177]) with measure $\{\bar{\mathbf{x}}_j\}$, $\{\bar{\mathbf{w}}_j\}$
    end if
    $\mathbf{x} \leftarrow \mathbf{x} \setminus \{x_i \text{ s.t. } x_i \in \mathbf{x}_j \text{ and } \bar{\mathbf{x}}_j \in \mathbf{x}^\star\}$ ▷ Eliminate the points
    rem_points $\leftarrow N-$Cardinality$(\{x_i \text{ s.t. } x_i \in \mathbf{x}_j \text{ and } \bar{\mathbf{x}}_j \in \mathbf{x}^\star\})$
    $\{w_i\} \leftarrow \{w_i\} \setminus \{w_i \text{ s.t. } x_i \in \mathbf{x}_j \text{ and } \bar{\mathbf{x}}_j \in \mathbf{x}^\star\}$
    $\{w_i\} \leftarrow \{w_i \times w_j^\star \text{ s.t. } x_i \in \bar{\mathbf{x}}_j\}$ ▷ Recalibrate the weights
  end while
  return ($\mathbf{x}^\star$,$w^\star$)
end procedure

3 Carathéodory Sampling for Stochastic Gradient Descent

Keywords: Probability, Machine Learning, Big data optimization, Block Coordinate Descent

Many problems require to optimize an empirical risk function. Gradient descent methods that calculate the full gradient in every descent step do not scale to large datasets. Various flavours of Stochastic Gradient Descent (SGD) address this by approximating the sum that computes the gradient with a sum over a small, randomly chosen subsample of the data set. We present a different approach that is inspired by classical results of Tchakaloff and Carathéodory about measure reduction. Using the results and the algorithms from Chapter 2 we can replace a discrete probability measure with another, carefully constructed, discrete probability measure that has a much smaller support but preserves certain statistics; in particular, one can choose it so that the expected gradient computed is the same. To turn this into scalable algorithms we firstly, select the descent steps where such a measure reduction is carried out adaptively; secondly, we combine this approach with Block Coordinate Descent which allows to carry out the measure reduction efficiently. Finally, we provide an experimental validation and comparison. This Chapter contains the material presented in Cosentino et al. [41].

3.1 Introduction

A common task in Machine Learning is the optimization problem

$$\arg\min_\theta \mathbb{E}[L(\theta, X, Y)],$$

where $L(\theta,\cdot)$ is a generic loss function and the expectation is taken with respect to the joint distribution of (X,Y); e.g. $L(\theta,x,y) = (x^\top\theta - y)^2 + \lambda|\theta|_1$ for LASSO, Murphy [133], Tibshirani [178]. In practice, the distribution of (X,Y) is not known and one approximates it with the empirical measure $\mu = \frac{1}{N}\sum_{i=1}^N \delta_{(x_i,y_i)}$ built from N samples (x_i,y_i) of the pair (X,Y). That is, one minimizes the so-called empirical risk

$$\theta^\star := \arg\min_\theta \mathbb{E}[L(\theta,Z)] = \frac{1}{N}\sum_{i=1}^N L(\theta,z_i), \tag{3.1}$$

where Z denotes the discrete random variable that takes the values $z_i = (x_i,y_i)$ with equal probability. If $L(\theta,\cdot)$ is smooth with bounded derivatives and convex, standard gradient descent (GD),

$$\theta_{j+1} - \theta_j = -\frac{\gamma}{N}\sum_{i=1}^N \nabla_\theta L(\theta,z_i)\Big|_{\theta=\theta_i} \tag{3.2}$$

converges if the learning rate γ is appropriately chosen, $\lim_{i\to\infty}\theta_i = \theta^\star$, Nesterov [138]. However, for large-scale problems when the number of samples N is huge, the evaluation of the gradient in Equation (3.2) in every iteration step is prohibitive.

Popular approaches to reduce the cost in each iteration step are the so-called stochastic gradient descent (SGD) algorithms. The gradient is approximated by selecting at each iteration step j a subset of the N points at random. Solving the minimization problem (3.1) via (S)GD has a long history and is a research topic that is still rapidly evolving; we refer to Bottou et al. [23], Robbins and Monro [156], Ruder [161] for a general overview. This Chapter is inspired by variants of SGD that produce better estimators for the expected gradient than the naive estimator given by subsampling a batch of data points uniformly at random. This reduces the variance in each step and often guarantees a better convergence rate. Popular examples of such SGD variants include stochastic average gradients (SAG), Roux et al. [158], Iterate Averaging, Polyak and Juditsky [147], Incremental Aggregated Gradient, Blatt et al. [19].

This Chapter relies on replacing the empirical measure $\mu = \frac{1}{N}\sum_{i=1}^N \delta_{z_i}$ by a measure $\hat{\mu}$ with much smaller support, which however has the property that $\mathbb{E}_{Z\sim\mu}[\nabla_\theta L(\theta,Z)] =$

$\mathbb{E}_{Z\sim\hat{\mu}}[\nabla_\theta L(\theta, Z)]$ at carefully selected iteration steps. The construction of this reduced measure $\hat{\mu}$ that matches certain expectations of μ is known as the *recombination problem*, see Chapter 2. In particular, we rely on Algorithm 3 that shows strong performance when the number of samples N is very large. However, we emphasize that the ideas below are independent of the choice of the concrete recombination algorithm and any improvement on recombination algorithms will result in an improvement of Caratheodory's subsampling for SGD.

In the second part, we combine our proposed Caratheodory subsampling with Block Coordinate Descent (BCD) to make Caratheodory sampling effective when θ is high-dimensional. Generally, under the block separability assumptions on the regularization term and usual conditions on the principal part of the loss function, e.g. convexity or the Polyak-Lojasiewicz condition, the convergence is proved and the rates of convergence have been found, e.g. Csiba and Richtárik [49], Nutini et al. [142]. Applications of BCD techniques have been studied in sparse settings, Nutini et al. [142], large scale Gaussian process regression, Bo and Sminchisescu [20], L1 Regularized Least Squares, Santis et al. [162], Group LASSO, Meier et al. [128], Training Support Vector Machines, Platt [146], matrix and tensor factorization, Xu and Yin [191] and other works.

Instead of approximating the sum $\mathbb{E}[L(\theta, Z)] = \frac{1}{N}\sum_{i=1}^{N} L(\theta, z)_i$ by subsampling, we construct at certain steps j in the GD iteration of θ_j a new probability measure $\hat{\mu}_j$ supported on a *very small subset* of the original N atoms. This measure $\hat{\mu}$, however, matches certain statistical functions of the empirical measure μ; in particular, we can choose it such that $\mathbb{E}_{Z\sim\mu}[L(\theta_j, Z)] = \mathbb{E}_{Z\sim\hat{\mu}}[L(\theta_j, Z)]$, see Chapter 2. Although it can be relatively costly to carry out the recombination at a given step, in return the gradient is perfectly matched at this step and the expectation can be computed as a sum over n weighted points rather than $N \gg n$ uniformly weighted points. We balance this tradeoff by combining two techniques:

(i) by using an approximation to the Hessian to derive a control statistic that tells us when to carry out the recombination step. In practice, this allows to do only a few recombination computations over the whole descent trajectory;

(ii) by using Block coordinate descent (BCD) to carry out the reduction only for a subset

of the coordinates of the gradient. This makes a recombination cheap even if θ is high-dimensional.

Section 3.2 provides the main theoretical results and a first application to logistic regression; Section 3.3 then recalls BCD techniques, and shows how this allows to efficiently carry out Caratheodory subsampling for high-dimensional θ. Further, it benchmarks the resulting Carathéodory BCD (CaBCD) against classic SGD algorithms SAG and ADAM. Section 3.3 also compares the rules regarding the selection of the coordinates' blocks introduced in Nutini et al. [142] with the same rules when the Carathéodory Sampling is applied. A Python implementation for all of our experiments can be found at github.com/FraCose/Caratheodory_GD_Acceleration.

3.2 Carathéodory Gradient Descent (CaGD)

Given a dataset $\{(x_i,y_i) : i = 1,\dots,N\}$ consisting of N observations x_i with labels y_i, we denote by Z the discrete random variable that takes the value $z_i = (x_i,y_i)$ with probability $\frac{1}{N}$. That is, the empirical risk (3.1) at θ equals $\mathbb{E}[L(\theta, Z)]$. Further, denote with $G(\theta,z) := \nabla_\theta L(\theta, z) \in \mathbb{R}^n$ the gradient at θ and with $H(\theta, z) := \nabla^2_\theta L(\theta, z)$ the Hessian. With this notation, the usual GD iteration reads as

$$\theta_{j+1} := \theta_j - \gamma \mathbb{E}[G(\theta,Z)],$$

and converges, under assumptions which we recall below, to the minimum $\theta^\star$ as given in (3.1), see Nesterov [138]. However, in every descent step j the evaluation of the sum

$$\mathbb{E}[G(\theta,Z)] = \frac{1}{N}\sum_{i=1}^{N} G(\theta,(x_i, y_i))$$

can be costly. Below we use Tchakalof Theorem 2.1 to derive a similar iteration $(\hat{\theta}_j)$, that also converges to $\theta^\star$, which however avoids the evaluation of $\mathbb{E}[G(\theta_j,Z)]$ at most of the steps.

The first recombination step. Initialize $\theta_0 \in \mathbb{R}^n$ as before in (S)GD, but before the first step, apply Tchakalof Theorem 2.1 to Z and the n coordinate functions of the gradient $z \mapsto G(\theta_0,z)$ to produce a discrete random variable $\hat{Z}_0$. By construction, this random variable $\hat{Z}_0$ has only $n+1$ different possible outcomes and these outcomes are part of the original dataset $\{z_i = (x_i,y_i), i = 1,\dots,n\}$. Now define the first descent step

$$\hat{\theta}_1 := \theta_0 - \gamma \mathbb{E}[G(\theta_0, \hat{Z}_0)].$$

Since by construction, $\mathbb{E}[G(\theta_0,Z)] = \mathbb{E}[G(\theta_0,\hat{Z}_0)]$, it follows that $\hat{\theta}_1 = \theta_1$. In general $\mathbb{E}[G(\hat{\theta}_1,Z)] \neq \mathbb{E}[G(\hat{\theta}_1,\hat{Z}_0)]$ but the intuition is that $\mathbb{E}[G(\hat{\theta}_1,\hat{Z}_0)]$ is a good approximation of $\mathbb{E}[G(\hat{\theta}_1, Z)]$ for reasonable choices of γ. Hence, we continue to iterate

$$\hat{\theta}_{j+1} := \hat{\theta}_j - \gamma \mathbb{E}[G(\hat{\theta}_j, \hat{Z}_0)]. \tag{3.3}$$

until the first time τ_1 a control statistic tells us that the error has become too large.

A control statistic. Let L be convex, twice differentiable, and its gradient be Lipschitz, we show later that a natural choice for control statistic is the quantity

$$\Delta_{j,0} := \mathbb{E}[G(\theta_0,\hat{Z}_0)] \cdot (\hat{\theta}_j - \theta_0) + \frac{c}{2}\|\hat{\theta}_j - \theta_0\|^2, \tag{3.4}$$

where c is such that $v^\top H(\theta,z)v \leq c$ for every $v \in \mathbb{R}^n$; the existence of such a c is justified by the assumptions on L. More precisely, $\Delta_{j,0} < \Delta_{j-1,0}$ guarantees that the loss function L continues to decrease. Hence, we follow the iteration (3.3) until $\Delta_{j,0} \geq \Delta_{j-1,0}$, that is until step $\tau_1 := \inf\{j \geq 1 : \Delta_{j,0} \geq \Delta_{j-1,0}\}$, where we fix $\Delta_{0,0} := 0$. At time τ_1 we then simply update $\hat{Z}_0$ to $\hat{Z}_1$ so that the gradients are matched at the point $\hat{\theta}_{\tau_1-1}$, that is $\hat{Z}_1$ is such that $\mathbb{E}[G(\hat{\theta}_{\tau_1-1},Z)] = \mathbb{E}[G(\hat{\theta}_{\tau_1-1},\hat{Z}_1)]$, and then we continue as before.

CaGD in a nutshell. To sum up, we set $\tau_0 := 1$, $\Delta_{0,0} = 0$, and construct $\hat{Z}_0$ such that $\mathbb{E}[G(\theta_0,Z)] = \mathbb{E}[G(\theta_0,\hat{Z}_0)]$. We then update, for $j \geq 0$,

$$\hat{\theta}_{j+1} := \hat{\theta}_j - \gamma \mathbb{E}[G(\hat{\theta}_j, \hat{Z}_0)] \text{ as long as } \Delta_{j,0} < \Delta_{j-1,0}.$$

At time τ_1 we compute $\hat{Z}_1$ such that

$$\mathbb{E}[G(\hat{\theta}_{\tau_1-1}, Z)] = \mathbb{E}[G(\hat{\theta}_{\tau_1-1}, \hat{Z}_1)]$$

and update for $j \geq \tau_1 - 1$

$$\hat{\theta}_{j+1} := \hat{\theta}_j - \gamma \mathbb{E}[G(\hat{\theta}_j, \hat{Z}_1)], \text{ as long as } \Delta_{j,1} < \Delta_{j-1,1}$$

where $\Delta_{j,1} := \mathbb{E}[G(\hat{\theta}_{\tau_1-1}, \hat{Z}_1)] \cdot (\hat{\theta}_j - \hat{\theta}_{\tau_1-1}) + \frac{c}{2}\|\hat{\theta}_j - \hat{\theta}_{\tau_1-1}\|^2$ and $\Delta_{\tau_1-1,1} = 0$. At time $\tau_2 := \inf\{j \geq \tau_1 : \Delta_{j,1} \geq \Delta_{j-1,1}\}$ we compute $\hat{Z}_2$ such that $\mathbb{E}[G(\hat{\theta}_{\tau_2-1}, Z)] = \mathbb{E}[G(\hat{\theta}_{\tau_2-1}, \hat{Z}_2)]$, etc.

Convergence and convergence rate

The above is the main structure of our first algorithm, denoted as Carathéodory Gradient descent (CaGD). However, we add three further modifications. First, we stop as soon as the gradient or the value of the loss function is smaller than a given ϵ; second, we bound the number of iterations between two recombinations by a constant, that is $\tau_{k+1} - \tau_k \leq$ it_max_Ca, to avoid pathological cases, see Theorem 3.1 for more details; third, we allow to match a general oracle direction D_j at step j. The choice $D_j = -G(\theta_j, Z)$ is the most relevant for this Section, but the general oracle formulation allows to use more involved choices; e.g. momentum strategies in Section 3.3. This leads to Algorithm 4. In Algorithm 4 we write $D_j(\{\theta\}, Z)$ to express its dependencies on the data Z and the sequence $\{\theta\}$ computed up to the step j, although it could depend also on the loss function L, in particular it could depend on its derivatives G, H, etc.

Theorem 3.1 shows that it converges whenever we match oracle directions.

Theorem 3.1. *Let $L(\theta, \cdot)$ be convex, twice differentiable in θ and its gradient G be Lipschitz. If the quantities $\{\theta_j\}$ defined as*

$$\theta_j - \theta_{j-1} = \gamma \mathbb{E}[D_{j-1}(\{\theta\}, Z)]$$

converge to a minimum θ^*, i.e. $\lim_{j\to\infty} \theta_j = \theta^*$, then the sequence of $\{\hat{\theta}\}$ computed via Algorithm 4 converges to a (possibly different) minimum $\hat{\theta}^*$, $\lim_{j\to\infty} \hat{\theta}_j = \hat{\theta}^*$.

Algorithm 4 Carathéodory Sampling Acceleration

Initialize $\hat{\theta}_0$
$j \leftarrow 1$, $k \leftarrow 0$ ▷ j counts steps, $k+1$ the number of recombinations
$\tau_0 \leftarrow 1$ ▷ $\tau_{k+1} - \tau_k$ is the step we made with the $(k+1)$th recombination
$\Delta_{\tau_0 - 1,0} \leftarrow 0$
$\mathrm{Grad}_{\tau_0} \leftarrow \mathbb{E}[G(\hat{\theta}_0, Z)]$
while ($\| \mathrm{Grad}_{\tau_k} \| > \epsilon_1$ **or** $|L(\hat{\theta}_{\tau_k}, Z)| > \epsilon_2$) **and** $j \leq$ it_max **do**
  Compute $\hat{Z}_k$ such that $\mathbb{E}[D_{\tau_k}(\{\hat{\theta}\}, \hat{Z}_k)] = \mathbb{E}[D_{\tau_k}(\{\hat{\theta}\}, Z)]$ ▷ Reduce Z to $\hat{Z}_k$
  while $\Delta_{j,k} < \Delta_{j-1,k}$ **and** $j - \tau_k \leq$ it_max_Ca **do**
    $\hat{\theta}_j \leftarrow \hat{\theta}_{j-1} + \gamma \mathbb{E}[D_{j-1}(\{\hat{\theta}\}, \hat{Z}_k)]$
    $\Delta_{j,k} \leftarrow \mathrm{Grad}_{\tau_k} \cdot (\hat{\theta}_j - \hat{\theta}_{\tau_k - 1}) + \frac{c}{2} \|\hat{\theta}_j - \hat{\theta}_{\tau_k - 1}\|^2$
    $j \leftarrow j + 1$
  end while
  if $j - \tau_k \neq$ it_max_Ca **then**
    $\tau_k, j \leftarrow j - 1$
  else
    $\tau_k, j \leftarrow j$
  end if
  $\mathrm{Grad}_{\tau_k} \leftarrow \mathbb{E}[G(\hat{\theta}_{\tau_k - 1}, Z)]$, $\Delta_{\tau_k - 1, k} \leftarrow 0$
  $k \leftarrow k + 1$
end while and return j, $\hat{\theta}_j$

Proof. Thanks to the hypothesis there exists c s.t.

$$\begin{aligned}\mathbb{E}[L(\hat{\theta}_j, Z)] &= \mathbb{E}[L(\hat{\theta}_0, Z)] + \mathbb{E}[G(\hat{\theta}_0, Z)] \cdot (\hat{\theta}_j - \hat{\theta}_0) + \frac{1}{2} (\hat{\theta}_j - \hat{\theta}_0)^\top \cdot \mathbb{E}[H(\bar{\theta}, Z)] \cdot (\hat{\theta}_j - \hat{\theta}_0) \\ &\leq \mathbb{E}[L(\hat{\theta}_0, Z)] + \mathbb{E}[G(\hat{\theta}_0, Z)] \cdot (\hat{\theta}_j - \hat{\theta}_0) + \frac{c}{2} \|\hat{\theta}_j - \hat{\theta}_0\|^2, \quad \text{for } j \geq 0,\end{aligned}$$

where $\bar{\theta}$ is a convex combination of $\hat{\theta}_j$ and $\hat{\theta}_0$. It is now easy to see that we have a condition to check, in order to rebuild the measure: we update the measure after τ_1 steps, where

$$\begin{aligned}\tau_1 &:= \inf\{j \geq 1 : \Delta_{j,0} \geq \Delta_{j-1,0}\} \\ \Delta_{j,0} &:= \mathbb{E}[G(\hat{\theta}_0, Z)] \cdot (\hat{\theta}_j - \hat{\theta}_0) + \frac{c}{2} \|\hat{\theta}_j - \hat{\theta}_0\|^2,\end{aligned}$$

where $\Delta_{0,0} = 0$. We have that $\{\Delta_{0,0}, \Delta_{1,0}, \dots \Delta_{\tau_1 - 1, 0}\}$ is a negative decreasing sequence and therefore

$$\mathbb{E}[L(\hat{\theta}_{\tau_1 - 1}, Z)] \leq \mathbb{E}[L(\hat{\theta}_0, Z)].$$

In particular, note that $\Delta_{1,0} \leq 0$, since $\hat{\theta}_1 = \theta_1 := \theta_0 + \gamma \mathbb{E}[D_0(\{\theta\}, Z)]$, thanks to

Theorem 2.2 and the definition of $\hat{Z}_0$, therefore $\tau_1 \geq 2$. $\tau_1 - 1 = 1$ means that the reduced r.v. $\hat{Z}_0$ computed has been useless, i.e. we have done only one step with the reduced measure that we could have done directly using $\mathbb{E}[D_0(\{\theta\},Z)]$ without computing the reduced measure.

The reasoning can be generalized: we can define for $k \geq 1$, and $j > \tau_k - 1$

$$\begin{aligned} \Delta_{j,k} :=& \mathbb{E}[G(\hat{\theta}_{\tau_k-1}, Z)] \cdot (\hat{\theta}_j - \hat{\theta}_{\tau_k-1}) + \frac{c}{2}\|\hat{\theta}_j - \hat{\theta}_{\tau_k-1}\|^2 \\ \tau_k :=& \inf\{j \geq \tau_{k-1} : \Delta_{j,k-1} \geq \Delta_{j-1,k-1}\}, \end{aligned} \tag{3.5}$$

where $\Delta_{\tau_k-1,k} = 0$. The proof of the convergence follows since if $\tau_k - \tau_{k-1} = 1$ we follow the directions $D_j(\{\theta\},Z)$ which converge for the hypothesis, whereas if $\tau_k - \tau_{k-1} \geq 1$ the value of L decreases,

$$\mathbb{E}[L(\hat{\theta}_{\tau_k-1},Z)] \leq \mathbb{E}[L(\hat{\theta}_{\tau_{k-1}-1},Z)].$$

However, since L is convex, it is not possible to claim that the sequence $\{\hat{\theta}_j\}$ converges to the same minimum θ^*. Moreover, to avoid pathological cases, e.g. $\Delta_{1,k} > \Delta_{2,k} > \ldots \searrow -a$, $a > 0$ in which cases $L(\hat{\theta}_j,Z)$ cannot decrease "enough", we impose a number of maximum iterations that the Algorithm can do with the reduced measure. □

Theorem 3.1 can be extended to the case where the learning rate γ is not fixed. Theorem 3.2 gives the convergence rate for the choice $D_j(\{\theta\},Z) = -G(\theta_j, Z)$.

Theorem 3.2. *Let $L(\theta, \cdot)$ be convex, twice differentiable in θ and its gradient G be Lipschitz. Then if $D_j = -G(\hat{\theta}_j,Z)$, $\gamma \leq 1/Lip(G)$ Algorithm 4 converges to a minimum θ^* and its convergence rate is*

$$|\mathbb{E}L(\hat{\theta}_j, Z) - \mathbb{E}L(\theta^*, Z)| \leq \frac{1}{2\gamma} J \frac{\|\hat{\theta}_0 - \theta^*\|^2}{j}, \tag{3.6}$$

where j is the number of iterations, and J is the maximum number of times the reduced measure is used (as per Algorithm 4, we can conservatively bound $J <$ it_max_Ca).

Proof. The convergence is a simple application of Theorem 3.1. We can show that Algorithm 4 does not reduce the order of convergence of the standard GD. Let us call $\hat{\theta}_i$ the

sequence of weights obtained by Algorithm 4 in chronological order

$$\{\hat{\theta}_0,\hat{\theta}_1,\dots,\hat{\theta}_{\tau_1-1},\hat{\theta}_{\tau_1},\dots,\hat{\theta}_{\tau_2-1},\hat{\theta}_{\tau_2},\dots\},$$

where - fixing $\tau_0 = 1$ - for $k \geq 1$, $\tau_{k+1} - \tau_k$ indicates the number of times we use the reduced measure computed using $\hat{\theta}_{\tau_k-1}$. Moreover, let us suppose that for any step j we have a map S that tell us how many times we have recomputed the measure, so $S(j) = \max\{k : \tau_k \leq j\}$. In other words, given j, $\hat{\theta}_{\tau_{S(j)}-1}$ is the weight used to compute the recombined measure the last time. Let us call for brevity $L(\theta) := \mathbb{E}L(\theta, Z)$. Due to the Definitions of $\Delta_{j,k}$ and τ_k see Equation (3.5), we have that

$$L(\hat{\theta}_j) \leq L(\hat{\theta}_{\tau_{S(j)}-1}) + \Delta_{j,S(j)} \leq L(\hat{\theta}_{\tau_{S(j)}-1}) + \Delta_{\tau_{S(j)},S(j)},$$

therefore we have that

$$\begin{aligned}
L(\hat{\theta}_j) \leq & L(\hat{\theta}_{\tau_{S(j)}-1}) + \nabla L(\hat{\theta}_{\tau_{S(j)}-1}) \cdot (\hat{\theta}_{\tau_{S(j)}} - \hat{\theta}_{\tau_{S(j)}-1}) + \frac{c}{2}\|\hat{\theta}_{\tau_{S(j)}} - \hat{\theta}_{\tau_{S(j)}-1}\|^2 \\
\leq & L(\hat{\theta}_{\tau_{S(j)}-1}) - \frac{1}{2}\gamma\|\nabla L(\hat{\theta}_{\tau_{S(j)}-1})\|^2 \\
\leq & L(\theta^*) + \nabla L(\hat{\theta}_{\tau_{S(j)}-1})(\hat{\theta}_{\tau_{S(j)}-1} - \theta^*) - \frac{1}{2}\gamma\|\nabla L(\hat{\theta}_{\tau_{S(j)}-1})\|^2.
\end{aligned}$$

We have used in order (i) the Definition of $\Delta_{j,k}$, Equation (3.5); (ii) $\gamma \leq 1/\mathrm{Lip}(G)$, where $\mathrm{Lip}(G)$ indicates the Lipschitz constant of G; (iii) the fact that

$$\hat{\theta}_{\tau_{S(j)}} - \hat{\theta}_{\tau_{S(j)}-1} = -\gamma\mathbb{E}[G(\hat{\theta}_{\tau_{S(j)}-1}, \hat{Z}_{\tau_{S(j)}})] = -\gamma\mathbb{E}[G(\hat{\theta}_{\tau_{S(j)}-1}, Z)], \tag{3.7}$$

by Definition of $\hat{Z}_{\tau_{S(j)}}$; and that, (iv) $L(\theta) \leq L(\theta^*) + \nabla L(\theta)(\theta - \theta^*)$, since L is convex. Rearranging the terms and using again Equation (3.7) it is possible to obtain

$$L(\hat{\theta}_j) - L(\theta^*) \leq \frac{1}{2\gamma}\left(\|\hat{\theta}_{\tau_{S(j)}-1} - \theta^*\|^2 - \|\hat{\theta}_{\tau_{S(j)}} - \theta^*\|^2\right).$$

Thus,

$$\begin{aligned}\sum_{l=1}^{j} L(\hat{\theta}_l) - L(\theta^*) \leq & \frac{1}{2\gamma} \sum_{l=1}^{j} \left(\|\hat{\theta}_{\tau_{S(l)}-1} - \theta^*\|^2 - \|\hat{\theta}_{\tau_{S(l)}} - \theta^*\|^2 \right) \\ = & \frac{1}{2\gamma} \sum_{k:\tau_k \leq j} (\tau_k - \tau_{k-1}) \left(\|\hat{\theta}_{\tau_k-1} - \theta^*\|^2 - \|\hat{\theta}_{\tau_k} - \theta^*\|^2 \right) \\ \leq & \frac{1}{2\gamma} \max_{k:\tau_k \leq j} \{\tau_k - \tau_{k-1}\} \sum_{k:\tau_k \leq j} \left(\|\hat{\theta}_{\tau_k-1} - \theta^*\|^2 - \|\hat{\theta}_{\tau_k} - \theta^*\|^2 \right) \\ \leq & \frac{1}{2\gamma} \max_{k:\tau_k \leq j} \{\tau_k - \tau_{k-1}\} \|\hat{\theta}_0 - \theta^*\|^2.\end{aligned}$$

Therefore it holds that

$$L(\hat{\theta}_j) - L(\theta^*) \leq \frac{1}{j} \sum_{l=1}^{j} L(\hat{\theta}_l) - L(\theta^*) \leq \frac{1}{2\gamma} \max_{k:\tau_k \leq j} \{\tau_k - \tau_{k-1}\} \frac{\|\hat{\theta}_j - \theta^*\|^2}{j}.$$

□

First, note that the bound in Inequality (3.6) assumes that when we use the reduced measures the objective function does not decrease, thanks to Equation (3.4). Secondly, note that the constant c in Equation (3.4) in practice might be unknown and expensive to compute, and when known it might be quite conservative. In our implementation we use an approximation of the second derivative, so that $\Delta_{j,k}$ in Equation (3.4) becomes

$$\Delta_{j,k} := \mathbb{E}[G(\hat{\theta}_{\tau_k-1}, \hat{Z}_k)] \cdot (\hat{\theta}_j - \hat{\theta}_{\tau_k-1}) + \frac{1}{2} (\hat{\theta}_j - \hat{\theta}_{\tau_k-1})^\top \cdot \mathcal{H}_k \cdot (\hat{\theta}_j - \hat{\theta}_{\tau_k-1}), \quad j \geq \tau_k,$$

where $\mathcal{H}_k := \left[\mathbb{E}[G(\hat{\theta}_{\tau_k}, \hat{Z}_k)] - \mathbb{E}[G(\hat{\theta}_{\tau_k-1}, Z)] \right]^\top \cdot \left[1/(\hat{\theta}_{\tau_k} - \hat{\theta}_{\tau_k-1}) \right]$ and $[1/x]$ denotes a vector whose elements are the reciprocals of those in $[x]$. To compute the terms $\Delta_{(\cdot,\cdot)}$ we modify Algorithm 4 doing two iterations where $\mathbb{E}[G(\theta, Z)]$ is computed – see Algorithms 5 and 6 in Section 3.3. We do not discuss how to optimally select γ, since there exists a broad literature about the optimal selection of the step, Nesterov [138].

3.2.1 A first comparison of CaGD and GD: logistic regression

Already comparing the complexity of CaGD, Algorithm 4, to standard GD is not trivial, since the number of recombinations steps is not known a-priori (the times τ_k specify a

recombination computation depend on the data). Furthermore, the worst-case complexity of the recombination algorithm itself is much worse than its average complexity, see Chapter 2. However, the intuition remains that for a sufficiently large number of samples N and low-dimensional θ, the total cost of computing the recombinations is negligible compared to evaluating the full gradient in each descent step. We present three numerical experiments to test this intuition. We use classic logistic regression for binary classification (it is easy to check that the assumptions of Theorem 3.2 are fulfilled in this case, see [133, Exercise 8.3]) and use synthetic data which allows to study various regimes of N. We run both GD and CaGD until either the norm of the gradient is less than 1×10^{-3}, or the number of iterations is greater than 1×10^4.

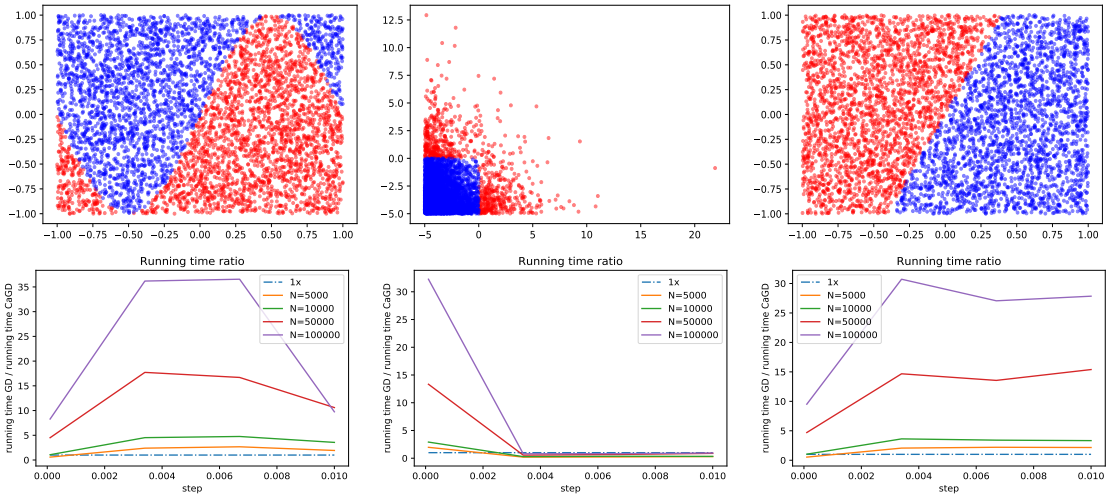


Figure 3.1: Synthetic data generated by sampling from (i) a uniform distribution, classified with a sine function, (ii) a shifted exponential, classified as 1 in the third octant, and 0 otherwise, (iii) a uniform distribution, classified with a logistic model with parameter $(-5; 2)$. The top row shows samples with $N = 5000$. The bottom row shows the ratios of running times between standard GD and CaGD, as a function of the step size for various sample sizes N.

The results are shown in Figure 3.1 and indicate that the improvement in the running time is up to 35-fold. Generally, the improvement increases as the number of points N increases and when the step size is small (the step of the GD must be small enough for the algorithm to converge to a minimum). Another advantageous observation is that CaGD reaches lower gradient values than GD, because in Algorithm 4 the "true" gradient $\mathbb{E}[G(\theta,Z)]$ is only computed at the step 18, but we modify $\hat{\theta}_j$ at step 9. In these instances

we have employed it_max_Ca= max{10/step,10^4}.

Of course, the real benchmarks are SGD variants like SAG and ADAM. However, CaGD in its simple form above is not competitive to such SGD variants since its computational bottleneck is that the recombination step scales cubically in the dimension n of θ which makes it infeasible for many datasets. In Section 3.3 below, we combine CaGD with BCD to resolve these issues. This then results in a competitive alternative to SGD variants like SAG and ADAM.

3.3 Carathéodory Block Coordinate Descent

The computational bottleneck of CaGD is in step 9, where a recombination algorithm is run to compute $\hat{Z}$. As mentioned before, the recombination algorithm in Tchernychova [177] has the worst case complexity $O(Nn + n^3 \log(N/n))$ where n is the dimension of θ and N the number of samples. However, unlike the other recombination algorithms, the greedy Algorithm 2 in Chapter 2 is designed to have a much lower average complexity in N. Unfortunately, in terms of the dimension n, it suffers like all the other recombination algorithms from the cubic term n^3 which makes CaGD not suitable for high-dimensional problems. In this Section, we combine CaGD with Block Coordinate Descent (BCD), Csiba and Richtárik [49], Nesterov [137], Nutini et al. [142], Richtárik and Takáč [154], Wright [189] to resolve this cubic complexity. This leverages the strengths of BCD in terms of computational efficiency and of CaGD in terms of low variance estimators for the expected gradient estimate.

3.3.1 From Block Coordinate Descent to Carathéodory Block Coordinate Descent

BCD selects in every descent step a small subset of the n coordinates of θ. This, in turn, allows us to apply the recombination algorithm to a small number of coordinates, typically n is between 2 and 5 so that the cubic complexity in the number of coordinates becomes

negligible. The core idea of BCD is to update only a subset of the coordinates

$$\theta_{j+1} = \theta_j - \gamma \mathbb{E}[G^{(B(j))}(\theta, Z)],$$

where $B(j)$ denotes a set of coordinates, and $G^{(B(j))}$ denotes the function that returns the gradient for the coordinate in $B(j)$, and sets it to be equal to 0 for the other coordinates. If the problem we want to solve is of the form

$$\min_{\theta} L(\theta, \cdot) = \min_{\theta}(f(\theta, \cdot) + g(\theta)),$$

where f is convex and g is (block) separable, i.e. $g = \sum_{m=1}^{b} g_m$, $g_m : \mathbb{R}^{n_m} \to \mathbb{R}$ and $\sum_m n_m \leq n$, $b \leq n$, then BCD converges to a minimum and the rate of convergence is that of standard GD, up to some constants depending on different factors, e.g. the number of directions we update at any step, the strategy to choose such directions, the separability of the objective function g; see Richtárik and Takáč [154], Wright [189] for a detailed study. Notable examples of optimisation problems with functions (f,g) abiding by the previous condition are least-squares problems with LASSO regularisation, Fercoq and Richtárik [57], Nesterov [137], Richtárik and Takáč [154], which is where BCD appears to be more effective (we analyse this problem in the next subsection). A well-studied aspect of BCD is its parallelisation, Fercoq and Richtárik [57], Scherrer et al. [163], which can be studied in terms of the spectral radius of the data, Bradley et al. [27]. In the following, we focus on simple applications of BCD to highlight the improvements that are due to CaBCD, rather than BCD optimizations.

3.3.2 Update rules

An important aspect is how to select the directions for the descent step, Glasmachers and Dogan [68], Li et al. [118], Nutini et al. [142], Qu and Richtárik [151], and how many directions to select: cyclic versus acyclic, deterministic versus random, or via the Gauss-Southwell (GS) rule (discussed below). The main differences amongst these options hinges on whether or not we can afford to compute the full gradient: if we can, then the GS rule is

expected to be the best strategy; if we cannot, then the random acyclic strategy seems to perform better than the deterministic cyclic ones, Lee and Wright [115]. Moreover, these strategies can be implemented in parallel, exploiting multiple processors. We focus on the following two acyclic strategies, whose respective procedures are presented in Algorithm 5 and 6; for a detailed comparison see Lee and Wright [115], Nutini et al. [141]:

- *Modified Gauss-Southwell (GS).* If we can compute the full gradient, then we can select directions where the gradient is larger in absolute value. A rule of thumb is to consider only a percentage of the "total" value of the gradient (in our experiments we consider 75%):

  (i) let us call ∇_S the vector with the absolute value of the directions of the GD sorted in descending order, i.e. $\left|\nabla L^{\left(\nabla_S^{(r)}\right)}\right| \geq \left|\nabla L^{\left(\nabla_S^{(q)}\right)}\right|$ if $r \leq q$;

  (ii) we consider the directions where the gradient is bigger in absolute value, namely the first $\hat{n}$ components of ∇_S, where

  $$\hat{n} := \inf\left\{ q : \sum_{jr=1}^{q} \left|\nabla L^{\left(\nabla_S^{(r)}\right)}\right| > \text{Percentage} \right\};$$

  (iii) we split the $\hat{n}$ directions in $b = \hat{n}/s$ blocks of size s, respecting the ordering.

- *Random.* If we cannot compute the full gradient, then we can group the directions into n/s blocks of size s, and perform BCD over the blocks. In the experiments of the next subsection, we randomly group half of the directions per iteration. The condition to terminate in Algorithm 6 depends "only" on the loss function L since we cannot compute the full gradient.

In Fercoq and Richtárik [57], Nesterov [137], the selection of the coordinate step γ is given as a sub-optimisation problem, which for simplicity we skip in the following. Furthermore, [137] shows that a momentum application to the single block of directions can improve the rate of convergence: knowing the convexity of the function to be optimised, it is possible to obtain an accelerated method with an improved rate of convergence $O(1/j^2)$, where j is the number of iterations, in the same spirit of Nesterov [139]. Our implementation has

been done in a synchronous and parallel manner (cf. discussion above).

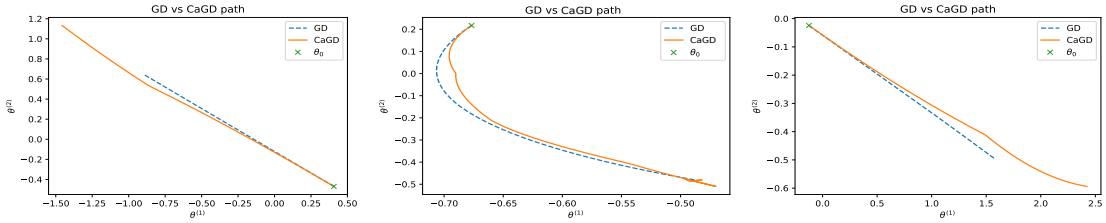


Figure 3.2: Paths generated by CaGD (Theorem 3.2) and GD, for the experiments of Figure 3.1, same order.

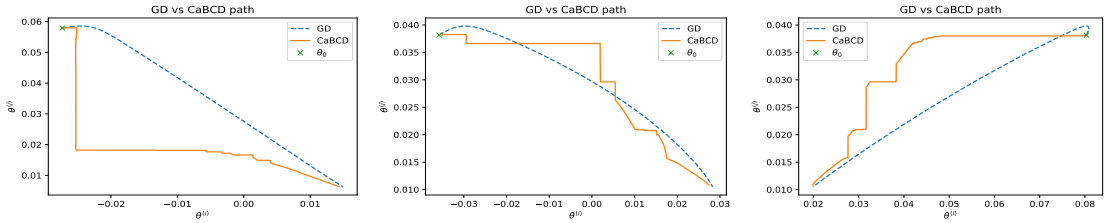


Figure 3.3: Samples of trajectories followed by the GD and the CaBCD over the parameter space. The dotted blue trajectories and the continuous orange trajectories converge to the same desired minimum, though via different paths. The CaBCD, between change of directions, uses only a subset of the total points N, namely $s+1$ if the size of the selected block is s. This Figure has been obtained using the data of Figure 3.1 (center) in the multi-dimensional case.

3.3.3 Experiments: CaBCD vs ADAM vs SAG for LASSO

We consider a least-squares problems with LASSO regularisation, i.e.

$$\min_{\theta} \frac{1}{N} \sum_{i} (x_i \theta^\top - y_i)^2 + \lambda |\theta|_1.$$

We have used the following datasets:

(i) Household power consumption [9], which consists of $N = 2\,075\,259$ data points. We want to predict the *voltage* given *active power, reactive power, intensity*. We have raised to the tensor power of 5[1], scaled the data, and applied PCA to reduce the number of features to 7.

[1]Raising to the tensor power of α means that we have added all the "mixed" products up to order α: if we indicate with x_m^i, $i \in \{1,\dots,n\}$, the i-th feature of the m-th point, in the case $\alpha = 3$, we create all the new features of the form $x_m^i \times x_m^j$ and $x_m^i \times x_m^j \times x_m^h$, $i,j,k \in \{1,\dots,n\}$ for all the points $m \in \{1,\dots N\}$.

Algorithm 5 Carathéodory BCD - Modified Gauss-Southwell (GS) rule

Initialize $\hat{\theta}_0$
$j \leftarrow 1$, $k \leftarrow 0$ ▷ j counts steps
$\text{Grad}_0 \leftarrow \mathbb{E}[G(\hat{\theta}_0,Z)]$
while $(\| \text{Grad}_{j-1} \| > \epsilon_1$ **or** $|L(\hat{\theta}_{j-1},Z)| > \epsilon_2)$ **and** $j \leq$ it_max **do**
  $\hat{\theta}_j \leftarrow \hat{\theta}_{j-1} + \gamma \mathbb{E}[D_{j-1}(\{\hat{\theta}\},Z)]$
  $\text{Grad}_j \leftarrow \mathbb{E}[G(\hat{\theta}_j,Z)]$
  Build b_k blocks $B(m,k)$, $m = 1,...,b_k$ using the $\mathbb{E}[G(\hat{\theta}_j,Z)]$ and the GS rule
  $\tau_k \leftarrow j$
  $j \leftarrow j + 1$
  for $m = 1,...,b_k$, **in parallel do**
    $j_m \leftarrow j, \quad \Delta^{(m)}_{\tau_k,k} \leftarrow 0$ ▷ $j_m - 1 = \tau_k$
    $\text{Hessian}^{(m)}_k \leftarrow \left[\text{Grad}^{(m)}_{\tau_k} - \text{Grad}^{(m)}_{\tau_k-1}\right]^\top \cdot \left[1/(\hat{\theta}^{(m)}_{\tau_k} - \hat{\theta}^{(m)}_{\tau_k-1})\right]$
    Compute $\hat{Z}^{(m)}_k$ s.t. $\mathbb{E}\left[D^{(m)}_{\tau_k}\left(\{\hat{\theta}\},\hat{Z}^{(m)}_k\right)\right] = \mathbb{E}\left[D^{(m)}_{\tau_k}\left(\{\hat{\theta}\},Z\right)\right]$
    while $\Delta^{(m)}_{j_m,k} \leq \Delta^{(m)}_{j_m-1,k}$ **and** $j_m - \tau_k \leq$it_max_Ca **do**
      $\hat{\theta}^{(m)}_{j_m} \leftarrow \hat{\theta}^{(m)}_{j_m-1} + \mathbb{E}\left[D^{(m)}_{j_m-1}\left(\{\hat{\theta}\},\hat{Z}^{(m)}_k\right)\right]$
      $\delta^{(m)}_{j_m,k} \leftarrow \hat{\theta}^{(m)}_{j_m} - \hat{\theta}^{(m)}_{\tau_k}$
      $\Delta^{(m)}_{j_m,k} \leftarrow \text{Grad}^{(m)}_{\tau_k} \cdot \delta^{(m)}_{j_m,k} + \left(\delta^{(m)}_{j_m,k}\right)^\top \cdot \text{Hessian}^{(m)}_k \cdot \delta^{(m)}_{j_m,k}$
      $j_m \leftarrow j_m + 1$
    end while
    if $j_m - \tau_k \neq$ it_max_Ca **then**
      $\tau_{m,k+1} \leftarrow j_m - 1$ ▷ $\tau_{m,k+1} - \tau_k$ steps in $(k+1)$th recombination relative to $B(m,j)$
    else
      $\tau_{m,k+1} \leftarrow j_m$
    end if
  end for
  $j \leftarrow j + \sum_m \tau_{m,k+1}$
  $\hat{\theta}^{(m)}_j \leftarrow \hat{\theta}^{(m)}_{\tau_{m,k+1}}, \quad \forall m$ ▷ synchronise and update $\hat{\theta}$
  $\text{Grad}_j \leftarrow \mathbb{E}[G(\hat{\theta}_j,Z)]$
  $k \leftarrow k + 1, \quad j \leftarrow j + 1$
end while and return j, $\hat{\theta}_j$

We write $\cdot^{(m)}$ in place of $\cdot^{(B(m,k))}$ to indicate the restriction to the components in the blocks $B(m,k)$.

Algorithm 6 Carathéodory BCD - Random

Initialize $\hat{\theta}_0$
$j \leftarrow 1$, $k \leftarrow 0$ ▷ j counts steps
while $|L(\theta_{j-1},Z)| > \epsilon$ **and** $j \leq$it_max **do**
  Build b blocks $B(m,k)$, $m = 1,...,b$ using the *Random* rule
  for $m = 1,...,b$, **in parallel do**
    $\text{Grad}_{j-1}^{(m)} \leftarrow \mathbb{E}[G^{(m)}(\hat{\theta}_{j-1},Z)]$
    $\hat{\theta}_j^{(m)} \leftarrow \hat{\theta}_{j-1}^{(m)} + \gamma\mathbb{E}[D_{j-1}^{(m)}(\{\hat{\theta}\},Z)]$
    $\text{Grad}_j^{(m)} \leftarrow \mathbb{E}[G^{(m)}(\hat{\theta}_j,Z)]$
    $\tau_k \leftarrow j$
    $j_m \leftarrow j+1, \quad \Delta_{\tau_k,k}^{(m)} \leftarrow 0$ ▷ $j_m - 1 = \tau_k$
    $\text{Hessian}_k^{(m)} \leftarrow \left[\text{Grad}_{\tau_k}^{(m)} - \text{Grad}_{\tau_k-1}^{(m)}\right]^\top \cdot \left[1/(\hat{\theta}_{\tau_k}^{(m)} - \hat{\theta}_{\tau_k-1}^{(m)})\right]$
    Compute $\hat{Z}_k^{(m)}$ s.t. $\mathbb{E}\left[D_{\tau_k}^{(m)}\left(\{\hat{\theta}\},\hat{Z}_k^{(m)}\right)\right] = \mathbb{E}\left[D_{\tau_k}^{(m)}\left(\{\hat{\theta}\},Z\right)\right]$
    while $\Delta_{j_m,k}^{(m)} \leq \Delta_{j_m-1,k}^{(m)}$ **and** $j_m - \tau_k \leq$it_max_Ca **do**
      $\hat{\theta}_{j_m}^{(m)} \leftarrow \hat{\theta}_{j_m-1}^{(m)} + \mathbb{E}\left[D_{j_m-1}^{(m)}\left(\{\hat{\theta}\},\hat{Z}_k^{(m)}\right)\right]$
      $\delta_{j_m,k}^{(m)} \leftarrow \hat{\theta}_{j_m}^{(m)} - \hat{\theta}_{\tau_k}^{(m)}$
      $\Delta_{j_m,k}^{(m)} \leftarrow \text{Grad}_{\tau_k}^{(m)} \cdot \delta_{j_m,k}^{(m)} + \left(\delta_{j_m,k}^{(m)}\right)^\top \cdot \text{Hessian}_k^{(m)} \cdot \delta_{j_m,k}^{(m)}$
      $j_m \leftarrow j_m + 1$
    end while
    if $j_m - \tau_k \neq$ it_max_Ca **then**
      $\tau_{m,k+1} \leftarrow j_m - 1$ ▷ $\tau_{m,k+1} - \tau_k$ steps in $(k+1)$th recombination relative to $B(m,j)$
    else
      $\tau_{m,k+1} \leftarrow j_m$
    end if
  end for
  $j \leftarrow j + \sum_m \tau_{m,k+1}$
  $\hat{\theta}_j^{(m)} \leftarrow \hat{\theta}_{\tau_{m,k+1}}^{(m)}, \quad \forall m$ ▷ synchronise and update $\hat{\theta}$
  $k \leftarrow k+1, \quad j \leftarrow j+1$
end while and return j, $\hat{\theta}_j$

We write $\cdot^{(m)}$ in place of $\cdot^{(B(m,k))}$ to indicate the restriction to the components of $\cdot$ in the blocks $B(m,k)$.

(ii) 3D Road Network [8], which consists of $N = 434\,874$ data points. We want to predict the *altitude*, given *longitude* and latitude. We have raised to the tensor power of 5, scaled the data, and applied PCA to reduce the number of features to 7.

(iii) NYC Taxi Trip Duration [95], which consists of $N = 1\,458\,644$ data points. We want to predict the *trip duration*, given *pickup time/longitude/latitude* and *dropoff longitude/latitude*. We consider only the time of the feature *pickup_datetime*, without the date. We have raised to the tensor power of 3, scaled the data, and applied PCA to reduce the number of features to 8. In this case we have considered as outliers the points such that $y_i > 10\,000$ – this amounts to 2123 points (0.14%).

In all datasets, the variance reduction by PCA is greater than 99.9%, which results from eliminating the symmetries introduced via the tensor power. Throughout we have chosen $\lambda = 0.01$ for the Lasso regularisation.

We have implemented the BCD with and without the Carathéodory sampling procedure with Gauss-Southwell rule (*CaBCD GS*, *BCD GS*), with a momentum strategy and the GS rule (*CaBCD mom GS*, *BCD mom GS*), and with the Random rule (*CaBCD mom random*, *BCD mom random*). For the momentum strategy, we have chosen the momentum parameter $\beta = 0.9$. As benchmarks we used ADAM Kingma and Ba [96] and SAG, Roux et al. [158] with standard mini-batches with size of 256. The learning rate for the CaBCD Algorithms and ADAM is 1×10^{-3}, as suggested in [96]; we selected it_max_Ca $= 1/\gamma/10 = 100$. SAG was more sensitive to the step size and we decreased it to 1×10^{-6} to preserve the convergence.

3.3.4 Discussion of results

The results are summarized in Figure 3.4. Overall CaBCD strongly outperforms the other methods and within the CaBCD variants the ones that use moments do better. Some further observations are that, firstly, the size of the blocks s has been fixed to two. The reason is that experimentally we have observed that if the block's size is between 2 and 5 the reduced measure is used for longer, i.e. the algorithm does more steps with the reduced measure, thus decreasing the runtime. Secondly, in the case of CaBCD

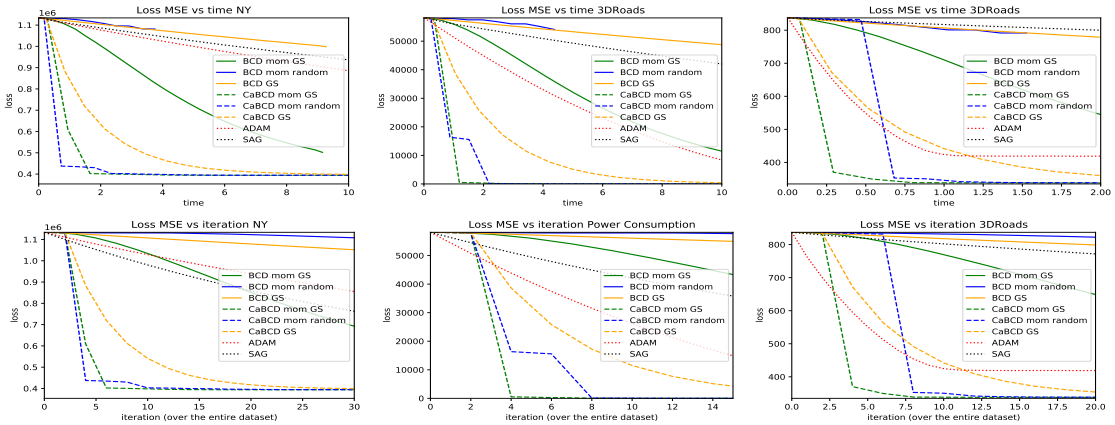


Figure 3.4: Running times and iterations of the different Algorithms. For *CaBCD mom GS*, *BCD mom GS* and *CaBCD mom random*, *BCD mom random* the directions have been computed using a standard momentum strategy, and chosen respectively by the GS rule and by the Random rule. For *CaBCD GS*, *BCD GS* the directions have been computed using the standard GD method, and chosen by the GS rule.

algorithms we count 1 iteration when a full gradient has been computed, while we count $\frac{\text{number of points in the reduced measure}}{N}$ for any iteration done with the reduced measure (if the size of the block is s, the reduced measure has support on $s+1$ points, see Theorem 2.2). An analogous reasoning is used to count the iterations of SAG and ADAM. Third, the CaBCD algorithms are for the first two iterations are "slower". This is due to the fact that we compute $\mathcal{H}$, i.e. the approximation of the second derivative. Finally, using the GS rule, the parallelisation of the code has often no effect because the directions to optimise belong to only one block.

3.3.5 Further Acceleration of the Carathéodory Block Coordinate Gradient Descent

The central question of BCD is the choice of the update rule. In the previous Section, we used the arguably simple ones, randomized and Gauss–Southwell, for CaBCD. However, more sophisticated update rules are possible which in turn could lead to a further performance improvement. The aim of this Section is to understand this better and therefore we revisit the study of different BCD rules of Nutini et al. [142] in the context of our CaBCD. To do

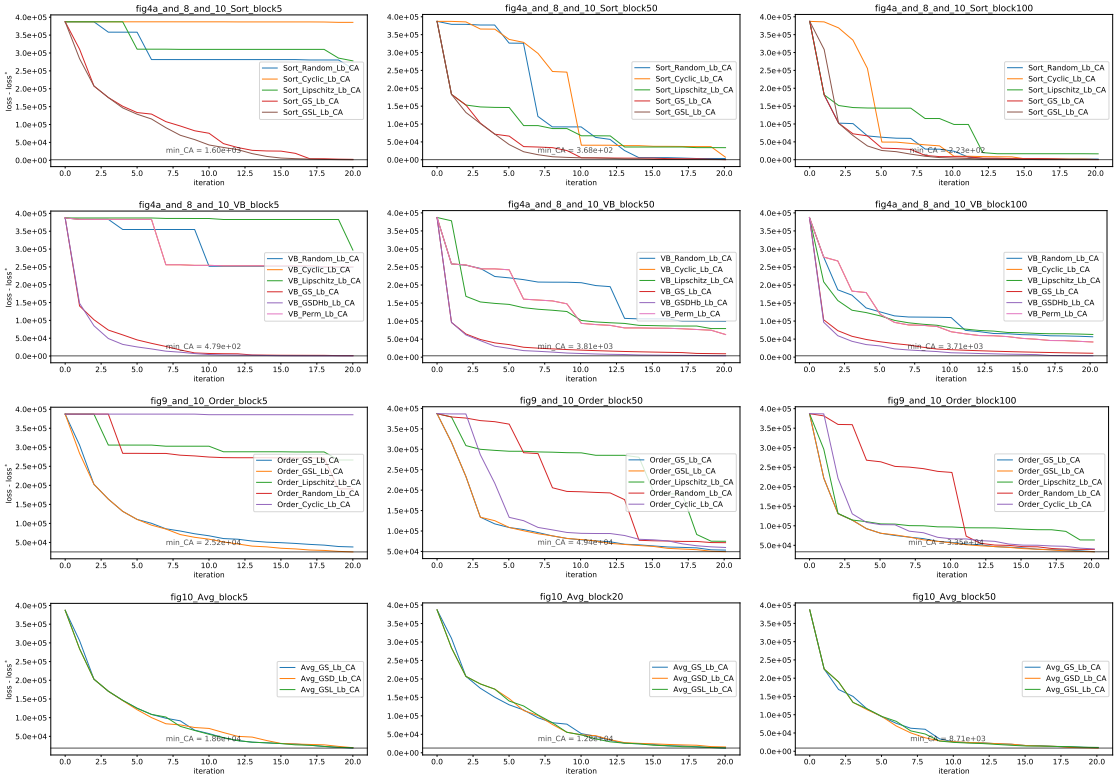


Figure 3.5: CaBCD applied with different block sizes, rules used in [142, Figure 4, 8, 10] (top-two lines), [142, Figure 9] (third line) and [142, Figure 10].

so we follow [142] and focus on a least-squares problem

$$\min_{\theta} \sum_{i} (x_i \theta^\top - y_i)^2.$$

We use [142, Dataset A] with $N = 1\,000\,000$ and $n = 500$. The data are generated following the same procedure explained in [142, Appendix F.1]. The x_i values are sampled from a standard normal random variable, then 1 is added to induce a dependency between columns and each column is multiplied by a sample from a standard normal random variable multiplied by ten, to induce different Lipschitz constants across the coordinates. Finally, each entry is kept non-zero with probability $10\log(m)/m$. $y_i = x_i \cdot \theta^\times + e_i$, where the e_i are drawn from a standard normal random variable. 90% of $\theta^\times$ is set to zero and the remaining values are sampled from a standard normal random variable.

BCD update rules

The rules presented in [142] can be represented as

$$\theta_{j+1} = \theta_j + \langle \Gamma, \sum_i \nabla L(\theta_j, (x_i,y_i)) \rangle,$$

where Γ can be a function of the Hessian of L, of Lipschitz bounds related to L, or however it can depend on $L(\theta_j, (x_i,y_i))$ non linearly in the data, e.g. the inverse of the Hessian. Due to the non-linearity, we compute the reduced measure for $\sum_i \nabla L(\theta_j, (x_i,y_i))$ and consider Γ as an independent factor. In general, Lipschitz bounds are difficult to find, whilst precise Hessian information is expensive computationally unless a closed formula is available, which is the case only for a small portion of models.

In [142] the step-size of the BCD is determined as a function of the computed Lipschitz bounds. While using the recombined measure we use a factor $\gamma = 1 \times 10^{-2}$, i.e.

$$\hat{\theta}_{j+1} = \hat{\theta}_j + \gamma \times \langle \Gamma, \sum_i \nabla L(\hat{\theta}_j, (\hat{x}_i,\hat{y}_i)) \rangle.$$

In place of $\mathbb{E}[\nabla L(\theta_j, (X,Y))] = \frac{1}{N} \sum_i \nabla L(\theta_j, (x_i,y_i))$, in [142] $\sum_i \nabla L(\theta_j, (x_i,y_i))$ is used, which results in higher loss values.

Compared to the previous experiments, we want to underline that some of the rules used in [142] compute precisely the Lipschitz Constants of the different blocks. Indeed, for least-squares problems Lipschitz constants can be written explicitly as a function of (x_i,y_i) and θ_j, see e.g. [142, Appendix B].

Dataset A used in [142] is synthetic and sparse. While the rules to select the directions of [142] prefer sparse matrices, we did not optimize the algorithms' code to find the reduced measures to efficiently deal with sparse datasets. Nevertheless, we can imagine improvements given a significant presence of matrices multiplications in the implementations.

3.3.6 A list of rules

We briefly introduce the rules below, for an exhaustive description we refer to [142]. We structure the experiments and plots as follows: any rule is represented by the following

string format *"partition_block-selection_direction"* with an additional suffix "*_CA*" to note that we have applied it with CaBCD. The possible choices for *partition*, *block-selection*, *direction* are[2]

$$\begin{aligned} \text{partition} &\in \{\text{VB, Sort, Order, Avg}\}, \\ \text{block-selection} &\in \{\text{Random, Cyclic, Lipschitz, Perm, GS, GSD, GSL,} \\ &\qquad \text{GSDHb, GSQ, IHT}\}, \\ \text{direction} &\in \{\text{Hb, Lb}\}. \end{aligned}$$

We give details on the choices below: VB stands for Variable Blocks which indicates that the partition of the directions can change at any iteration of the optimization procedure. Sort fixes the partition from the beginning, organizing the blocks of directions according to their Lipschitz values: the largest Lipschitz values into the first block, and so on. Order it fixes the partition from the beginning, subdividing the directions in order, e.g. if the block size is 2, the blocks will be $(1,2), (3,4)$, etc. Avg fixes the partition alternating between adding large and small Lipschitz values. Between the previous, VB is the only one that allows the partition to change between iterations. The "*block-selection*" rules prescribe how blocks are selected given the partition of the directions and we refer to [142] for details. The two choices of "*direction*" are "*Lb*" and "*Hb*". *Lb* means that the direction for the update is G_{block}/L_{block}; *Hb* signifies that the direction is $H_{block}^{-1} \cdot G_{block}$, where $L_{block}, G_{block}, H_{block}$ represent respectively the Lipschitz value, the Gradient and the Hessian of the chosen block.

The plots are named analogously to the plots in [142], but we additionally include the values of the size of the blocks. For the implementation of the blocks' selection rules, we have used the code[3] provided by the authors of [142].

[2]Not all the combinations are possible, see [142] and the official repository github.com/IssamLaradji/BlockCoordinateDescent from more details.

[3]Freely available at github.com/IssamLaradji/BlockCoordinateDescent.

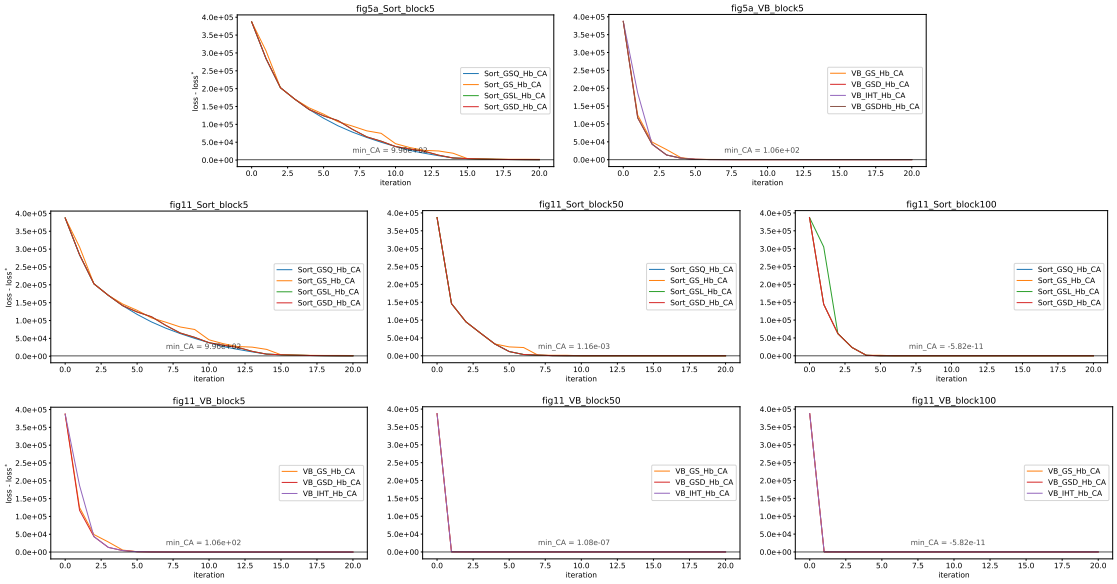


Figure 3.6: CaBCD applied with different block sizes, rules used in [142, Figure 5] (top-line) and [142, Figure 11].

3.3.7 Discussion of results

The results show that the general conclusion of [142] also applies to CaBCD. Firstly, from Figure 3.5 the GS based rules should be preferred when possible. Secondly, from Figure 3.5 it can be observed that between the partition rules we should prefer VB or Sort. In our experiments, the differences between the partition rules VB and Sort are less evident. In particular, we can notice that the VB partition rule attains its minimum loss when the block size is 5, which is congruent with our observation of Section 3.3.3 that the CaBCD makes more steps with the reduced measure when the block's size is low. Thirdly, from Figure 3.5 and 3.6 the differences between the selection rules vanish when the blocks' size increases. Lastly, the (quasi-)Newton updates Hb Figure 3.6 reach a lower minimum faster, as one can expect. However, we recall that the Carathéodory reduced measure matches only the gradient; thus a possible future research direction is to apply the Carathéodory Sampling "exactly" also to the second derivative, i.e. (quasi-)Newton methods.

3.4 Summary

We introduced a new approach to empirical risk minimization by gradient descent. Similar to SGD variants, CaGD replaces the sum of the expected gradient with a smaller sum. But in contrast to the usual SGD paradigm, the points are not random but deterministic and carefully selected by solving a recombination problem. A control statistic determines adaptively the descent steps at which we need to recombine again. To deal with high-dimensional optimization problems and reduce the computational complexity of the recombination steps we combined CaGD with BCD. Our experiments show that the resulting method, CaBCD, can lead to remarkable improvements compared to very competitive baselines such as ADAM and SAG. Many extensions are possible, e.g. on the theoretical side, studying the behaviour under non-convex losses and on the applied side, combination with Quasi-Newton methods, or BCD rules that are specialized to CaBCD. Independently of these, any improvement for recombination algorithms can lead to a further speed up of CaGD resp. CaBCD.

4 Recombined Lattice-Tree Models for Diffusions Approximation

Keywords: Multinomial Lattices, Tree Recombination, States Space Explosion, Discrete Space Markov chains

In Chapter 1, we have seen that discretizing the time, and possibly the space, is a well studied technique to approximate the law of stochastic processes. To show the weak convergence of an approximation scheme, one of the most used conditions to check is the "good" approximation of the moments of the increments of the original stochastic process, namely the local consistency, see Definition 4.4. This implies a significant problem if we are specifically interested in the discretization of the space: the number of states can explode exponentially with the number of steps done. Therefore, being able to build recombining trees is crucial to limit this behaviour - in one dimension [136] a tree is said to recombine if the number of states increases linearly with the number of steps done. In this Chapter, we find sufficient conditions to build approximations of stochastic processes on lattices that match exactly the first two moments. The recombining property of the resulting trees follows directly from the fact that the model is constrained to have support on a lattice. Our results do not need transformations and do not require the smoothness of the coefficients, see [136] and the literature below. We give general results for the construction of such approximation schemes in dimensions 1 (trinomial) and 2 (6-nomial), then we treat particular cases for a general dimension d. Moreover, the lattices we consider are universal [36], i.e. on the same lattice it is possible to build approximations for different stochastic processes. This Chapter contains the material presented in Cosentino et al. [43].

4.1 Introduction

Let W be a h-dimensional Brownian motion defined on a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t), \mathbb{P})$. Consider a d-dimensional Stochastic Differential Equation of the form

$$dX_t = \mu(X_t)dt + \sigma(X_t)dW_t, \qquad X_0 = x \in \mathbb{R}^d, \tag{4.1}$$

where $\mu : \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \to \mathbb{R}^{d\times h}$.

Definition 4.1. *A process $X^{s,x} \in \mathbb{R}^d$ is called solution of Equation* (4.1) *if $X^{s,x}$ is adapted to $(\Omega, \mathcal{F}, (\mathcal{F}_t), \mathbb{P})$, and*

$$X_t^{s,x} = x + \int_s^t \mu(X_u)du + \int_s^t \sigma(X_u)dW_u$$

where the last integral is the Ito version of the stochastic integral.

In general $X^{s,x}$ denotes the solution of (4.1) starting from x at time s. When there is no confusion, we denote $X^{s,x}$ as X and we use X^x if $s = 0$. We recall that if μ,σ are Lipschitz continuous and have bounded linear growth, then there exists a pathwise unique solution to Equation (4.1), see for example N. Ikeda [134]. For weaker conditions and weaker definition of Solution, we refer to [171].

As we have seen in Chapter 1, the law of X, see Definition 4.1, is typically not known in closed form, this requires to approximate X with discrete-time processes for which the quantities of interest can be computed on a computer. For many applications, this can be accomplished by discretizing the time but not the state space: for example, the Euler Scheme

$$X_{i+1}^n = X_i^n + n^{-1}\mu(X_i^n) + n^{-1/2}\sigma(X_i^n)W_i,$$

where W_i are independent Normal r.v. , approximates X by a Markov process that evolves in discrete time but a continuous state space [13, 175]. However, for other applications, it is preferable to discretize time and space so that the approximating process can be represented as a tree model; prominent examples include the Cox–Ross–Rubinstein [46] model for American option pricing or the work of Kushner [108] in optimal control. Also, in

Chapters 1 we have seen that the Formal Verification community is particularly interested in discrete-space discrete-time approximations scheme. Approximations of X by tree models are the topic of this Chapter: we construct a sequence of discrete-time and discrete state space Markov processes $(X^n)_{n\geq 1}$, such that: X^n converges weakly to X, the cardinality of the state space grows at most polynomially respect to the number of time steps, and the transition matrix is sparse.

Before we review the literature and summarize our approach, it is instructive to recall that naive discretization strategies generically fail to accomplish the goals simultaneously. If we consider the "Discrete" Euler Scheme,

$$X^n_{i+1} = X^n_i + n^{-1}\mu(X^n_i) + n^{-1/2}\sigma(X^n_i)B_i,$$

where B_i are independent Bernoulli r.v. and X^n_i indicates the process at the time $t_i = in^{-1}$, we can conclude that the scheme X^n converges weakly to X, see [98], Theorem 4.5. However, if μ, σ are not constant, the state space does not recombine, therefore we would have an exponential explosion of the state space, i.e. $\mathrm{card}[\mathrm{supp}(X^n_i)] = O(2^{hi})$.

Let us state the definition of weak convergence for the lattice schemes we aim to build.

Definition 4.2. *Let $(\gamma_n)_{n\geq 1} \subset (0,\infty)$ be a sequence that converges to 0. A sequence $(X^n)_{n\geq 1}$ of Markov processes $X^n = (X^n_i)_{i\geq 0}$ is a lattice approximation to X if $X^n_i \in \gamma_n\mathbb{Z}^d$ for all $n > 0$, $i \geq 0$ and (X^n) converges weakly to X respect to a class of functions $\mathbb{F}$, that is for every $f \in \mathbb{F}$, $t \in [0,T]$ $\lim_{n\to\infty} \mathbb{E}[f(X^n_{\lfloor nt/T \rfloor})] = \mathbb{E}[f(X_t)]$.*

As usual, the speed of convergence is quantified with the concept of rate.

Definition 4.3. *Let $(X^n)_{n\geq 1}$ be a lattice approximation to X. We say that $(X^n)_{n\geq 1}$ converges weakly with rate α respect to the class of functions $\mathbb{F}$ if for every $f \in \mathbb{F}$, $t \in [0,T]$*

$$\mathbb{E}[f(X_t)] - \mathbb{E}[f(X^n_{\lfloor nt/T \rfloor})] = O(n^{-\alpha}).$$

Literature. Tree approximations of stochastic processes have been well-studied in finance, where the main goal is to compute the price of European or American type derivatives. One of the eldest works in this direction is [46], where Cox, Ross and Rubinstein develop

the CRR model for option pricing and show the convergence to the Black-Scholes model [17]. CRR multivariate extensions are considered in Boyle et al. [26], Ekvall [53].

It is well known that discretizing the space can lead to an exponential explosion of the number of states needed. One of the first works that tries to mitigate this problem, and probably the most influential, is Nelson and Ramaswamy [136]. Nelson and Ramaswamy develop a rather general procedure to approximate an SDE using a recombining tree. A recombining tree is built when the number of states increases at most linearly with the number of steps. They work in the one-dimensional case and the fundamental idea is to use the Lamperti transform to obtain an SDE with constant volatility. Nelson and Ramaswamy show how to build a binomial tree when σ is constant. Finally, the inverse of the Lamperti transform is applied on the tree to "go back" to the original SDE. The main drawbacks in [136] are in order: the possibility of having negative probabilities, the necessity of using the Lamperti transform, which requires smoothness assumptions on the SDE coefficients, and being able to deal with $\int \sigma^{-1}(y)dy$. The idea presented in [136] has been refined and applied to particular instances: we cite Li et al. [117], Nawalkha and Beliaeva [135], Xu and Li [190] for the CIR and CEV models; for American options pricing see Finucane and Tomas [58]. The bivariate case of a Geometric Brownian Motion (GBM) is studied in Boyle [25] and for the extension to the multivariate case we refer to Gamba and Trigeorgis [65, 66]. Natural extensions to regime switching models where, for any regime, the process is a GBM or follows a "mean-reverting" behaviour with constant σ are studied in Liu and Zhao [120], Liu [121], Wahab and Lee [181].

Specific approaches for the construction of recombining trees have also been studied for Stochastic Volatility models (SVM), Florescu and Viens [59], Leisen [116], Moretto et al. [132]. In particular, Leisen [116], Moretto et al. [132] use the Lamperti transform. In Primbs et al. [150] it is shown how to build a pentanomial lattice to match the first four moments, but then they must remain constant in time and space, this is generalized in Lundengård et al. [124] to further moments.

In Ho et al. [81], the multivariate case is studied when the process to be approximated is assumed to have a log-normal distribution with known mean and volatility, possibly non-constant. However, their procedure requires splitting the time interval into n steps

and then any interval must be split into further n intervals, which implies $O(n^2)$ total time intervals. Then, for any of the n steps, a binomial tree on the respective n steps is used to approximate the wanted log-normal distribution. It is relevant to emphasise that the jumps of any respective trees can be different.

In Haahtela [75], Lok and Lyuu [122] is shown how to build a recombining tree for SDEs in dimension 1, with general instantaneous variance, possibly depending on time and space, but with a specified drift. Their method produces valid probabilities, i.e. non-negative, if the time discretization is fine enough. However, the tree considered in Haahtela [75], Lok and Lyuu [122] is model specific, following Chen and Yang [36] is not universal, i.e. changing the probabilities of the branches it is not possible to approximate a different SDE. From another point of view, we can say that the height of the jumps depends strongly on the value of the coefficients of the SDE.

Approximating SDEs without using the Lamperti transform has been also studied, but either it is assumed to be able to match exactly the moments on the chosen grid or to be able to approximate the moments up to the wanted precision. Then, numerical procedures are used to match the moments, Costabile et al. [44, 45], Ji and Brorsen [92], Leccadito et al. [114]. We show that this cannot be the case using a binomial approach, while it could be under some conditions if a multinomial procedure is considered on a lattice. Note that general numerical solutions, could lead theoretically to negative probabilities unless a constraint is added.

Other different approaches to build recombining trees relevant to mention are Rubinstein [159, 160], Simonato [167]. They focus on the one-dimensional case and they are based on numerical approximations and edgeworth expansions, or other approximations, of probability density functions. Although the approximation of probability density functions is an attractive idea, it is well-known, Cramér [48], that they are not density functions themselves. Furthermore, edgeworth-like expansions do not always converge.

We can summarise that in the literature tree-based approximations are well-studied for models such that, by means of transforms, can be seen as GBM or SVM in the one-dimensional case, or if the process is assumed to have a multivariate log-normal distribution. The jumps height of the trees depends strictly on the model considered, in particular on

σ. In this sense, the trees[1] are not universal Chen and Yang [36]. Often the moments are approximated through numerical procedures, e.g. numerical solutions to non-linear systems or solutions to quadratic optimization problems, which could lead to negative probabilities or moments too far from the original ones, since theoretical bounds are not available.

Formal Verification. As we have seen in Chapters 1 a discrete-time discrete-space approximating Markov chain is the desire object for the Formal Verification (FV) community. Let us suppose that the system we study is modelled by the SDE (4.1) and we are interested in the likelihood of the system to be safe. The safety problem is a central problem for FV community: given a safety region $A \in \mathbb{R}^d$ the desired object is the probability that the SDE leaves the region A in a specified time period $[0, T]$, i.e. if τ indicates the exit time the goal is the probability $\mathbb{P}(\tau \leq T)$. Remember that, the standard way to compute such probabilities for FV community is to use ad-hoc software, for instance PRISM [111], that can exploit particular structures of the transition matrix, e.g. sparsity. In this Chapter, we obtain Markov chains whose transition matrix is sparse and such that the state space increases polynomially, which is a necessary characteristic from a computational perspective when n increases.

Contribution and Outline In this work, we study sufficient conditions for the existence of universal lattices in the spirit of Chen and Yang [36]: let us consider the lattices $\gamma_n \mathbb{Z}^d$, given different processes $X_1, \ldots, X_m$, we find sufficient conditions for γ_n such that it is possible to build lattice approximations $(X_1^n, \ldots, X_m^n)_{n \geq 1}$ to $X_1, \ldots, X_m$ on $\gamma_n \mathbb{Z}^d$. In other terms, we can approximate different processes only by changing the branching probabilities. We generalize the results in [36] linking the parameters of the space and time discretizations, using all the possible combinations of moves on the lattice. Moreover, we can generalise to multi-dimensional cases. In dimensions 1 and 2, the proof is general; in dimension d, we treat particular cases. Although we do not show it, the reasoning of the proof of the case $d = 2$, see Theorem 4.18, could be generalised to general dimension d. Unfortunately, we

[1] We remark that the states of a recombined tree do not form a lattice unless the jumps of the tree are constant in time. Although this work does not require particular properties of lattices, the interested reader can refer to [152, Chapter 5] for an introduction.

have not found an elegant way to prove the general result for finite d, more insights are given in the next Sections. The aim of this Chapter is to construct lattice approximations that can be efficiently represented in memory. We focus on building approximations of processes with the least possible number of states that match exactly the increments' moments of the original processes. We do not consider numerical instability problems. In relation to this, we recall that tree approximations of stochastic processes are related to finite-difference methods from numerical PDE theory, Brennan and Schwartz [28], Hull and White [89].

Section 4.2 introduces the logic behind the construction of the approximations schemes in an informal way. Sections 4.3, 4.4 and 4.5 contain the main theoretical results and the division is based on the dimension treated: we study dimension 1 and 2 without specific assumptions on the form of μ and σ, whilst the general dimension d is more restrictive. For any dimension examined, we present the solution to a more theoretical problem and then apply it to the approximation of stochastic processes. In Section 4.6, we show how to use the idea of transforming the original SDE, as in [136], and show the advantages we have compared to them. Moreover, we introduce also possible extensions, e.g. optimal stopping problems, easily obtainable by the theory here presented. Finally, in Section 4.7 we present the Algorithm and consider practical aspects from a computational point of view, then we show the results of a toy example.

4.2 Background and Method

In this Section, we outline the basic idea of our construction. We recall that $C_P^l(E,F)$ indicates the space of functions from E to F l times continuously differentiable with polynomial growth, including the derivatives; similarly $C_b^l(E,F)$ indicates the space of functions from E to F l times continuously differentiable with linear growth, with uniformly bounded derivatives.

4.2.1 Diffusion Approximations

Under mild assumptions, e.g. $\mu, \sigma \in C_b^4$, an application of the Ito's rule[2] implies that for some $q \geq 1$, $\alpha > 0$, $n > N$, $c > 0$ depending only on μ,σ the solution X of the SDE (4.1) fulfils

$$\begin{aligned} |\mathbb{E}[X_{t+n^{-1}} - X_t | X_t = x] - \mu(x)n^{-1}| \leq & c(1+|x|^q)n^{-1-\alpha} \\ |\mathbb{E}[(X_{t+n^{-1}} - X_t - n^{-1}\mu(X_t))^{\otimes 2} | X_t = x] - \Sigma(x)n^{-1}| \leq & c(1+|x|^q)n^{-1-\alpha}, \end{aligned}$$

where $\Sigma(x) = \sigma(x)\sigma(x)^\top$. These two estimates characterize the SDE and locally matching these two moments with a sufficient precision guarantees weak convergence.

Definition 4.4. *[Local consistency [98, page 328]] Let $(X^n)_{n\geq 1}$ be a sequence of discrete time Markov chains in $\mathbb{R}^d$. We say that X^n satisfies the local consistency assumption if there exists a function $\delta(n)$ such that $\delta(n) \to 0$ as $n \to \infty$ and for every $i \leq n$*

$$\begin{aligned} \mathbb{E}\left(\mathbb{E}\left[\frac{X_{i+1}^n - X_i^n}{n^{-1}}\middle| X_i^n\right] - \mu(X_i^n)\right)^2 = & O(\delta(n)) \\ \mathbb{E}\left(\mathbb{E}\left[\frac{(X_{i+1}^n - X_i^n)^{\otimes 2}}{n^{-1}}\middle| X_i^n\right] - \Sigma(X_i^n)\right)^2 = & O(\delta(n)) \end{aligned} \tag{4.2}$$

Theorem 4.5. *[98, Theorem 9.7.4] Let us assume that X^n is local consistent and*

$$\mathbb{E}[|X_{i+1}^n - X_i^n|^6] \leq \delta(n)n^{-2}, \qquad \mathbb{E}[\max_i |X_i^n|^{2q}] \leq c(1+|X_0|^{2q}), \qquad \text{for } q \geq 1,$$

where $\delta(n)$ is as in Definition 4.4. If the diffusion coefficients μ, σ of Equation (4.1) are in C_b^4, then[3]

$$|\mathbb{E}f(X_n^n) - \mathbb{E}f(X_T)| \leq c \cdot \left(\min\left\{\sqrt{\delta(n)}, n^{-1/2}\right\}\right), \quad f \in C_P^3(\mathbb{R}^d, \mathbb{R}). \tag{4.3}$$

With some work it can be shown that the convergence criterion of Equation (4.3) is equivalent to the convergence criterion of Definitions 4.2 and 4.3.

[2] See for example [98]. For weaker sufficient conditions see [171].

[3] The rate of convergence is stated in the proof of [98, Theorem 9.7.4].

Remark 4.6. *Many similar results in the literature apply under different regularity assumptions, see for example Ethier and Kurtz [55], Stroock and Varadhan [171] for general results regarding Markov processes. The main results in Kushner [108], Kushner and Dupuis [109] are also in this direction, but generalising to controlled SDEs and optimal stopping problems, we examine these problems in Section 4.6.2. The cited works do not state the order of convergence and require conditions similar to the local consistency one, Definition 4.4, however often the smoothness assumptions on μ, σ, f are weaker. In this Chapter, we state the main results using as baseline the results in [98], but it is important to underline that the procedure we implement to build the approximation scheme X^n, such that the local consistency condition is satisfied, is independent of the smoothness assumptions required by the convergence Theorem 4.5. Thus, using the results in [55, 108, 109, 171] it might be possible to weaken the smoothness assumptions on f, μ, σ.*

4.2.2 Matching Moments with Carathéodory's Theorem

The previous subsection shows that it is sufficient to match locally the moments of the increments of the diffusion to construct a Markov chain approximation for the solution of an SDE. To make this computationally efficient, we should aim to do this with a random variable that has small support. Hence, our aim is to construct a discrete probability measure $\sum_{k=1}^{r} \delta_{l_k} p_k$ supported on a lattice such that for an increment

$$\text{Law}(X_{i+i}^n - X_i^n) = \sum_{k=1}^{r} \delta_{l_k} p_k$$

and such that $\sum_{k=1}^{r} \delta_{l_k} p_k$ satisfies the local consistency conditions. More generally, we can formulate these requirements on (l_k,p_k) as being solutions of the non-linear system

$$\begin{cases} \sum_{i=1}^{r} p_i l_i = a \\ \sum_{i=1}^{r} p_i l_i^{\otimes 2} = a^{\otimes 2} + B, \end{cases} \tag{4.4}$$
$$\text{subject to} : p_k \geq 0, \quad \sum_{k=1}^{r} p_k = 1, \quad l_i \in \gamma Q \mathbb{Z}^d$$

where $a \in \mathbb{R}^d$ and $B \in \mathbb{R}^{d\times d}$ symmetric and positive semi-definite. Ideally, we would like γ to be the biggest possible, because this would implies less necessary points in any compact region to approximate a stochastic process, without neglecting the "quality" of the convergence. The role of the matrix $Q \in \mathbb{R}^{d\times d}$ will become clear later but we consider two cases: $Q = I$ or Q such that $B = Q^T \Lambda Q$, where Λ is a diagonal matrix whose elements are the eigenvalues λ_i of B, and therefore Q is the orthonormal matrix whose columns are the eigenvectors q_i of B. Note that such a decomposition exists because B is symmetric and positive semi-definite. Stated in this way, it should be evident the relation with the recombination problem we studied in Chapter 2. We re-state here for convenience the Carathéodory's Theorem that will be used also in the following.

Theorem 4.7 (Carathéodory [51])**.** *Let $\{x_i\}_{i=1}^N$ be a set of N points in $\mathbb{R}^n$ and $N > n+1$. Any point z that lies in the convex hull of these N points, z can be expressed as a convex combination of maximum $n+1$ points, i.e. $z = \sum_{j=1}^{n+1} p_j x_j^\star$, where $x_j^\star \in \{x_i\}_{i=1}^N$ and $0 \le p_i \le 1$, $\sum_i p_i = 1$.*

4.2.3 Constructing Lattice-Tree models

The key idea to construct a lattice approximation X^n to X is to specify the transition matrix $(P_{x,y}^{(n)}) \coloneqq \mathbb{P}(X_{i+1}^n = y | X_i^n = x)$ inductively. Let $X_0 = x$ a.s., we define $X_0^n \coloneqq x$ and use the results of the next Sections to construct a random variable Y_x such that

$$\mathbb{E}[Y_x] \approx \mu(x)n^{-1}, \qquad \mathbb{E}[Y_x^{\otimes 2}] \approx \mu(x)^{\otimes 2} n^{-2} + \Sigma(x)n^{-1},$$

solving system (4.4) with a lattice-valued random variable Y_x that has small support. We set $P_{x,y}^{(n)} \coloneqq \mathbb{P}(Y_x = y - x)$ which determines the row $P_{x,:}^{(n)}$. For each y such that $P_{x,y}^{(n)} \neq 0$ we then repeat the same procedure for the row $P_{y,:}^{(n)}$, see Algorithm 7. Since each Y_x has a small support and the support is constrained on a lattice $\gamma_n Q\mathbb{Z}^d$, the transition matrix $P^{(n)}$ results to be sparse, and by construction the moments of the increments are matched:

$$\mathbb{E}[X_1^n - X_0^n | X_0^n = x] \approx \mu(x)n^{-1}, \quad \mathbb{E}[(X_1^n - X_0^n)^{\otimes 2} | X_0^n = x] \approx \mu(x)^{\otimes 2} n^{-2} + \Sigma(x)n^{-1}.$$

Moreover, since the r.v. $Y_{(\cdot)}$ are built on a lattice $\gamma\mathbb{Z}^d$, the final Markov chain X^n has support on the lattice $\gamma\mathbb{Z}^d$ as well - some attention is needed for the first step if $X_0 \notin \gamma\mathbb{Z}^d$. Below we make the above ideas rigorous.

4.3 Lattice-Tree Models for One-Dimensional Diffusions

We now apply the two-steps methodology outlined in Section 4.2 to one-dimensional SDEs. The first step establishes the existence of a random variable that matches (or closely approximates) any given first two moments and is additionally supported on a lattice. The second step uses this random variable to define the increments of a tree approximation.

Matching the Moments

For $\gamma > 0$, $a \in \mathbb{R}$ denote

$$\lfloor a \rfloor_\gamma := \max\{\gamma z, z \in \mathbb{Z} \ s.t. \ \gamma z \leq a\}, \qquad \lceil a \rceil_\gamma := \min\{\gamma z, z \in \mathbb{Z} \ s.t. \ \gamma z > a\}.$$

If $\gamma = 1$ we suppress the subscript γ and the above reduces to the standard notation for the ceiling and floor functions.

Theorem 4.8. *Let a,$b \in \mathbb{R}$, $b \geq 0$, and $\gamma > 0$. Then there exists a random variable Y such that*

1. $\mathbb{E}[Y] = a$ *and* $|\mathbb{E}[Y^2] - a^2 - b^2| \leq \frac{\gamma^2}{4}$,
2. $\text{supp}(Y) \subset \{y \in \gamma\mathbb{Z} : |y| \leq \sqrt{a^2 + b^2} + \gamma\}$.

Moreover, if

$$a^2 + b^2 \geq (\lceil a \rceil_\gamma^2 - \lfloor a \rfloor_\gamma^2)(a - \lfloor a \rfloor_\gamma)\gamma^{-1} + \lfloor a \rfloor_\gamma^2,$$

then one can additionally assume that $\mathbb{E}[Y^2] = a^2 + b^2$.

Proof. Wlog suppose $a \geq 0$. For a discrete random variable Y denote $p_i := \Pr(Y = y_i)$. If

$$\mathbb{E}[(Y, Y^2)] = \sum_i p_i(y_i, y_i^2) = (a, m) \tag{4.5}$$

for a m such that $|m-(a^2+b^2)| \leq \gamma^2/4$ and $y_i \in \gamma\mathbb{Z}$ then the result follows. We distinguish three cases depending on the position of (a,a^2+b^2) in relation to the points $(\lceil a \rceil_\gamma, \lceil a \rceil_\gamma^2)$, $(\lfloor a \rfloor_\gamma, \lfloor a \rfloor_\gamma^2)$ as depicted in Figure 4.1. Formally,

$$\begin{cases} \text{Case 1} & : a^2 \leq a^2+b^2 \leq \frac{\lceil a \rceil_\gamma^2 - \lfloor a \rfloor_\gamma^2}{\lceil a \rceil_\gamma - \lfloor a \rfloor_\gamma}(a-\lfloor a \rfloor_\gamma) + \lfloor a \rfloor_\gamma^2 , \\ \text{Case 2} & : \frac{\lceil a \rceil_\gamma^2 - \lfloor a \rfloor_\gamma^2}{\lceil a \rceil_\gamma - \lfloor a \rfloor_\gamma}(a-\lfloor a \rfloor_\gamma) + \lfloor a \rfloor_\gamma^2 \leq a^2+b^2 \leq \lceil a \rceil_\gamma^2 , \\ \text{Case 3} & : \lceil a \rceil_\gamma^2 \leq a^2+b^2 \end{cases} \tag{4.6}$$

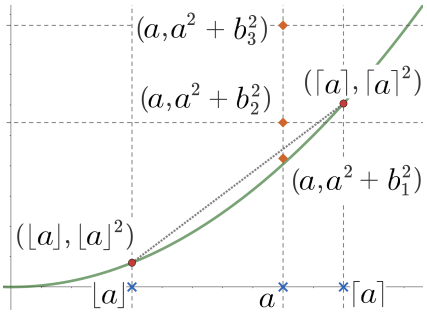


Figure 4.1: Proof of Theorem 4.8. b_i represents b under the hypothesis of Case i.

Case 1. We claim that we can choose $y_1 = \lfloor a \rfloor_\gamma$, $y_2 = \lceil a \rceil_\gamma$, $p_1 = \frac{a-\lfloor a \rfloor_\gamma}{\lceil a \rceil_\gamma - \lfloor a \rfloor_\gamma}$ and $p_2 = 1-p_1$. This gives $\mathbb{E}[Y] = a$ and the bound on $|\mathbb{E}Y^2 - a^2 - b^2|$ follows from a direct calculation:

$$\begin{aligned} \mathbb{E}[Y^2] =& \frac{a-\lfloor a \rfloor_\gamma}{\lceil a \rceil_\gamma - \lfloor a \rfloor_\gamma} \lceil a \rceil_\gamma^2 + \left(1 - \frac{a-\lfloor a \rfloor_\gamma}{\lceil a \rceil_\gamma - \lfloor a \rfloor_\gamma}\right) \lfloor a \rfloor_\gamma^2 = \frac{a-\lfloor a \rfloor_\gamma}{\gamma}(\lceil a \rceil_\gamma^2 - \lfloor a \rfloor_\gamma^2) + \lfloor a \rfloor_\gamma^2 \\ =& \frac{a-\lfloor a \rfloor_\gamma}{\gamma}((\lfloor a \rfloor_\gamma + \gamma)^2 - \lfloor a \rfloor_\gamma^2) + \lfloor a \rfloor_\gamma^2 = (a-\lfloor a \rfloor_\gamma)(\gamma + 2\lfloor a \rfloor_\gamma) + \lfloor a \rfloor_\gamma^2 . \end{aligned}$$

Using that b satisfies the inequalities (4.6)-Case 1, gives

$$\begin{aligned} \mathbb{E}[Y^2] - a^2 - b^2 =& (a-\lfloor a \rfloor_\gamma)(\gamma + 2\lfloor a \rfloor_\gamma) + \lfloor a \rfloor_\gamma^2 - a^2 - b^2 \\ \geq& (a-\lfloor a \rfloor_\gamma)(\gamma + 2\lfloor a \rfloor_\gamma) - \frac{\lceil a \rceil_\gamma^2 - \lfloor a \rfloor_\gamma^2}{\gamma}(a-\lfloor a \rfloor_\gamma) \\ \geq& (a-\lfloor a \rfloor_\gamma)(\gamma + 2\lfloor a \rfloor_\gamma) - (\gamma + 2\lfloor a \rfloor_\gamma)(a-\lfloor a \rfloor_\gamma) \geq 0. \end{aligned}$$

Denoting $\theta := (a - \lfloor a \rfloor_\gamma)/\gamma \in [0,1]$ we write

$$\begin{aligned}\mathbb{E}[Y^2] - a^2 - b^2 =& (a - \lfloor a \rfloor_\gamma)(\gamma + 2\lfloor a \rfloor_\gamma) + \lfloor a \rfloor_\gamma^2 - a^2 - b^2 \\ \leq& (a - \lfloor a \rfloor_\gamma)(\gamma + 2\lfloor a \rfloor_\gamma) + \lfloor a \rfloor_\gamma^2 - a^2 \\ \leq& \theta\gamma(\gamma + 2\lfloor a \rfloor_\gamma) + \lfloor a \rfloor_\gamma^2 - (\lfloor a \rfloor_\gamma + \theta\gamma)^2 \leq \theta\gamma^2(1-\theta) \leq \frac{\gamma^2}{4}\end{aligned}$$

Case 2. The point $(a, a^2 + b^2)$ is contained in the convex hull spanned by the four points[4]

$$\{(y,y^2) : y \in \{\pm\lceil a \rceil_\gamma, \pm \lfloor a \rfloor_\gamma\}\}. \tag{4.7}$$

By Carathéodory's Theorem 2.2, $(a, a^2 + b^2)$ is a convex combination of three points of the set (4.7). Hence, (4.5) holds with $m = a^2 + b^2$.

Case 3. Since $a^2 + b^2 \leq \left\lceil \sqrt{a^2 + b^2} \right\rceil_\gamma^2$ the point $(a, a^2 + b^2)$ is contained in the convex hull spanned by the four points $\{(y,y^2) : y \in \{\pm\lceil a \rceil_\gamma, \pm\left\lceil \sqrt{a^2 + b^2} \right\rceil_\gamma^2\}\}$ and by Carathéodory's Theorem 2.2 we conclude.

To finish the proof, we need the bounds on the support of the r.v. Y, however it is sufficient to note that this is true by construction. □

For special cases of a,b one gets stronger results (with simpler proofs).

Corollary 4.9. *Let $b \geq 0$, and $\gamma > 0$. Then there exists a random variable Y with*

1. $\mathbb{E}[Y] = 0$ *and* $\mathbb{E}[Y^2] = b^2$,
2. $\operatorname{supp}(Y) \subset \{y \in \gamma\mathbb{Z} : |y| \leq b + \gamma\}$.

Proof. Let us call $a_{++} = \min\{\gamma z > 0, z \in \mathbb{Z} \text{ s.t. } \gamma^2 z^2 > b^2\}$, then one possible solution (but not the only one) is the r.v. Y with support on $\pm a_{++}$ with probability $b^2/(2a_{++}^2)$ and 0 with probability $1 - b^2/a_{++}^2$. □

Corollary 4.10. *Let $a, b \in \mathbb{R}$, $b \geq 0$, and $\gamma > 0$. Then there exists a r.v. Y with*

1. $\mathbb{E}[Y] = a$ *and* $\mathbb{E}[Y^2] = a^2 + b^2$;

[4]Other solutions can exist.

2. $\text{supp}(Y) \subset \{y \in \gamma\mathbb{Z} \cup \{a\} \,:\, |y| \leq \sqrt{a^2+b^2} + \gamma\}$.

Proof. It should be clear from Figure 4.1 that adding a to the lattice $\gamma\mathbb{Z}$, the convex hull of the 6 points $(\pm\lfloor a\rfloor_\gamma, \lfloor a\rfloor_\gamma^2), (\pm a, a^2), (\pm\lceil a\rceil_\gamma, \lceil a\rceil_\gamma^2), (\pm\lceil\sqrt{a^2+b^2}\rceil_\gamma, \lceil\sqrt{a^2+b^2}\rceil_\gamma^2)$ contains the point (a, a^2+b^2) and the Caratheodory's Theorem 2.2 concludes the proof. □

Remark 4.11. *In some applications, it can be required to constrain the support of the r.v. on specific intervals, e.g. in finance the price quantity should be non-negative. In this case, from Figure 4.1, it should be clear that one could proceed looking for the smallest point* $\gamma z_\star$ *such that the point* (a, a^2+b^2) *lays in the convex hull of the points* (γ, γ^2), $(\lfloor a\rfloor_\gamma, \lfloor a\rfloor_\gamma^2)$, $(\lceil a\rceil_\gamma, \lceil a\rceil_\gamma^2)$ *and* $(\gamma z_\star, \gamma^2 z_\star^2)$.

Constructing the Lattice-Tree Model

Now we are ready to build the Markov chain X^n which approximates X and such that the state space recombines if the SDE coefficients are bounded.

Definition 4.12. *The approximation scheme* X^n *recombines if* $\text{card}[\text{supp}(X_i^n)] = O(i^d)$.

Nelson and Ramaswamy [136] introduce the definition for the case $d = 1$, we extend here to dimensions greater than 1, see the next Sections.

Theorem 4.13. *Let* $d = 1$ *and* $\mu, \sigma \in C_b^4$, $\beta > 0$ *and* $\sigma_{\min}^2 := \inf_x \sigma^2(x)$. *Set*

$$\gamma_n := \begin{cases} 2n^{-1/2}\sigma_{\min} & \text{, if } \sigma_{\min} > 0, \\ \sqrt{n^{-1-\beta}}, & \text{, otherwise.} \end{cases}$$

Then there exists a lattice approximation $(X^n)_{n\geq 1}$ *to* X *on* $X_0 \cup \gamma_n\mathbb{Z}$ *respect to the class of functions* $f \in C_P^3$ *with rate* $1/2$ *if* $\sigma_{\min} > 0$, *or* $\min(\beta, \frac{1}{2})$ *otherwise. Moreover*

$$\begin{cases} \text{card}[\text{supp}(X_i^n)] \leq \frac{ic n^{-1/2}}{\gamma_n} + ic & \text{, if } \mu,\sigma \text{ are bounded,} \\ \text{card}[\text{supp}(X_i^n)] \leq \frac{i(cn^{-1/2}+\gamma_n)\exp(cin^{-1/2})}{\gamma_n} & \text{, if } \mu,\sigma \text{ have linear growth.} \end{cases}$$

Note that, if μ, σ *are bounded and* $\sigma_{\min} > 0$, *then* $\text{card}[\text{supp}(X_i^n)] \leq ic$, *i.e.* X^n *recombines.*

The proof below allows to state the same result but with the lattice approximation supported on the "shifted lattice" $X_0 + \gamma_n\mathbb{Z}$ instead of $X_0 \cup \gamma_n\mathbb{Z}$. However, an advantage of the above formulation is that several diffusions can be supported on the same lattice, namely $\gamma_n\mathbb{Z}$ and the union of their starting points, whereas shifting the lattice would require to build a complete new lattice every time a diffusion is added. We prepare the proof of Theorem 4.13 with the following Lemma.

Lemma 4.14. *If there exists $\alpha > 0$ and a $q \geq 1$ such that for $m = 1, 2$*

1. $\left|\mathbb{E}[(X_{i+1} - X_i)^{\otimes m} | X_i = X_i^n] - \mathbb{E}[(X_{i+1}^n - X_i^n)^{\otimes m} | X_i^n = X_i^n]\right| \leq c(1 + |X_i^n|^q)n^{-1-\alpha}$,
2. $\mathbb{E}[\max_i |X_i^n|^q] \leq c(1 + |X_0|^q)$,
3. $\mathbb{E}[|X_{i+1}^n - X_i^n|^6 | X_i^n] \leq c(1 + |X_i^n|^q)n^{-2-2\alpha}$,

then X^n converges weakly with rate $\min\{\alpha, \frac{1}{2}\}$ to X for $f \in C_P^3$.

Proof. The first two items imply the local consistency (4.2) for $\delta(n) = n^{-2\alpha}$, see Appendix, Lemma 4.24. Similarly, the second and third items imply the conditions required by Theorem 4.5, in particular the third item, thanks to item 2, implies $\mathbb{E}[|X_{i+1}^n - X_i^n|^6] \leq \delta(n)n^{-1}$ with $\delta(n) = n^{-2\alpha}$. □

Proof of Theorem 4.13. We first deal with the strictly elliptic case when $\sigma^2_{\min} := \inf_x \sigma^2(x) > 0$. Theorem 4.8 applied with $a = \mu(x)n^{-1}$ and $b(x) = \sigma(x)n^{-1/2}$ guarantees the existence of a random variable Y_x such that

$$\mathbb{E}[Y_x] = \mu(x)n^{-1}, \quad \mathbb{E}[Y_x^2] = \mu^2(x)n^{-2} + \sigma^2(x)n^{-1} \tag{4.8}$$

and such that

$$\mathrm{supp}(Y_x) = \left\{ z\gamma_n, z \in \mathbb{Z} \text{ s.t. } |z|\gamma_n \leq \sqrt{\mu(x)^2 n^{-2} + \sigma(x)^2 n^{-1}} + \gamma_n \right\}, \tag{4.9}$$

if we can show that

$$a^2 + b^2 \geq (\lceil a \rceil_{\gamma_n}^2 - \lfloor a \rfloor_{\gamma_n}^2)(a - \lfloor a \rfloor_{\gamma_n})\gamma_n^{-1} + \lfloor a \rfloor_{\gamma_n}^2 . \tag{4.10}$$

Substituting (4.10) yields

$$\frac{\sigma(x)^2}{n} \geq \frac{\lceil \frac{\mu(x)}{n} \rceil_{\gamma_n}^2 - \lfloor \frac{\mu(x)}{n} \rfloor_{\gamma_n}^2}{\gamma_n} \left(\frac{\mu(x)}{n} - \lfloor \frac{\mu(x)}{n} \rfloor_{\gamma_n} \right) + \lfloor \frac{\mu(x)}{n} \rfloor_{\gamma_n}^2 - \frac{\mu^2(x)}{n^2} = \theta \gamma_n^2 (1-\theta) \quad (4.11)$$

where θ in the last equality is defined as $\theta \coloneqq (\frac{\mu(x)}{n} - \lfloor \frac{\mu(x)}{n} \rfloor_{\gamma_n}) \gamma_n^{-1}$. Since $\theta \in [0,1]$, choosing $\gamma_n \leq 2n^{-1/2}\sigma_{\min}$ implies (4.11) because $\theta(1-\theta) \in [0, 1/4]$ if $\theta \in [0,1]$. Hence, the existence of the random variable Y_x matching the moments (4.8) and with support on the grid (4.9) follows. By applying this construction to different x we can define a Markov chain X^n as described in Algorithm 7 by inductively filling out the transition matrix $P^{(n)}$.

If X_0^n is not in the lattice $\gamma_n\mathbb{Z}$ some attention is needed: to have X_1^n in $\gamma_n\mathbb{Z}$ it is necessary to build $Y_{X_0^n}$ on the lattice $\gamma_n\mathbb{Z} - X_0^n$, indeed $X_1^n = X_0^n + Y_{X_0^n}$ would be in $\gamma_n\mathbb{Z}$. This is also equivalent to build the r.v. $\tilde{Y}_{X_0^n}$ on $\gamma_n\mathbb{Z}$ s.t. $\mathbb{E}\tilde{Y}_{X_0^n} = \mu(X_0^n) + X_0^n$ and $\mathbb{E}\tilde{Y}_{X_0^n}^2 = \sigma(X_0^n)^2 + (\mu(X_0^n) + X_0^n)^2$ and define $Y_{X_0^n} \coloneqq \tilde{Y}_{X_0^n} - X_0^n$ with support on $\gamma_n\mathbb{Z} - X_0$ s.t. $\mathbb{E}Y_{X_0^n} = \mu(X_0^n)$ and $\mathbb{V}Y_{X_0^n} = \sigma(X_0^n)^2$.

The resulting Markov chain X^n evolves in the state space $\gamma_n\mathbb{Z}$ and it only remains to show the claimed convergence rate and growth of the support as the time progresses. We do this by showing that the assumptions of Lemma 4.14 apply. Lemma 4.14-Item (1) holds by (4.8). Lemma 4.14-Item (2) holds in great generality for Markov chains following this construction, see Lemma 4.25. For the remaining Lemma 4.14-Item (3) note that for some $z_j \in \mathbb{Z}$

$$\mathbb{E}\,|Y_x|^6 = \sum_{j=1}^{3} p_j |2n^{-1/2}\sigma_{\min} z_j|^6 \leq c \left| n^{-6}\mu(x)^6 + n^{-3}\sigma(x)^6 + 2^6 n^{-3}\sigma_{\min}^3 \right| \leq c(1+|x|^6)n^{-3},$$

which we can apply to $x = X_i^n$ and shows that the assumptions of Lemma 4.14 are satisfied which in turn finishes the proof of the convergence rate.

When μ, σ have bounded linear growth, to bound the growth of the support we use the discrete Grönwalls Inequality 4.26,

$$|X_i^n| \leq \sum_{j=0}^{i-1} \left| Y_{X_j^n} \right| \leq \sum_{j=0}^{i-1} n^{-1/2} c(1 + |X_j^n|) + \gamma_n \leq (cn^{-1/2}i + i\gamma_n) \exp(cin^{-1/2}).$$

Since the points $l_k \in \gamma_n\mathbb{Z}$ such that $|l_k| \leq c$ are c/γ_n we have that

$$\text{card}[\text{supp}(X_i^n)] \leq \frac{i(cn^{-1/2} + \gamma_n)\exp(cin^{-1/2})}{\gamma_n}.$$

If μ, σ are bounded, it is easy to obtain through similar reasoning that

$$|X_i^n| \leq \sum_{j=0}^{i-1} \left|Y_{X_j^n}\right| \leq icn^{-1/2} + i\gamma_n$$

which implies that

$$\text{card}[\text{supp}(X_i^n)] \leq \frac{icn^{-1/2}}{\gamma_n} + ic.$$

This finishes the proof of the elliptic case $\sigma^2_{\min} > 0$.

The non-elliptic case, $\inf \sigma^2(x) = 0$, follows analogously: as before we rely on Theorem 4.8 that ensures the existence of a random variable Y_x such that

$$\mathbb{E}[Y_x] = n^{-1}\mu(x), \quad \mathbb{E}[Y_x^2 - n^{-2}\mu^2(x) - n^{-1}\sigma^2(x)] \leq \frac{\gamma_n^2}{4}$$

and that is supported on $\gamma_n\mathbb{Z}$. Note that unlike in the elliptic case, this time we do not match the second moment exactly and the second moment condition shows that for any any $\beta > 0$ we can take $\gamma_n \leq n^{\frac{-1-\beta}{2}}$.

For the rates of convergence it is enough to observe that $\sqrt{\delta(n)}$ of Theorem 4.5 is equal to β due to approximation of the second moment, see Lemma 4.14-Item (1). Moreover, remember that the rate of convergence in Theorem 4.5 is bounded by 1/2.

Note that both the choices of γ_n satisfy $n\gamma_n^2 < c$ for some $c > 0$, as required by Lemma 4.25. □

For the special case $\mu(x) = 0$ we can prove the following Corollary.

Corollary 4.15. *Let $d = 1$, $\mu = 0$, $\sigma \in C_b^4$ and for any $c > 0$ set $\gamma_n = cn^{-1/2}$. If $X_0^n \in \gamma_n\mathbb{Z}$, then there exists a lattice approximation $(X^n)_{n\geq 1}$ to X on $\gamma_n\mathbb{Z}$ respect to the class of functions $f \in C_P^3$ with rate 1/2. Moreover, the same bounds on the cardinality of the support of Theorem 4.13 hold true.*

Proof. If $\mu(x) = 0$ we can use Corollary 4.9 and then follow the same reasoning of the previous proof. Since the first two moments can be approximated exactly independently of the lattice considered, we can choose $\gamma_n = cn^{-1/2}$ for any $c > 0$. The dependence of γ_n on n is chosen because required by Lemma 4.25: γ_n must satisfy $n\gamma_n^2 < c$ for some $c > 0$. □

4.4 Lattice-Tree Models for Two-Dimensional Diffusions

We now carry out the same procedure in the two-dimensional case. The key difference to the one-dimensional case is in the first step: the existence of a random variable that matches the moments closely and is supported on a lattice is much more involved. Nevertheless, we show that by using the eigenvalues of the matrix B the system (4.4) can be solved.

Matching the Moments

Throughout this Section we fix $a \in \mathbb{R}^d$ and $B \in \mathbb{R}^{d\times d}$. We assume that B is symmetric and positive definite and denote the eigenvalues of B with $(\lambda_i)_i$ ordered from biggest to smallest, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{\min}$ with $\lambda_{\min}$ denoting the smallest eigenvalue. We need the following two theorems to prove the main statements of this Section.

Theorem 4.16. *[Hoffman and Wielandt [82]] If A and B are normal matrices with eigenvalues $\lambda_i[A]$ and $\lambda_j[B]$, then there exists a suitable numbering of the eigenvalues s.t.*

$$\sum_i |\lambda_i[A] - \lambda_i[B]|^2 \leq |A - B|^2$$

Theorem 4.17. *[Gerschgorin Circle Theorem [67]] Let $B \in \mathbb{R}^{d\times d}$ with entries b_{ij}, then every eigenvalues λ_i of B lies in at least one of the Gershgorin circle*

$$\mathit{Circle}(b_{ii}) := \{c \in \mathbb{C} : |c - b_{ii}| \leq \sum_{j\neq i} |b_{ij}|\}$$

Now we are ready to prove the main statement.

Theorem 4.18. *Let $d = 2$, $a \in \mathbb{R}^d$, $B \in \mathbb{R}^{d\times d}$ symmetric and positive definite, $\lambda_{\min} > 0$, then for every γ such that $0 < \gamma \leq \sqrt{\lambda_{\min}/3}$ there exists a random variable Y such that*

1. $\mathbb{E}[Y] = a$ *and* $\mathbb{E}[Y^{\otimes 2}] = a^{\otimes 2} + B$

2. $\operatorname{supp}(Y) \subset \{y \in \gamma\mathbb{Z}^2 : |y|_\infty \leq |a|_\infty + \sqrt{2\lambda_1} + \sqrt{2\lambda_2} + 6\gamma\}$

Moreover, if $a = 0$*, the above applies for every* $0 < \gamma \leq \sqrt{\lambda_{\min}}$.

Proof. We first consider the case when $a = 0$. In this case, note that if $\bar{Y}$ is a discrete random variable that matches the second moment, $\mathbb{E}\bar{Y}^{\otimes 2} = B$ and Z is a Bernoulli random variable independent of $\bar{Y}$, $\mathbb{P}(Z = \pm 1) = \frac{1}{2}$, then the random variable $Y \coloneqq Z \cdot \bar{Y}$ has mean $\mathbb{E}[Y] = a = 0$ and $\mathbb{E}[Y^{\otimes 2}] = B$. Hence, solving the system (4.4) reduces to find $l_1, \ldots, l_r \in \gamma\mathbb{Z}^2$ such that

$$\sum_{i=1}^{r} p_i l_i l_i^T = B, \quad p_i \geq 0, \quad \sum_{i=1}^{r} p_i = 1, \tag{4.12}$$

for some $r \in \mathbb{N}_+$. Moreover, since B is symmetric, there exists $Q \in \mathbb{R}^{d\times d}$ orthonormal whose columns are the eigenvectors of B, such that $B = Q\Lambda Q^T$ and Λ is diagonal with elements the eigenvalues of B, which are strictly positive by assumption. Applying this, allows to rewrite (4.12) as

$$\sum_{i=1}^{r} p_i Q^T l_i l_i^T Q = \Lambda, \quad p_i \geq 0, \quad \sum_{i=1}^{r} p_i = 1. \tag{4.13}$$

Henceforth, we denote with $q_1 = (q_{11}, q_{12})$ and $q_2 = (q_{21}, q_{22})$ the two eigenvectors of B. It follows that if q_1 is eigenvector for Q then $-q_1$ is also an eigenvector for the same eigenvalue and $-q_1$ still forms an orthonormal basis with $\pm q_2$. We use these properties of eigenvalues in the following without loss of generality.

Case $q_{12} = 0$ or $q_{21} = 0$. Assume $q_{12} = 0$ (the case $q_{21} = 0$ follows analogously), then $q_1 = e_1$ and $q_2 = e_2$. Hence, $Q = I$ and we can refer to Theorem 4.21, which applies to d greater than 2 and B semi-positive definite.

Case $q_{11}, q_{21} > 0$**.** We first note that

$$\begin{cases} \langle q_1,q_1\rangle^2 = q_{11}^2 + q_{12}^2 = 1 \\ \langle q_2,q_2\rangle^2 = q_{21}^2 + q_{22}^2 = 1 \\ \langle q_1,q_2\rangle = q_{11}q_{21} + q_{12}q_{22} = 0 \end{cases} \quad \text{if and only if} \quad \begin{cases} q_{11} = \text{sign}(q_{11})\sqrt{1-q_{12}^2} \\ q_{21} = \text{sign}(q_{21})\sqrt{1-q_{22}^2} \\ \frac{q_{11}}{q_{12}} = -\frac{q_{22}}{q_{21}} \end{cases}$$

Using the last equation on the system on the right hand side, we obtain

$$\text{sign}(q_{11})\text{sign}(q_{12})\frac{\sqrt{1-q_{12}^2}}{|q_{12}|} = -\text{sign}(q_{21})\text{sign}(q_{22})\frac{|q_{22}|}{\sqrt{1-q_{22}^2}}$$

This in turn implies that

$$\text{sign}(q_{11})\text{sign}(q_{12}) = -\text{sign}(q_{21})\text{sign}(q_{22}).$$

We now claim that without loss of generality, we can assume that $q_{11} > 0$, $q_{12} > 0$ and $q_{21} > 0, q_{22} < 0$: indeed if we flip the sign of the eigenvectors they are still eigenvectors and we can re-label q_1, q_2. Moreover, note that

$$\begin{aligned} \frac{q_{11}}{q_{12}} = -\frac{q_{22}}{q_{21}} &\iff q_{22}^2 = \left(\frac{q_{11}}{q_{12}}\right)^2\left(1-q_{22}^2\right) \iff q_{22}^2\left(1+\left(\frac{q_{11}}{q_{12}}\right)^2\right) = \left(\frac{q_{11}}{q_{12}}\right)^2 \\ &\iff q_{22} = -\frac{\frac{q_{11}}{\sqrt{1-q_{11}^2}}}{\sqrt{1+\frac{q_{11}^2}{1-q_{11}^2}}} = -q_{11} \end{aligned}$$

which implies that

$$q_1 = \left(q_{11}, \sqrt{1-q_{11}^2}\right) \text{ and } q_2 = \left(\sqrt{1-q_{11}^2}, -q_{11}\right). \tag{4.14}$$

In the following, we assume that $1 > q_{11} \geq 1/\sqrt{2}$ - otherwise we can state q_1, q_2 as functions

of q_{12} and proceed similarly.[5] Now note that if there exist r points $l_1, \dots, l_r \in \gamma\mathbb{Z}^2$ s.t.

$$\Lambda \in \text{ConvexHull}\{(Q^T l_1)^{\otimes 2}, \dots, (Q^T l_r)^{\otimes 2}\}, \tag{4.15}$$

then Equation (4.13) hold true for some p_i to be computed. It only remains to show that (4.15) holds. This is a hard problem, but in our case a simple argument works. We make use of the simple observation that if $A = \{a_1, \dots, a_r\} \subset \mathbb{R}^h$ and $b \in \mathbb{R}^h$ s.t.

$$\forall \sim \in \{\leq, \geq\}^h \quad \exists a \in A \text{ such that } a \sim b \text{ then } b \in \text{ConvexHull}(A). \tag{4.16}$$

We now apply (4.16) with $b = \Lambda$ and $A = \{(Q^T l_1)^{\otimes 2}, \dots, (Q^T l_r)^{\otimes 2}\}$. Since Λ and $(Q^T l_i)^{\otimes 2}$ are symmetric we work in dimension $h = 3$ instead of $h = 4$. The premise in Equation (4.16) reduces to show that for each of the eight possible choices of $\sim = (\sim_2, \sim_1, \sim_1, \sim_3) \in \{\geq, \leq\}^3$, there exists a point $l \in \gamma\mathbb{Z}^2$ such that

$$\begin{pmatrix} \langle q_1, l\rangle^2 & \langle q_1, l\rangle\langle q_2, l\rangle \\ \langle q_1, l\rangle\langle q_2, l\rangle & \langle q_2, l\rangle^2 \end{pmatrix} \sim \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

Spelled out in coordinates the above reads as

$$\begin{cases} \langle q_1, l\rangle\langle q_2, l\rangle \sim_1 0 \\ \langle q_1, l\rangle^2 \sim_2 \lambda_1 \\ \langle q_2, l\rangle^2 \sim_3 \lambda_2 \end{cases} \xLeftrightarrow[\text{Using Equation (4.14)}]{} \begin{cases} \left(q_{11} l_{i1} + \sqrt{1-q_{11}^2}\, l_{i2}\right)\left(\sqrt{1-q_{11}^2}\, l_{i1} - q_{11} l_{i2}\right) \sim_1 0 \\ \left(q_{11} l_{i1} + \sqrt{1-q_{11}^2}\, l_{i2}\right)^2 \sim_2 \lambda_1 \\ \left(\sqrt{1-q_{11}^2}\, l_{i1} - q_{11} l_{i2}\right)^2 \sim_3 \lambda_2 \end{cases}$$

We denote $\varphi \coloneqq \frac{\sqrt{1-q_{11}^2}}{q_{11}}$, therefore the system we study is

$$\begin{cases} (l_{i1} + \varphi l_{i2})\,(\varphi l_{i1} - l_{i2}) \sim_1 0 \\ (l_{i1} + \varphi l_{i2})^2 \sim_2 \frac{\lambda_1}{q_{11}^2} \\ (\varphi l_{i1} - l_{i2})^2 \sim_3 \frac{\lambda_2}{q_{11}^2} \end{cases} \tag{4.17}$$

[5] The implications regarding q_1, q_2 at first glance can seem hostile, however geometrically they are more intuitive. They depend on the fact that q_1, q_2 form an orthonormal basis and changing the signs they are still eigenvectors and they still form an orthonormal basis.

where $l_i = (l_{i1}, l_{i2})$, $l_{i1} \in \gamma\mathbb{Z}$ and $l_{i2} \in \gamma\mathbb{Z}$. We now claim that for every of the $r = 8$ possible choices of $\sim_1$, $\sim_2$, $\sim_3 \in \{\leq, \geq\}^3$ one can find a $l_i \in \gamma\mathbb{Z}^2$ such that Equation (4.17) holds. All the possible combinations of $\sim_1$, $\sim_2$, $\sim_3$ are $\{\geq\geq\geq, \geq\geq\leq, \geq\leq\geq, \geq\leq\leq, \leq\geq\geq, \leq\geq\leq, \leq\leq\geq, \leq\leq\leq\}$ and we study them in order. We suppress the subscript i for $l_i = (l_{i1}, l_{i2})$. Let us also resume some bounds

$$\frac{1}{\sqrt{2}} \leq q_{11} < 1, \quad 1 \leq \frac{1}{q_{11}} < \sqrt{2}, \quad 0 \leq \varphi \leq 1, \quad 1 \leq \frac{1}{\varphi} < \infty. \tag{4.18}$$

- ($\geq\geq\geq$) We have the following system, whose solutions are on the right

$$\begin{cases} (l_1 + \varphi l_2)(\varphi l_1 - l_2) \geq 0 \\ (l_1 + \varphi l_2)^2 \geq \frac{\lambda_1}{q_{11}^2} \\ (\varphi l_1 - l_2)^2 \geq \frac{\lambda_2}{q_{11}^2} \end{cases} \iff \begin{cases} \text{Solution 1 } \geq\geq & & \text{Solution 2 } \leq\leq \\ l_1 \geq \frac{\sqrt{\lambda_1}}{q_{11}} - \varphi l_2 & \text{or} & l_1 \leq -\frac{\sqrt{\lambda_1}}{q_{11}} - \varphi l_2 \\ l_1 \geq \frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 & & l_2 \leq -\frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 \end{cases}$$

  "Solution 1 $\geq\geq$" means that we consider the case where $l_1 + \varphi l_2 \geq 0$, $\varphi l_1 - l_2 \geq 0$ and similarly for the other case "Solution 2 $\leq\leq$". In this and the successive cases we focus on "Solution 1 $\geq\geq$". In this case the solution can be obtained choosing e.g. $l_2 = 0$ and l_1 big enough, which however would result in an unbounded solution given that $1/\varphi \geq 1$ is not bounded from above. A bounded solution is[6]

$$\begin{aligned} l_2 &= \left\lfloor -\frac{\sqrt{\lambda_2}}{q_{11}} \right\rfloor_\gamma \in [-\sqrt{2\lambda_2} - \gamma, -\sqrt{\lambda_2}] \\ l_1 &= \left\lceil \frac{\sqrt{\lambda_1}}{q_{11}} - \varphi \left\lfloor -\frac{\sqrt{\lambda_2}}{q_{11}} \right\rfloor_\gamma \right\rceil_\gamma \in \left[\sqrt{\lambda_1}, \sqrt{2\lambda_1} + \sqrt{2\lambda_2} + 2\gamma\right] \end{aligned}$$

  which is bounded. This is obtained choosing l_2 so that $\frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 \approx 0$ and then using it in the first equation of "Solution 1".

[6] In this and the successive cases, we have found really useful plotting the lines (l_1 as function of l_2) represented by the two equations in "Solution 1".

- ($\geq\geq\leq$) We have the following system whose, solutions are on the right

$$\begin{cases} (l_1+\varphi l_2)(\varphi l_1 - l_2) \geq 0 \\ (l_1+\varphi l_2)^2 \geq \frac{\lambda_1}{q_{11}^2} \\ (\varphi l_1 - l_2)^2 \leq \frac{\lambda_2}{q_{11}^2} \end{cases} \iff \begin{cases} \text{Solution 1 } \geq\geq & & \text{Solution 2 } \leq\leq \\ l_1 \geq \frac{\sqrt{\lambda_1}}{q_{11}} - \varphi l_2 & \text{or} & l_1 \leq -\frac{\sqrt{\lambda_1}}{q_{11}} - \varphi l_2 \\ \frac{1}{\varphi} l_2 \leq l_1 \leq \frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 & & -\frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 \leq l_1 \leq \frac{1}{\varphi} l_2 \end{cases}$$

This case is more tedious. We choose $l_1 = \gamma n$, $l_2 = \gamma \lfloor n\varphi \rfloor$, where n is the smallest integer satisfying $\gamma n \geq \sqrt{2\lambda_1}$. This implies that the first equation of "Solution 1" is satisfied: using Equation (4.18)

$$\gamma n \geq \sqrt{2\lambda_1} \geq \frac{\sqrt{\lambda_1}}{q_{11}} \geq \frac{\sqrt{\lambda_1}}{q_{11}} - \varphi\gamma \lfloor n\varphi \rfloor .$$

We choose therefore $l_1 = \gamma n = \lceil \sqrt{2\lambda_1} \rceil_\gamma \in [0, \sqrt{2\lambda_1} + \gamma]$. The same bound for l_2 holds true since $l_2 = \gamma \lfloor n\varphi \rfloor \leq l_1$, again using Equation (4.18).

It remains to prove that the second equation of "Solution 1" is satisfied. Dividing the second equation of "Solution 1" by l_2 we prove the left inequality

$$\frac{l_1}{l_2} = \frac{\gamma n}{\gamma \lfloor n\varphi \rfloor} \geq \frac{n}{n\varphi} = \frac{1}{\varphi},$$

and, since by assumption $\gamma \leq \sqrt{\lambda_2}$ we conclude that

$$\frac{l_1}{l_2} - \frac{1}{\varphi} = \frac{\gamma n\varphi - \gamma \lfloor n\varphi \rfloor}{\gamma \lfloor n\varphi \rfloor \varphi} \leq \frac{\gamma}{\gamma \lfloor n\varphi \rfloor \varphi} = \frac{\gamma}{l_2 \varphi} \implies \frac{l_1}{l_2} \leq \frac{1}{\varphi}\left(1 + \frac{\gamma}{l_2}\right) \leq \frac{1}{\varphi}\left(1 + \frac{1}{l_2}\frac{\sqrt{\lambda_2}}{q_{11}}\right)$$

because $1 \leq \frac{1}{q_{11}} \leq \sqrt{2}$, which is equivalent to the second equation of "Solution 1".

- ($\geq\leq\geq$) We have the following system, whose solutions are on the right

$$\begin{cases} (l_1+\varphi l_2)(\varphi l_1 - l_2) \geq 0 \\ (l_1+\varphi l_2)^2 \leq \frac{\lambda_1}{q_{11}^2} \\ (\varphi l_1 - l_2)^2 \geq \frac{\lambda_2}{q_{11}^2} \end{cases} \iff \begin{cases} \text{Solution 1 } \geq\leq & & \text{Solution 2 } \leq\geq \\ -\varphi l_2 \leq l_1 \leq \frac{\sqrt{\lambda_1}}{q_{11}} - \varphi l_2 & \text{or} & -\frac{\sqrt{\lambda_1}}{q_{11}} - \varphi l_2 \leq l_1 \leq -\varphi l_2 \\ l_1 \geq \frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 & & l_1 \leq -\frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 \end{cases}$$

In this case we can take

$$l_2 = \left\lfloor -\frac{\sqrt{\lambda_2}}{q_{11}} \right\rfloor_\gamma \in [-\sqrt{2\lambda_2} - \gamma, -\sqrt{\lambda_2}], \quad l_1 = \left\lceil -\varphi \left\lfloor -\frac{\sqrt{\lambda_2}}{q_{11}} \right\rfloor_\gamma \right\rceil_\gamma \in \left[0, \sqrt{2\lambda_2} + 2\gamma\right].$$

This solution has been found choosing l_2 so that $\frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 \approx 0$ and then using it in the first equation of "Solution 1". The second equation of "Solution 1" is satisfied since l_1 is positive. For the first one, the left inequality is satisfied by definition of l_1, whilst the right inequality because $\gamma \leq \sqrt{\lambda_1} \leq \sqrt{\lambda_1}/q_{11}$.

- $(\geq\leq\leq)$ We have the following system, whose solutions are on the right

$$\begin{cases} (l_1 + \varphi l_2)(\varphi l_1 - l_2) \geq 0 \\ (l_1 + \varphi l_2)^2 \leq \frac{\lambda_1}{q_{11}^2} \\ (\varphi l_1 - l_2)^2 \leq \frac{\lambda_2}{q_{11}^2} \end{cases} \iff \begin{cases} \text{Solution 1} \geq\geq & & \text{Solution 2} \leq\leq \\ -\varphi l_2 \leq l_1 \leq \frac{\sqrt{\lambda_1}}{q_{11}} - \varphi l_2 & \text{or} & -\frac{\sqrt{\lambda_1}}{q_{11}} - \varphi l_2 \leq l_1 \leq -\varphi l_2 \\ \frac{1}{\varphi} l_2 \leq l_1 \leq \frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 & & -\frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 \leq l_1 \leq \frac{1}{\varphi} l_2 \end{cases}$$

In this case two possible solutions for the case "Solution 1" are $(l_1, l_2) = (0, 0)$ and $(l_1, l_2) = (\gamma, 0)$, since by assumption $\gamma \leq \min\{\sqrt{\lambda_1}, \sqrt{\lambda_2}\}$.

- $(\leq\geq\geq)$ We have the following system, whose solutions are on the right

$$\begin{cases} (l_1 + \varphi l_2)(\varphi l_1 - l_2) \leq 0 \\ (l_1 + \varphi l_2)^2 \geq \frac{\lambda_1}{q_{11}^2} \\ (\varphi l_1 - l_2)^2 \geq \frac{\lambda_2}{q_{11}^2} \end{cases} \iff \begin{cases} \text{Solution 1} \geq\leq & & \text{Solution 2} \leq\geq \\ l_1 \geq \frac{\sqrt{\lambda_1}}{q_{11}} - \varphi l_2 & \text{or} & l_1 \leq -\frac{\sqrt{\lambda_1}}{q_{11}} - \varphi l_2 \\ l_1 \leq -\frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 & & l_1 \geq \frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 \end{cases}$$

We could take simply $l_1 = 0$ and l_2 big enough, but this solution would be unbounded (depending on $1/\varphi$). Therefore, we reason similarly to the case $(\geq\geq\geq)$: let us rewrite the equations for "Solution 1" respect to l_2

$$l_2 \geq \frac{1}{\varphi}\frac{\sqrt{\lambda_1}}{q_{11}} - \frac{1}{\varphi} l_1, \qquad l_2 \geq \frac{\sqrt{\lambda_2}}{q_{11}} + \varphi l_1$$

and we can take as a solution

$$l_1 = \left\lfloor \frac{\sqrt{\lambda_1}}{q_{11}} \right\rfloor_\gamma \in \left[\sqrt{\lambda_1} - \gamma, \sqrt{2\lambda_1}\right]$$

$$l_2 = \left\lceil \frac{\sqrt{\lambda_2}}{q_{11}} + \varphi \left\lfloor \frac{\sqrt{\lambda_1}}{q_{11}} \right\rfloor_\gamma \right\rceil_\gamma \in \left[\sqrt{\lambda_2}, \sqrt{2\lambda_2} + \sqrt{2\lambda_1} + 2\gamma\right].$$

This solution has been found choosing l_1 so that $\frac{1}{\varphi}\frac{\sqrt{\lambda_1}}{q_{11}} - \frac{1}{\varphi} l_1 \approx 0$ and then using it in the other equation.

- ($\leq\geq\leq$) We have the following system, whose solutions are on the right

$$\begin{cases} (l_1 + \varphi l_2)(\varphi l_1 - l_2) \leq 0 \\ (l_1 + \varphi l_2)^2 \geq \frac{\lambda_1}{q_{11}^2} \\ (\varphi l_1 - l_2)^2 \leq \frac{\lambda_2}{q_{11}^2} \end{cases} \iff \begin{cases} \text{Solution 1 } \geq\leq & & \text{Solution 2 } \leq\geq \\ l_1 \geq \frac{\sqrt{\lambda_1}}{q_{11}} - \varphi l_2 & \text{or} & l_1 \leq -\frac{\sqrt{\lambda_1}}{q_{11}} - \varphi l_2 \\ -\frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 \leq l_1 \leq \frac{1}{\varphi} l_2 & & \frac{1}{\varphi} l_2 \leq l_1 \leq \frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 \end{cases}$$

We choose $l_1 = \gamma n$, $l_2 = \gamma\lceil n\varphi \rceil$, where n is the smallest integer satisfying $\gamma n \geq \sqrt{2\lambda_1}$. This implies that the first equation of "Solution 1" is satisfied: using Equation (4.18)

$$\gamma n \geq \sqrt{2\lambda_1} \geq \frac{\sqrt{\lambda_1}}{q_{11}} \geq \frac{\sqrt{\lambda_1}}{q_{11}} - \varphi\gamma \left\lceil n\varphi \right\rceil.$$

We choose therefore $l_1 = \gamma n = \lceil \sqrt{2\lambda_1} \rceil_\gamma \in [0, \sqrt{2\lambda_1} + \gamma]$. The same bound for l_2 holds true since $l_2 = \gamma\lceil n\varphi \rceil \leq l_1$, again using Equation (4.18).

It remains to prove that the second equation of "Solution 1" is satisfied. Dividing the second equation of "Solution 1" by l_2 we prove the left inequality

$$\frac{l_1}{l_2} = \frac{\gamma n}{\gamma\lceil n\varphi \rceil} \leq \frac{n}{n\varphi} = \frac{1}{\varphi},$$

and, since by assumption $\gamma \leq \sqrt{\lambda_2}$ we conclude that

$$\frac{l_1}{l_2} - \frac{1}{\varphi} = \frac{\gamma n\varphi - \gamma\lceil n\varphi \rceil}{\gamma\lceil n\varphi \rceil \varphi} \geq \frac{-\gamma}{\gamma\lfloor n\varphi \rfloor \varphi} = \frac{-\gamma}{l_2\varphi} \implies \frac{l_1}{l_2} \geq \frac{1}{\varphi}\left(1 - \frac{\gamma}{l_2}\right) \geq \frac{1}{\varphi}\left(1 - \frac{1}{l_2}\frac{\sqrt{\lambda_2}}{q_{11}}\right)$$

because $-1 \geq -\frac{1}{q_{11}} \geq -\sqrt{2}$, which is equivalent to the second equation of Solution 1.

- ($\leq\leq\geq$) We have the following system, whose solutions are on the right

$$\begin{cases} (l_1+\varphi l_2)(\varphi l_1 - l_2) \leq 0 \\ (l_1+\varphi l_2)^2 \leq \frac{\lambda_1}{q_{11}^2} \\ (\varphi l_1 - l_2)^2 \geq \frac{\lambda_2}{q_{11}^2} \end{cases} \iff \begin{cases} \text{Solution 1 } \geq\leq & & \text{Solution 2 } \leq\geq \\ -\varphi l_2 \leq l_1 \leq \frac{\sqrt{\lambda_1}}{q_{11}} - \varphi l_2 & \text{or} & -\frac{\sqrt{\lambda_1}}{q_{11}} - \varphi l_2 \leq l_1 \leq -\varphi l_2 \\ l_1 \leq -\frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 & & l_1 \geq \frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 \end{cases}$$

  We can proceed as in the case $\geq\leq\geq$: we can take

$$l_2 = \left\lceil \frac{\sqrt{\lambda_2}}{q_{11}} \right\rceil_\gamma \in \left[\sqrt{\lambda_2}, \sqrt{2\lambda_2} + \gamma\right], \qquad l_1 = \left\lceil -\varphi \left\lceil \frac{\sqrt{\lambda_2}}{q_{11}} \right\rceil_\gamma \right\rceil_\gamma \in \left[-\sqrt{2\lambda_2} - \gamma, 0\right]$$

  This solution has been found choosing l_2 so that $\frac{1}{\varphi}\frac{\sqrt{\lambda_2}}{q_{11}} + \frac{1}{\varphi} l_2 \approx 0$ and then using it in the first equation of "Solution 1". The second equation of "Solution 1" is satisfied since l_1 is negative. For the first one, the left inequality is satisfied by definition of l_1, whilst the right inequality because $\gamma \leq \sqrt{\lambda_1} \leq \sqrt{\lambda_1}/q_{11}$.

- ($\leq\leq\leq$) In this case it is easy to see that we can take the origin.

In all the cases we have studied the "Solution 1", however, for the "Solution 2" we can take $-(l_1, l_2)$. This concludes the proof for the case $a = 0$.

For a general $a \in \mathbb{R}^d$ we can build for any $\gamma > 0$ a random variable $\tilde{Y}$ with support on $\gamma\mathbb{Z}^d$ such that $\mathbb{E}\tilde{Y} = a$. This follows from Carathéodory's Theorem 2.2, considering the smallest hypercube with vertices in $\gamma\mathbb{Z}^d$ that contains a, that is of the 2^d possible choices of $\sim \in \{\leq, \geq\}^d$ determines a point $y_\sim$ on the hypercube around a s.t.

$$y_\sim \in \gamma\mathbb{Z}^d \text{ and } y_\sim \sim a$$

We denote the resulting 2^d points as $y_1, \ldots, y_{2^d}$ and now estimate the variance $\mathbb{V}[\tilde{Y}]$,

$$\left|\mathbb{V}[\tilde{Y}]\right| = \left|\mathbb{E}\left[\left(\tilde{Y} - a\right)^{\otimes 2}\right]\right| = \left|\sum_i \tilde{p}_i \left(y_i - a\right)^{\otimes 2}\right| \leq \sum_i \tilde{p}_i |y_i - a|^2 \leq d\gamma^2.$$

We now build a random variable $Y_\star$ supported on $\gamma\mathbb{Z}^2$ ($d = 2$) that is independent of Y and has mean and variance $\mathbb{E}Y_\star = 0$ and $\mathbb{V}Y_\star = \mathbb{E}Y_\star^{\otimes 2} = B - \mathbb{V}\tilde{Y}$. It follows that $Y \coloneqq \tilde{Y} + Y_\star$

has mean and variance $\mathbb{E}Y = a$ and $\mathbb{V}Y = B$. To build $Y_\star$ we use the first part of this proof. Therefore we need to verify that the eigenvalues of $B - \mathbb{V}\tilde{Y}$ are strictly positive. Since $\mathbb{V}\tilde{Y}$ and B are symmetric it follows from the Hoffman–Wielandt Theorem, Theorem 4.16, that there exists a permutation π such that for any k

$$-|\mathbb{V}\tilde{Y}| \leq \lambda_k \left[B - \mathbb{V}\tilde{Y}\right] - \lambda_{\pi(k)}\left[B\right] \leq |\mathbb{V}\tilde{Y}|$$

where $\lambda_k[B - \mathbb{V}\tilde{Y}]$ indicates the k-th eigenvalue of $B - \mathbb{V}\tilde{Y}$. Therefore,

$$\lambda_{\min}[B - \mathbb{V}\tilde{Y}] \geq \lambda_{\pi(\min)}\left[B\right] - |\mathbb{V}\tilde{Y}| \geq \lambda_{\min} - 2\gamma^2.$$

Hence, choosing $\gamma \leq \sqrt{\lambda_{\min}/3}$ guarantees $\gamma \leq \sqrt{\lambda_{\min}[B - \mathbb{V}\tilde{Y}]}$ since

$$\lambda_{\min} - 2\gamma^2 \geq \gamma^2 \Longleftrightarrow \gamma \leq \sqrt{\lambda_{\min}/3}.$$

Note that the cardinality of the support of $Y = \tilde{Y} + Y_\star$, eventually, can be reduced using the Caratheodory's Theorem 2.2.

We have proved that, by construction, the support of the r.v. $Y_\star$ is included in

$$\mathrm{supp}(Y_\star) \subset \{y \in \gamma\mathbb{Z}^2 : |y|_\infty \leq \sqrt{2\lambda_1[B - \mathbb{V}\tilde{Y}]} + \sqrt{2\lambda_2[B - \mathbb{V}\tilde{Y}]} + 2\gamma\}.$$

Thanks again to the Hoffman–Wielandt Theorem, Theorem 4.16 we have that

$$\begin{aligned}
\sqrt{2\lambda_1[B - \mathbb{V}\tilde{Y}]} + \sqrt{2\lambda_2[B - \mathbb{V}\tilde{Y}]} \leq & \sqrt{2\left(|\mathbb{V}\tilde{Y}| + \lambda_{\pi(1)}\left[B\right]\right)} + \sqrt{2\left(|\mathbb{V}\tilde{Y}| + \lambda_{\pi(2)}\left[B\right]\right)} \\
\leq & \sqrt{2\lambda_{\pi(1)}[B]} + \sqrt{2\lambda_{\pi(2)}[B]} + 2\sqrt{2|\mathbb{V}\tilde{Y}|} \\
\leq & \sqrt{2\lambda_1[B]} + \sqrt{2\lambda_2[B]} + 2\gamma\sqrt{2d}.
\end{aligned}$$

This implies that the support of $Y = \tilde{Y} + Y_\star$ is included in

$$\mathrm{supp}(Y) \subset \{y \in \gamma\mathbb{Z}^2 : |y|_\infty \leq |a|_\infty + \sqrt{2\lambda_1[B]} + \sqrt{2\lambda_2[B]} + 6\gamma\}.$$

□

Remark 4.19. *We conjecture that Theorem 4.18 generalizes to dimensions higher than $d = 2$ but the above proof strategy "fails" - it is infeasible to solve all the inequalities one by one for the equivalent of System* (4.17) *in higher dimension. Concretely, we believe that for $a \in \mathbb{R}^d$, $B \in \mathbb{R}^{d\times d}$ symmetric and positive definite, $\lambda_{\min} > 0$, then for every γ such that $0 < \gamma \leq \sqrt{\lambda_{\min}/(d+1)}$ there exists a random variable Y such that*

1. $\mathbb{E}[Y] = a$ *and* $\mathbb{E}[Y^{\otimes 2}] = a^{\otimes 2} + B$
2. $\text{supp}(Y) \subset \{y \in \gamma\mathbb{Z}^2 : |y|_\infty \leq |a|_\infty + \sum_{i=1}^d \sqrt{2\lambda_i} + c_d\gamma\}$

Note that it only remains to show the existence of a random variable $Y_\star$ such that $\mathbb{E}[Y_\star] = 0$ and $\mathbb{E}[Y_\star^{\otimes 2}] = B$ for any positive definite matrix B, since this allows to simply proceed as in the second part of the proof of Theorem 4.18.

Constructing the Lattice-Tree Model

Now we are ready to build the Markov chain X^n which approximates X for the 2-dimensional case and such that the state space recombines if the SDE coefficients are bounded. We indicate $\Sigma = \sigma(x)\sigma(x)^T$ and $\lambda_i(x)$ its eigenvalues.

Theorem 4.20. *Let $\mu, \sigma \in C_b^4$, $d = 2$, $\epsilon := \inf_x \lambda_{\min}(x) > 0$ and $\gamma_n = n^{-1/2}\sqrt{\epsilon/3}$. There exists a lattice approximation $(X^n)_{n\geq 1}$ to X on $X_0^n \cup \gamma_n\mathbb{Z}^2$ respect to the class of functions $f \in C_P^3$ with rate $1/2$. Moreover,*

$$\begin{cases} \text{card}[\text{supp}(X_i^n)] \leq ci^2 & \text{, if } \mu,\sigma \text{ are bounded,} \\ \text{card}[\text{supp}(X_i^n)] \leq ci^2\exp(2cin^{-1/2}) & \text{, if } \mu,\sigma \text{ have bounded linear growth.} \end{cases}$$

Proof. We argue as in Theorem 4.13, but using Theorem 4.18 above. In contrast to the one-dimensional case, the support of the random variable Y_x that gives the increment $X_{i+1}^n - X_i^n$ when $X_i^n = x$ now depends also on the eigenvalues of Σ. However, the Gerschgorin's Circle Theorem [67], Theorem 4.17, bounds the eigenvalues of Σ. We conclude by noting that the number of points of the lattice $\gamma_n\mathbb{Z}^2$ in a square of side c are $(c/\gamma_n)^2$ which yields the square factor in the cardinality of the support.

If X_0^n is not in the lattice $\gamma_n\mathbb{Z}^2$ more care is needed: for X_1^n to be supported on $\gamma_n\mathbb{Z}^2$ it is necessary to build the random variable $Y_{X_0^n}$ with support on the lattice $\gamma_n\mathbb{Z}^2 - X_0^n$, indeed $X_1^n = X_0^n + Y_{X_0^n}$ would be in $\gamma_n\mathbb{Z}^2$. This is equivalent to build the random variable $\tilde{Y}_{X_0^n}$ on $\gamma_n\mathbb{Z}^2$ such that $\mathbb{E}\tilde{Y}_{X_0^n} = \mu(X_0^n) + X_0^n$ and $\mathbb{E}\tilde{Y}^2_{X_0^n} = \Sigma(X_0^n) + (\mu(X_0^n) + X_0^n)^{\otimes 2}$. We define $Y_{X_0^n} \coloneqq \tilde{Y}_{X_0^n} - X_0^n$ with support on $\gamma_n\mathbb{Z}^2 - X_0$ s.t. $\mathbb{E}Y_{X_0^n} = \mu(X_0^n)$ and $\mathbb{V}Y_{X_0^n} = \Sigma(X_0^n)$. To build $\tilde{Y}_{X_0^n}$ using Theorem 4.18, we need to bound the eigenvalues of $\Sigma(X_0^n)+(\mu(X_0^n)+X_0^n)^{\otimes 2}$ using that

$$\begin{aligned}
\lambda_{\min}[\Sigma(X_0^n) + (\mu(X_0^n) + X_0^n)^{\otimes 2}] &= \inf_{d\neq 0} \frac{d^T[\Sigma(X_0^n) + (\mu(X_0^n) + X_0^n)^{\otimes 2}]d}{d^t d} \\
&\geq \inf_{d\neq 0} \frac{d^T\Sigma(X_0^n)d}{d^T d} + \inf_{e\neq 0} \frac{e^T(\mu(X_0^n) + X_0^n)^{\otimes 2}e}{e^T e} \\
&\geq \lambda_{\min}[\Sigma(X_0^n)] + \lambda_{\min}[(\mu(X_0^n) + X_0^n)^{\otimes 2}] \\
&\geq \lambda_{\min}[\Sigma(X_0^n)],
\end{aligned}$$

noting that $\lambda_{\min}[(\mu(X_0^n) + X_0^n)^{\otimes 2}] \geq 0$, which concludes the proof. □

4.5 Lattice-Tree Models for Multi-Dimensional Diffusions

We now carry out the same procedure in the multi-dimensional case. In this case, our assumptions will be more stringent and we will consider the lattice to depend on the structure of the matrix B.

Matching the Moments

In this Subsection we solve System (4.4) for a general dimension d, but on pre-specified lattice. We call $Q \in O(d)$ the orthonormal matrix such that $B = Q\Lambda Q^T$, with Λ the diagonal matrix consisting of the eigenvalues of B.

Theorem 4.21. *Let $B \in \mathbb{R}^{d\times d}$ symmetric and positive semi-definite, $\gamma > 0$, then there exists a random variable Y such that*

1. $\mathbb{E}[Y] = 0$ *and* $\mathbb{E}[Y^{\otimes 2}] = B$.
2. $\text{supp}(Y) \subset \{y \in \gamma Q\mathbb{Z}^d : |y|_\infty \leq \sqrt{\sum_i \lambda_i} \cdot \max_i |Qe_i| + \gamma\}$

Proof. If all the eigenvalues λ_i of B are 0, then $Y = 0$ a.s. is a solution, so let us suppose that at least one eigenvalue is strictly positive.

Recall that, from the proof of Theorem 4.18, if $\bar{Y}$ is a discrete random variable that matches the second moment, $\mathbb{E}\bar{Y}^{\otimes 2} = B$ and Z is a Bernoulli random variable independent of $\bar{Y}$, $\mathbb{P}(Z = \pm 1) = \frac{1}{2}$, then the random variable $Y \coloneqq Z \cdot \bar{Y}$ has mean $\mathbb{E}[Y] = 0$ and $\mathbb{E}[Y^{\otimes 2}] = B$. Hence, solving the system (4.4) reduces to find $\bar{l}_i \in \gamma Q\mathbb{Z}^d$ s.t.

$$\sum_i \bar{p}_i \bar{l}_i \bar{l}_i^T = B, \qquad \bar{p}_i > 0, \sum_i \bar{p}_i = 1$$

then the measure on the points $\bar{l}_i$, $-\bar{l}_i$ with weights respectively $p_i = \bar{p}_i/2$ represents the r.v. Y we are looking for. Since we suppose B to be positive semi-definite and symmetric, there exists an orthonormal matrix Q such that $B = Q\Lambda Q^T$, whose columns are the eigenvector of B and Λ is a diagonal matrix whose value are the non-negative eigenvalues. If we indicate $\bar{l}_i = \gamma Q\bar{z}_i$, $\bar{z}_i \in \mathbb{Z}^d$, we have that

$$\sum_i^r \bar{p}_i \bar{l}_i \bar{l}_i^T = B = Q\Lambda Q^T \Leftrightarrow \sum_i^r \bar{p}_i \gamma^2 Q\bar{z}_i \bar{z}_i^T Q^T = Q\Lambda Q^T \Leftrightarrow \sum_i^r \bar{p}_i \gamma^2 \bar{z}_i \bar{z}_i^T = \Lambda.$$

Therefore, rearranging the last equation we obtain

$$\sum_i^r p_i \gamma^2 \bar{z}_i \bar{z}_i^T = \sum_{j=1}^d \frac{\lambda_j}{\sum_{k=1}^d \lambda_k} \left(\sqrt{\sum_{k=1}^d \lambda_k} \cdot e_j \right)^{\otimes 2}.$$

Since r is a free parameter for us, we can take $r = d^* + 1$, where d^* is the number of different eigenvalues not equal to 0, and we can solve the problem independently for any eigenvalue, then we merge together all the solutions. Thus, we obtain

$$\bar{l}_{d^*+1} = \mathbf{0}, \qquad \bar{p}_{d^*+1} = 1 - \left\{ \frac{\sum_{j=1}^d \lambda_j}{\left\lceil \sqrt{\sum_{j=1}^d \lambda_j} \right\rceil_\gamma^2} \right\}$$

$$\bar{l}_i = Q \left\lceil \sqrt{\sum_{j=1}^d \lambda_j} \right\rceil_\gamma e_i \qquad \bar{p}_i = \frac{m(\lambda_i)\lambda_i}{\sum_{j=1}^d \lambda_j} \left\{ \frac{\sum_{j=1}^d \lambda_j}{\left\lceil \sqrt{\sum_{j=1}^d \lambda_j} \right\rceil_\gamma^2} \right\}$$

where $m(\lambda_i)$ is the algebraic multiplicity of the eigenvalue λ_i. At this point we have a probability measure done by the d^*+1 weights $\bar{p}_i$ and the points $\bar{l}_1, ..., \bar{l}_{d^*}, \mathbf{0} \in \gamma Q\mathbb{Z}^d$ and we refer to this random variable with $\bar{Y}$. We can now build the r.v. Y taking the $2d^*+1$ points $\{\bar{l}_i, -\bar{l}_i\}$ with probability $p_i = \bar{p}_i/2$, such that $\mathbb{E}Y = 0$ and $\mathbb{E}Y^{\otimes 2} = B$. □

Constructing the Lattice-Tree Model

Although we have not proved Theorem 4.20 for general dimension d, we can still prove one particular case. We are ready to build the Markov chain X^n which approximates X in the multi-dimensional case under more stringent conditions than before.

Theorem 4.22. *Let $\mu(x) = 0, \sigma \in C_b^4$, $d \geq 2$, $\Sigma(x) = Q\Lambda(x)Q^T$, $\Lambda(x) \in \mathbb{R}^{d\times d}$ a diagonal matrix and Q an orthogonal matrix. For any $c > 0$, set $\gamma_n = cn^{-1/2}$ and let us suppose that $X_0^n \in \gamma_n Q\mathbb{Z}^d$. There exists a lattice approximation $(X^n)_{n\geq 1}$ to X on $\gamma_n Q\mathbb{Z}^d$ respect to the class of functions $f \in C_P^3$ with rate $1/2$. Moreover,*

$$\begin{cases} \text{card}[\text{supp}(X_i^n)] \leq i^d c. & \text{, if } \mu,\sigma \text{ are bounded,} \\ \text{card}[\text{supp}(X_i^n)] \leq ci^d \exp(dcin^{-1/2}) & \text{, if } \mu,\sigma \text{ have bounded linear growth.} \end{cases}$$

Proof. The proof follows the same reasoning used in Theorem 4.13 and 4.20, but using Theorem 4.21. The bounds on the cardinality of the support can be obtained as explained in the proof of Theorem 4.20 □

It is relevant to recall that to obtain $\mu(x) = 0$ the Girsanov Theorem can be helpful.

4.6 Variations and Relation to Previous Work

Let us see how to recover some particular cases treated in the literature and some (apparently) new cases.

4.6.1 Transformations

In Nelson and Ramaswamy [136], the authors are able to build a recombining binomial tree if σ is constant. Their method can be briefly resumed in the following way: first,

obtain a diffusion with a constant volatility term using the Lamperti transform; second, build the recombining binomial tree; third, apply the Lamperti inverse transform. However, using this procedure it is possible to have negative probabilities for the branches of the tree [136]. We show how to use the same logic in our setting; for simplicity, we set $d = 1$. In Theorem 4.13 we have seen that, if σ satisfies an ellipticity condition, then

$$\begin{cases} \text{card}\left[\,\text{support}(X_i^n)\,\right] \leq c\left(i \times e^{n^{-1/2}i}\right), & \text{if } \mu \text{ and } \sigma^2 \text{ have bounded linear growth,} \\ \text{card}\left[\,\text{support}(X_i^n)\,\right] \leq ci, & \text{if } \mu \text{ and } \sigma^2 \text{ are bounded.} \end{cases} \tag{4.19}$$

If we are interested in recombining trees[7] we can still do something in the spirit of [136] to obtain a recombining tree also when μ and σ are not bounded.

Let us suppose we are interested in approximating the solution Y_t of an SDE whose coefficients are linearly bounded: using the results of Theorem 4.13 the Markov chain built would lead to the cardinality explosion of the state space, Equation (4.19). The aim is to find a (smooth enough) function f such that $Y_t = f(X_t)$, where X_t is the solution of the SDE (4.1), with bounded μ, σ. Thus, using the Ito's rule, Y solves

$$dY_t = \left[\mu(X_t)\partial f(X_t) + \frac{\sigma(X_t)^2}{2}\partial^2 f(X_t)\right] dt + \sigma(X_t)\partial f(X_t) dW_t. \tag{4.20}$$

For example, if Y is a Geometric Brownian Motion, using $f(x) = e^x$, we obtain that X is a Brownian Motion with a bounded mean and variance, i.e.

$$\frac{dY_t}{Y_t} = \mu dt + \sigma dW_t, \qquad dX_t = \left[\mu - \frac{\sigma^2}{2}\right] dt + \sigma dW_t.$$

The benefit is that the SDE of X has bounded coefficients, thus we can build a Markov chain X^n with support on a lattice that recombines, i.e. the cardinality increases linearly. Using the results of the previous Sections, we have two advantages compared to [136]: first, we do not need the volatility of X to be constant, i.e. f^{-1} could be different from the Lamperti Transform; second, we do not have issues with possible negative probabilities.

[7] Note that the states space of a recombining tree is not a lattice a priori.

Going back to Equation (4.20) we can see that[8]

$$\mathbb{E}_t[Y_{t+n^{-1}} - Y_t] = \left[\mu(X_t)\partial f(X_t) + \frac{\sigma(X_t)^2}{2}\partial^2 f(X_t)\right] n^{-1} + O(n^{-2}),$$
$$\mathbb{E}_t[Y_{t+n^{-1}} - Y_t]^2 = \left[\sigma(X_t)\partial f(X_t)\right]^2 n^{-1} + O(n^{-2}).$$

Now, if we suppose to have built the Markov chain approximation X_i^n as in Theorem 4.13, then we want to see if $Y_i^n = f(X_i^n)$ satisfy Lemma 4.14-Item (1), respect to Y. Using a Taylor expansion we get

$$\begin{aligned}\mathbb{E}_i Y_{i+1}^n - Y_i^n =& \mathbb{E}_i\left[f(X_{i+1}^n) - f(X_i^n)\right] \\ =& \mathbb{E}_i\left[\partial f(X_i^n)\Delta X_i^n + \frac{1}{2}\partial^2 f(X_i^n)(\Delta X_i^n)^2 + O(|\Delta X_i^n|^3)\right] \\ =& \partial f(X_i^n)\mathbb{E}_i\left[\Delta X_i^n\right] + \frac{1}{2}\partial^2 f(X_i^n)\mathbb{E}_i\left[\Delta X_i^n\right]^2 + O(n^{-3/2}),\end{aligned}$$

where $\Delta X_i^n = X_{i+1}^n - X_i^n$ and by construction $\mathbb{E}_i O(|\Delta X_i^n|^3) \leq c\sum_j p_j \left|n^{-1/2} z_j\right|^3 \leq cn^{-3/2}$, $z_j \in \mathbb{Z}$. Given that X_i^n satisfies Lemma 4.14-Item (1) respect to X, we have that $\mathbb{E}_i[Y_{i+1}^n - Y_i^n] = \mathbb{E}_{t_i}[Y_{t+n^{-1}} - Y_t] + O(n^{-3/2})$. For the second moment we proceed similarly

$$\begin{aligned}\mathbb{E}_i\left[Y_{i+1}^n - Y_i^n\right]^2 =& \mathbb{E}_i\left[f(X_{i+1}^n) - f(X_i^n)\right]^2 \\ =& \mathbb{E}_t\left[\partial f(X_i^n)\Delta X_i^n + O(|\Delta X_i^n|^2)\right]^2 \\ =& \partial f(X_i^n)^2\mathbb{E}_t\left[\Delta X_i^n\right]^2 + 2\partial f(X_i^n)\mathbb{E}_t\left[\Delta X_i^n \times O(|\Delta X_i^n|^2)\right] + \mathbb{E}_t O(|\Delta X_i^n|^4),\end{aligned}$$

note that $|\mathbb{E}_t[\Delta X_i^n \times O(|\Delta X_i^n|^2)]| \leq cn^{-3/2}$. Since

$$\mathbb{E}_i\left[\Delta X_i^n\right]^2 = \mu(X_i^n)^2 n^{-2} + \sigma(X_i^n)^2 n^{-1} + O(n^{-3/2}) = \sigma(X_i^n)^2 n^{-1} + O(n^{-2}),$$

we can conclude that $\mathbb{E}_i[Y_{i+1}^n - Y_i^n]^2 = \mathbb{E}_{t_i}[Y_{t_i+n^{-1}} - Y_{t_i}]^2 + O(n^{-3/2})$. In the same way we obtain $\mathbb{E}_i|Y_{i+1}^n - Y_i^n|^3 = O(n^{-3/2})$, therefore we can conclude that Y^n satisfies Lemma 4.14-Item (1) respect to Y for $\alpha = 1/2$.

[8] We have to suppose $\mathbb{E}_t \int_t^{t+n^{-1}} \sigma(X_s)\partial f(X_s) dW_s = 0$. We recall that a sufficient condition for the integrand is to be (locally) square integrable.

If $Y^n_\star$ represents the lattice-tree model built using the techniques presented in Theorem 4.13, the main difference between $Y^n_\star$ and Y^n is that the cardinality of state space of Y^n increases linearly with the number of steps by construction, since the cardinality of state space of X^n increases linearly with the number of steps. However, the states space is not any more a lattice a priori, indeed it is a non linear transformation of a lattice.

4.6.2 Optimal Control and Optimal Stopping

A particular case where it is often necessary to consider discrete-space approximations of the SDE (4.1) is when control problems are studied. In these instances, it is easier to solve the problems if the time and the space are discrete, due to the Dynamic Programming principle. As a general reference, we cite Kushner [108], Kushner and Dupuis [109] for control theory and [109, Chapter 10, Theorem 6.2] for optimal stopping problems. In particular, the local consistency conditions of [108, 109][9] are equivalent with those considered in this Chapter, see Definition 4.4 and Lemma 4.24. Therefore, this Chapter can be applied also to the more general case of optimal control problems, e.g. in finance Portfolio Construction or American options.

We briefly introduce the problem of optimal stopping. Let us consider $\mathcal{T}$ and $\mathcal{T}^n$ the set of a.s. finite stopping times with respect to the filtration of the SDE (4.1) and the filtration of the approximation scheme X^n, moreover let us define for some functions g, k

$$V(x) := \min_{\tau \in \mathcal{T}} \mathbb{E}\left[\int_0^{\tau \wedge 1} k(X_s) ds + g(X_{\tau \wedge 1}) \middle| X_0 = x \right],$$
$$V^n(x) := \min_{\tau^n \in \mathcal{T}^n} \mathbb{E}\left[\frac{1}{n} \sum_{i=0}^{\tau^n \wedge n} k(X^n_i) + g(X_{\tau^n \wedge n}) \middle| X^n_0 = x \right].$$

Theorem 4.23. *([109, Chapter 10, Theorem 6.2]) Let X be the solution of the SDE (4.1) and X^n be one of the schemes built in Theorems 4.13, 4.20, 4.22. If μ, σ, g, k are bounded Lipschitz continuous[10], then $V^n(x) \to V(x)$, as $n \to \infty$.*

[9] See [109, Equation (1.3), page 71].

[10] Other rather general and technical assumptions on V are needed.

Proof. Thanks to [109, Chapter 10, Theorem 6.2], it is enough to check that the conditions in [109, Equation (1.3), page 71] are satisfied for the approximation schemes of Theorems 4.13, 4.20, 4.22. [109, Equation (1.3), page 71] has two different types of conditions: the first type of conditions is equivalent to Lemma 4.14-(1); the second type of conditions requires that a.s. $\sup_i |X^n_{i+1} - X^n_i| \to 0$, as $n \to \infty$, but this is a consequence of the fact that the r.v. $Y_{(\cdot)}$ built in the proof of Theorems 4.13, 4.20, 4.22 are bounded by the bounds of Theorems 4.8, 4.18, 4.21. □

To deal with control problems the reasoning is the same: we should introduce controls in the coefficients μ and σ and define the related value functions V, V^n, [108, 109]. Nevertheless, in this case, we should build the r.v. $Y_{(\cdot)}$ for any different control available, which can have a remarkable effect from a computational perspective.

4.6.3 Dealing with hypo-elliptic case.

If we do not know a priori if Σ is elliptic, we can take a $\gamma > 0$ reasonably small and then start building the random variables $Y_{(\cdot)}$ for the lattice $\gamma\mathbb{Z}^d$. If we arrive in a point x where $\sqrt{\lambda_{\min}[\Sigma(x)]} \leq \gamma$, then it is enough to add to the lattice $\gamma\mathbb{Z}^d$ some points in the lattice $\mu(x) + \gamma Q\mathbb{Z}^d$, where Q depends as usual from $\Sigma(x)$.

4.6.4 Switching Diffusions.

Switching diffusions can be treated in a straightforward manner. Suppose that the functions Σ and μ depend on one (or more) independent dynamic process $\theta(t)$, the "switch", in a Markovian way. For simplicity, let us suppose that $\theta(t) \in \{0,1\}$. We consider four functions Σ_0, μ_0 and Σ_1, μ_1:

$$\mathbb{E}_t[X_{t+n^{-1}} - X_t] \approx \begin{cases} \mu_1(X_t) & \text{if } \theta_t = 1 \\ \mu_0(X_t) & \text{if } \theta_t = 0 \end{cases}, \qquad \mathbb{V}_t[X_{t+n^{-1}} - X_t] \approx \begin{cases} \Sigma_1(X_t) & \text{if } \theta_t = 1 \\ \Sigma_0(X_t) & \text{if } \theta_t = 0 \end{cases},$$

Thus, we can approximate two different processes defined by Σ_0, μ_0 and Σ_1, μ_1 independently, assuming to know or to be able to build a "good" discrete time approximation θ^n of $\theta(t)$.

4.6.5 More general processes.

We focused on SDEs because they are the most frequent and most used models in the literature and because it is plenty of results showing that the local consistency - or similar - assumption, see Definition 4.4, is sufficient for the weak convergence of the scheme. For simplicity, we have presented the case of time-homogeneous SDEs, but since the convergence results in [98] apply also to time-inhomogeneous SDEs, it is straightforward to extend the results here presented. Nevertheless, from a computational perspective, time-inhomogeneous SDEs are more challenging.

Finally, we are not aware of similar sufficient conditions to Definition 4.4 for the weak convergence of more general processes, e.g. Markovian or non-Markovian ones. However, it should be clear that if they exist, the procedure here presented could be extended - probably having some dependence on the full history of the tree.

4.7 Algorithm and Experiments

First, let us state the pseudo-code of the algorithm for the construction of the Transition Matrix. We suppose for simplicity $X_0 = 0$, so that $X_0^n \in \gamma_n Q\mathbb{Z}^d$ for any n, Q.

Algorithm 7 Recombining Lattice-Tree Construction

Inputs: μ, σ, x_0, n

Choose γ_n and Q appropriately, see Theorem 4.13, 4.20, 4.22
queue_todo.append[x_0]
while $i \leq n$ **do**
    while queue_todo is not empty **do**
        $x \leftarrow$queue_todo[0]
        Solve System (4.4) for $a = n^{-1}\mu(x)$, $B = n^{-1}\Sigma(x)$
        Call $l_j \in \gamma_n Q\mathbb{Z}^d$ the points that solve System (4.4)
        Transition Matrix $\leftarrow$ $\{x$ can go to $\{x + l_j\}$ with probabilities $\{p_j\}$ $\}$
        queue_todo_nextstep.append[for any j, $x + l_j$ such that $x + l_j \notin$ queue_done $\cup$ queue_todo_nextstep]
        queue_todo.remove[x]
        queue_done.append[x]
    end while
    queue_todo$\leftarrow$queue_todo_nextstep
    queue_todo_nextstep.remove[all elements]
end while
Return Transition Matrix

Remember that, if X_0^n is not in the lattice, X_1^n must be built directly to be in the considered lattice, see the proof of Theorem 4.13.

4.7.1 Solving the Moment Matching System

Before presenting the experiment, we analyse how the system (4.4) can be solved in practice, step 6. This Section presents the relation of this Chapter with Chapter 2.
Let the dimension be 2, given that the experiment has been done in this case, but the reasoning follows similar logic in other situations. In Theorem 4.18, we have proved that it is possible to solve system (4.4) on $\gamma\mathbb{Z}^2$ if γ is small enough and we consider the points $l_i \in \gamma\mathbb{Z}^2$ such that $|l_i| \leq c$ for some c specified in Theorem 4.18, depending on the eigenvalues of the matrix B. In particular, Theorem 4.18 requires that the eigenvalues must always be far from 0. However, we have a strong belief that this condition is far to be necessary and the solution to System (4.4) probably is not unique, see also Chapter 2. We can think of three ways to solve (4.4):

(i) follow the proof of Theorem 4.18;

(ii) solve the minimization problem

$$\{w_j,l_j\} := \arg\min \left| \sum_j \begin{pmatrix} z_j \cdot w_j \\ z_j^{\otimes 2} \cdot w_j \end{pmatrix} - \begin{pmatrix} n^{-1}\mu(x) \\ n^{-2}\mu(x)^{\otimes 2} + n^{-1}\Sigma(x) \end{pmatrix} \right|^2 \tag{4.21}$$
$$\text{s.t.} \quad l_j \in \gamma\mathbb{Z}^2, \quad |l_j| \leq c$$
$$0 \leq w_j \leq 1, \quad \sum_j w_j = 1$$

(iii) use Algorithm 2 in Chapter 2;

Using (i) we have to check that the hypothesis of Theorem 4.18 are satisfied, i.e. γ is small enough. (ii) always finds a solution and we do not have to check if the hypothesis of Theorem 4.18 are satisfied. However, the solution found can be an approximation if the minimized function is different from 0, or if γ is not small enough. Both the solutions found by (i) and (ii) can be still reduced: let us call $l_j^\star$ the points found by (i) or (ii), for

which $w_j > 0$. Then, by the Caratheodory's Theorem 2.2 we know that the points $l_j^\star$ can be more than the ones required.

As we have seen in Chapter 2, different algorithms have been presented in the literature to reduce the support of the measure, see Litterer and Lyons [119], Maalouf et al. [126], Tchernychova [177] and the algorithms in Chapter 2. There are two main differences between them: Algorithm 2 in Chapter 2 is the only one that is randomized and it does not require an initial sequence of weights. From another point of view, Algorithm 2 can approximate any point in the convex hull of the given points, whilst [119, 126, 177] can approximate "only" the "barycenter" with respect to the given weights, which are therefore required as input for the algorithms. Nevertheless, we recall that the algorithms in Chapter 2 can also work if an initial sequence of weights is given. The advantages of using (*iii*) are that we do not have to check if the hypothesis of Theorem 4.18 are satisfied and that it finds directly a reduced solution, i.e. one that minimizes the number of points used - after the application of Caratheodory's Theorem 2.2. Therefore, it is important to note that Algorithm 2 in Chapter 2 is the only one that can be directly applied to solve (4.4); as we said, the other ones need an initial sequence of weights, therefore in our case they can be applied after the minimization problem (4.21) has been solved. The disadvantages are that, being randomized, Algorithm 2 can take too many iterations, although experimentally in low dimensions[11] it terminates faster than the other algorithms, if the solution exists, see Chapter 2 and the experiments therein. Note that, if γ is not small enough, the solution of System (4.4) could not exists and, in this case, Algorithm 2 would run forever.

Since the sufficient conditions of Theorem 4.18 are quite tight, our strategy for the following experiment has been the following:

1. try to solve (4.4) using Algorithm 2 in Chapter 2;

2. after a certain number of iterations, if Algorithm 2 has not found a solution, we solve the minimization problem (4.21) and then we apply Algorithm 3 in Chapter 2, given the sequence of weights solving (4.4). Remember that Algorithm 3 is a combination of the Algorithms in [177] and Chapter 2.

[11] Considering $\gamma\mathbb{Z}^2$, the dimension of the measure reduction problem is 5, because we have to approximate the 2 components of the first moment and 3 of the covariance matrix, since it is symmetric.

The results in Chapter 2 can also help to reduce further the cardinality of the state space. Theorem 2.4-Item 3 in Chapter 2 tells us that given a solution of (4.4) we could possibly modify it changing the points (and therefore the weights) if the new points satisfy a certain condition. The idea is thus to search if there are points $|\bar{l}_j| \leq c, \bar{l}_j \in \gamma\mathbb{Z}^2$ such that $\bar{l}_j$ satisfies the condition explicited in Theorem 2.4-Item 3 and $\bar{l}_j$ have been already used in the past - indeed let us recall that we build the tree recursively starting from the initial point. Briefly, we indicate $\{l_j^\star\}, \{w_j^\star\}$ a solution of (4.4), when the Caratheodory's reduction has already taken place, and $C(\{\cdot\})$ indicates the cone built from the points $\{\cdot\}$. Theorem 2.4-Item 3 claims that any point into the cone $C(\{-l_j^\star\} \setminus -l_1^\star)$ can be exchanged with $l_1^\star$ to have a new solution to (4.4). Namely, considering that we want to find a solution in $\gamma\mathbb{Z}^2$, we can look for any point $\bar{l}$ in

$$\{|l| < c,\, l \in \gamma\mathbb{Z}^2\} \cap \{l \in \gamma\mathbb{Z}^2 : x + l \text{ has been used to build previous r.v.}\} \cap C(\{-l_j^\star\} \setminus -l_1^\star).$$

The points $\bar{l} \cup \{l_j^\star\} \setminus l_1^\star$ constitute another solution to (4.4), changing accordingly the weights. This can be generalized for any $l_i^\star$, not only $l_1^\star$ and can be iteratively done for all of them. This procedure can help us to reduce further the total number of states necessary to approximate the considered SDE, being able to re-use points already used, and therefore increasing the "level of recombination" of the tree. As a drawback, this procedure leads to a higher computational cost.

Sparsity of the Transition Matrix. It is relevant to underline that, thanks (also) to the Caratheodory's Theorem, the final transition matrix results to be sparse. For simplicity let us assume that the coefficients μ, σ are bounded. In dimension d we have that from a state x we can go to maximum $1 + d + d(d+1)/2$ different points. Moreover, the transition matrix has dimension $O(n^d)$, which means that for any row only $O(d^2/n^d)\%$ of elements are different from 0, see Figure 4.5.

4.7.2 Example: Mean reverting Heston model

We consider the Heston model (HM) [80], a Stochastic Volatility model in finance

$$\begin{aligned} dP_t =& \mu P_t dt + \sqrt{V_t} P_t dW_t^P \\ dV_t =& \lambda(\theta - V_t)dt + \xi\sqrt{V_t} dW_t^V, \end{aligned} \tag{4.22}$$

where P_t is the price of an asset and V_t its instantaneous variance; W_t^P, W_t^V are two standard Brownian Motions with correlation parameter ρ and $\lambda, \theta, \mu, \xi$ are all constants. We recall that if $2\lambda\theta > \xi^2$, then V_t is a.s. positive, see for example [47].

In the literature, "specialized" tree approximations for the HM (4.22) have been developed, e.g. Akyıldırım et al. [6], Zeng and Zhu [193]. However, to the best of our knowledge, once we let V depends on P it is not possible to build a tree with the techniques developed in the past. Instead, we can use Theorem 4.20 without overlooking at the technicalities (see below). Several extensions exist to the HM, in particular it is possible to make V dependent from P or modifying the correlation ρ. See Romano and Touzi [157] for a theoretical study.

We want to highlight that the aim of this Section is not to create a new state-of-the-art model for prices in finance, but to have an interesting toy example to test the procedure described in this Chapter.

It is an empirical observation that the price and the volatility have an inverse relation, and that V has a mean reverting behaviour. Thus, we assume

$$\theta := \theta_t = \theta(P_t) = C\frac{1}{P_t + 1} + k.$$

We can check that in this case V has still a mean-reverting property[12], $\lim_{t\to\infty} \mathbb{E}V_t = \mathbb{E}\theta_\infty$.

Let us now restate the "modified" HM we are considering in vectorial form

$$d\begin{pmatrix} P_t \\ V_t \end{pmatrix} = \begin{pmatrix} \mu P_t \\ \lambda\left(\frac{C}{1+P_t} + K - V_t\right) \end{pmatrix} dt + \sqrt{V_t}\begin{pmatrix} P_t & 0 \\ 0 & \xi \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho^2} \end{pmatrix} d\begin{pmatrix} W_t^{(1)} \\ W_t^{(2)} \end{pmatrix},$$

[12] This can be obtained as usual studying the differential form of $d(e^{t\lambda}V_t)$ using the Ito's rule. We have to suppose the existence of $\mathbb{E}\theta_\infty$.

where $W_t^{(1)}, W_t^{(2)}$ are two independent Brownian Motions.

However, Theorem 4.20, we have to be careful regarding the eigenvalues of $\Sigma = \sigma\sigma^\top$, where

$$\sigma = \sqrt{V_t} \begin{pmatrix} P_t & 0 \\ 0 & \xi \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho^2} \end{pmatrix},$$

which cannot be bounded away from 0. Therefore, this model does not satisfy the hypothesis of Theorem 4.20. Moreover, note that the coefficients are not bounded, but they have linear bounded growth, thus we test also the bounds found in Theorem 4.20.

As common in the literature, we study $\log P_t$, i.e. we use the methods explained in Section 4.6.1. From an approximation point of view, $\log P_t$ has an important advantage: it can take both positive and negative values. This is closer to the hypothesis of Theorem 4.20, because in Theorem 4.20 no constraints are considered, see also Remark 4.11. Finally, we obtain the SDE

$$d \begin{pmatrix} \ln P_t \\ V_t \end{pmatrix} = \begin{pmatrix} \mu - \frac{1}{2} V_t \\ \lambda \left(\frac{C}{1+P_t} + K - V_t \right) \end{pmatrix} + \sqrt{V_t} \begin{pmatrix} 1 & 0 \\ \xi\rho & \xi\sqrt{1-\rho^2} \end{pmatrix} d \begin{pmatrix} W_t^{(1)} \\ W_t^{(2)} \end{pmatrix}.$$

Results. We consider the following constants $\log(P_0) = \log(100)$, $V_0 = 5$, $\mu = 0$, $\lambda = 2$, $\rho = 0.2$, $\xi = 1$, $C = 2$, $K = 5$ and $t \in [0,1]$. The results of the experiments are in Figures 4.2, 4.3, 4.4 and 4.5. The code can be found at github.com/FraCose/Tree-Recombination.

In Figure 4.2 it is shown how the state space is built recursively from the initial point x_0. Starting from the initial point x_0 the algorithm finds recursively the states where to go - from where we are - with the probabilities such that the local consistency condition is satisfied. The plots have been done during the steps 1, 2, 4, 5, 8, 10 respectively. It is possible to notice that some states have been approximated, i.e. the first two moments are not exactly matched. This is due to two main reasons: first, the eigenvalues are not bounded away from 0 and therefore the γ of Theorem 4.20 cannot be computed; second, the process V can be only positive, which is not a considered case in Theorem 4.20[13].

From Figure 4.3 we can see that, as expected, the relative error with respect to the

[13] At any step, to build the r.v. $Y_{(\cdot)}$, we consider only the points $\{l \in \gamma\mathbb{Z}^2 : |l| \leq c\}$ such that $(\cdot) + Y_{(\cdot)}$ is a.s. in $\gamma(\mathbb{N}_+ \times \mathbb{Z})$.

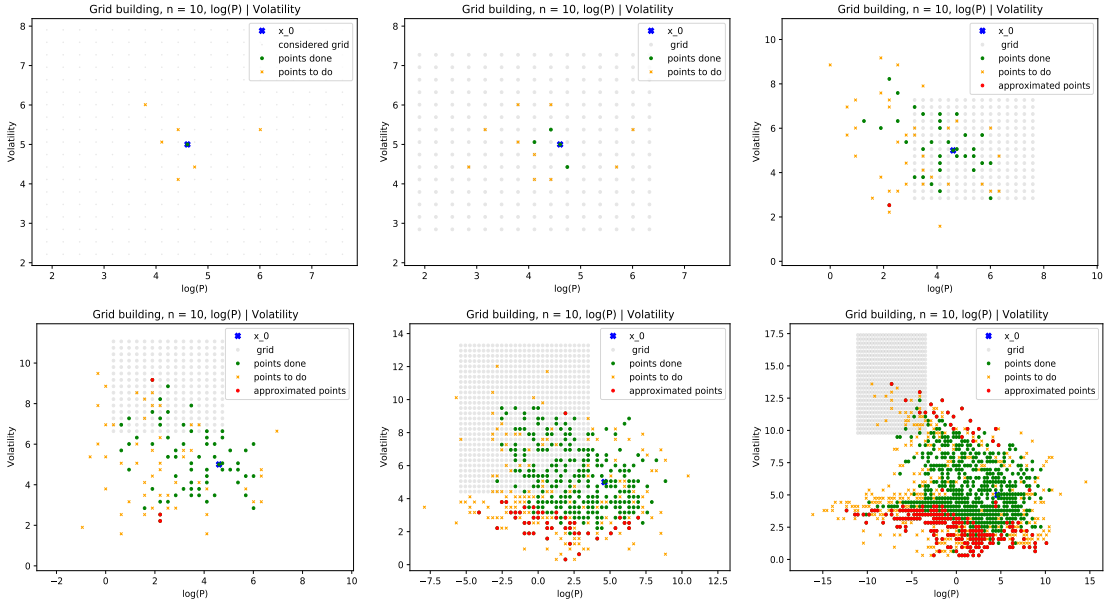


Figure 4.2: Building the transition matrix. Starting from x_0 we consider to build the random variable X_1 such that it has support on the grey points and the first two moments are matched exactly. In the top-left plot, X_1 has support on the yellow points. Then this procedure is iterated for all the possible points in the support of the measure, following Algorithm 7.

computation of some specified functions of $(log(P), V)$ decreases increasing the number of steps required. The only unsatisfying plot is the one relative to the option price, which requires further study. In Figure 4.4, left and middle plot, we study how the cardinality of the state space increases for different value of n.

In Figure 4.4, right plot, we can observe that the cardinality of the states does not seem to depend on the value n of time steps considered, but only on the steps done: although the coefficients are not bounded, from Theorems 4.13, 4.20, 4.22, the map card[support(X_i^n)] should depend on i, the step "at which we are", and n the total number of steps. However, from Figure 4.4 we see that card[support(X_i^n)] can be bounded by a function of only i, as per the case of bounded coefficients. This behaviour can either depend on the particular model considered, or because we use Theorem 2.4. Another possibility is that the bounds found in Theorems 4.13, 4.20, 4.22 are far to be tight.

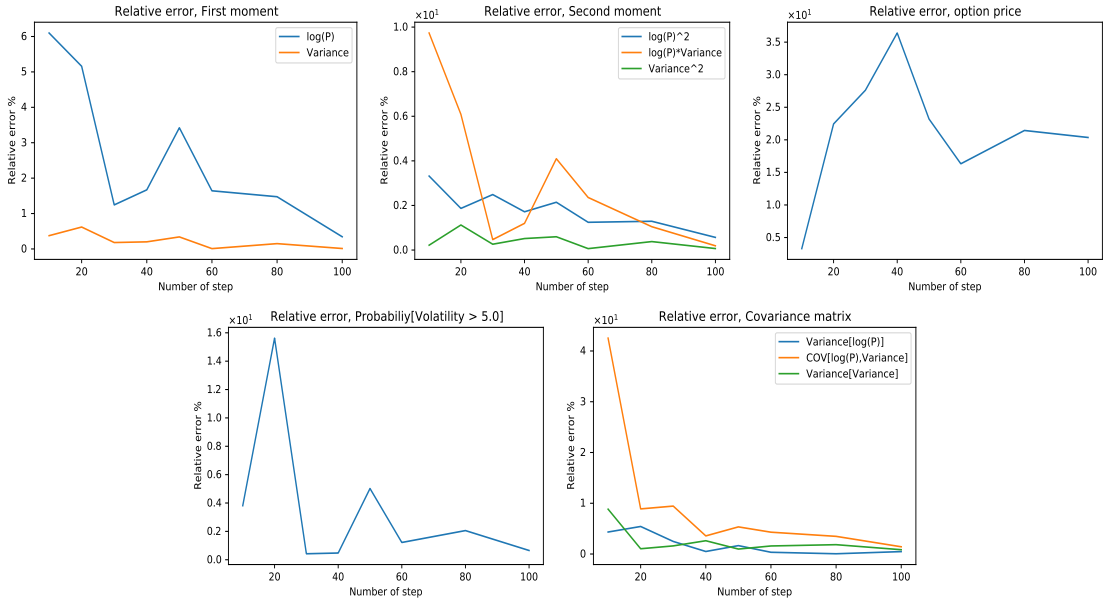


Figure 4.3: Relative errors for some functionals of P_1, V_1. For the option, we consider a European Option with strike 100.

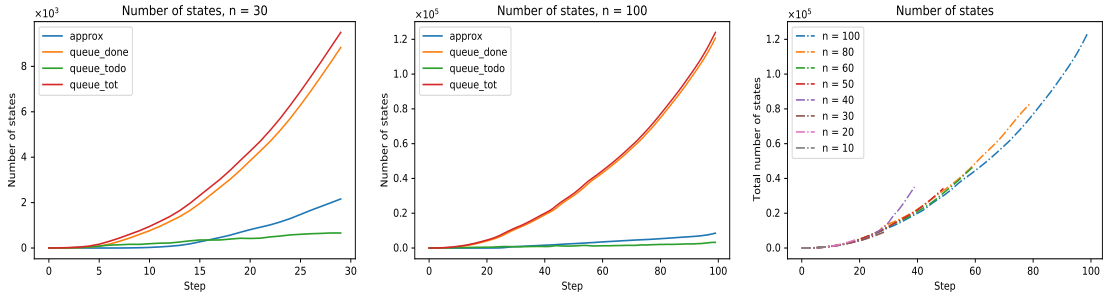


Figure 4.4: The left and middle plots show the growth of the cardinalities of the queues created by Algorithm 7 for fixed n as i increases. The right plot shows a comparison of the growth of the number of states for different n as i increases.

4.8 Summary

The idea of "declaring" from the beginning the lattice on which the Markov chain has support is a proxy to the explosion of the cardinality of the state space. Obviously, there is a price to pay computationally, which is the necessity to build the random variables such that the local consistency assumption is satisfied. Nevertheless, the exponential explosion of the cardinality of the state space with respect to the number of steps makes "not-recombining" Markov chains usable only via Monte Carlo techniques.

To summarise we have found sufficient conditions to build recombining Markov chains,

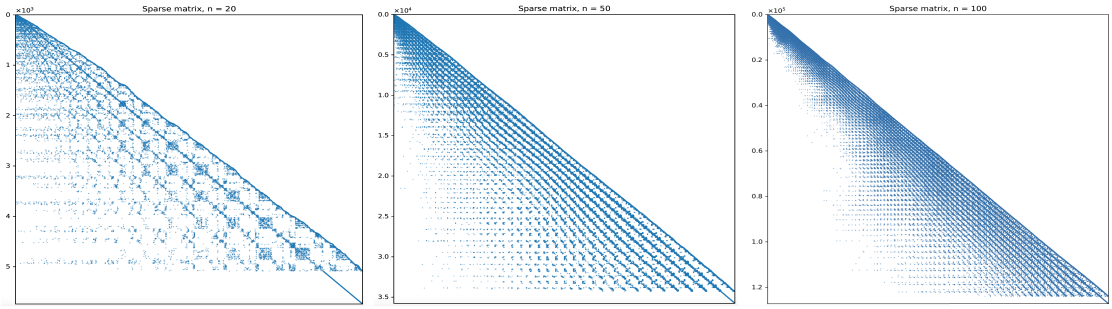


Figure 4.5: Representation as sparse matrix of the Probability Transition Matrix for different values of n.

which satisfy the local consistency assumption on a lattice in dimensions 1 and 2. In dimension d, we have proved a particular case. If one is interested in approximating different processes on the same state-space, we have presented sufficient conditions for the parameter γ such that it is possible to approximate all the processes on the same lattice $\gamma Q\mathbb{Z}^d$. Let us highlight that our procedure does not need the coefficients μ and σ to be smooth and all the transformations used in the literature can be used also with the procedure presented here.

In future works, we want to study the generalization of the procedure to match a higher number of moments. Unfortunately, we think this is much more difficult, see Comon [38], Comon et al. [39] for results regarding the decomposition of symmetric tensors, which however are far to be sufficient for the extension proposed in this Chapter.

4.A Appendix - Lemmas

Lemma 4.24. *If* $\mathbb{E}|X_i^n|^q \leq c(q,T,X_0)$*, then Lemma 4.14-Item* (1) *implies Equation* (4.2)*.*

Proof. Let us recall [98] that if the coefficients $\mu, \sigma \in C_b^4$, we have that for any $x \in \mathbb{R}^d$ and for some $q \geq 0$

$$\begin{aligned}\mathbb{E}[X_{t_{i+1}} - X_{t_i}|X_{t_i} = x] - n^{-1}\mu(x) =& O\left((1+|x|^q)n^{-2}\right),\\ \mathbb{E}[(X_{t_{i+1}} - X_{t_i})^{\otimes 2}|X_{t_i} = x] - n^{-2}\mu(x)^{\otimes 2} + n^{-1}\Sigma(x) =& O\left((1+|x|^q)n^{-2}\right),\end{aligned}$$

in particular the second inequalities implies that

$$\mathbb{E}[(X_{t_{i+1}} - X_{t_i})^{\otimes 2}|X_{t_i} = x] - n^{-1}\Sigma(x) = O\left((1+|x|^q)n^{-2}\right)$$

From Lemma 4.14-Item (1), dividing by n^{-1}, we obtain that

$$\begin{aligned}\mathbb{E}\left[\frac{X_{i+1}^n - X_i^n}{n^{-1}}\middle| X_i^n\right] - \mu(X_i^n) =& O\left((1+|X_i^n|^q)n^{-\alpha}\right),\\ \mathbb{E}\left[\frac{(X_{i+1}^n - X_i^n)^{\otimes 2}}{n^{-1}}\middle| X_i^n\right] - \Sigma(X_i^n) =& O\left((1+|X_i^n|^q)n^{-\alpha}\right).\end{aligned}$$

Taking the square on both sides and then the expectation, using that $\mathbb{E}|X_i^n|^q \leq c(q,T,X_0)$, we obtain the thesis. □

Lemma 4.25. *If* X^n *represents the schemes built in Theorem 4.13, 4.20, 4.22 and* $n\gamma_n^2 < c$ *for some* $c > 0$*, then*

$$\mathbb{E}[\max_i |X_i^n|^{2q}] \leq c(1+|X_0|^{2q}), \quad \text{for } q \geq 1.$$

Proof. Let us first note that

$$X_{i+1}^n - X_i^n | X_i^n = x \sim Y_x$$

In Theorems 4.13, 4.20 and 4.22 we have proved that if the SDE coefficients are bounded

the r.v. Y_x, can be built such that $|Y_x| \leq c_{\mu,\sigma} n^{-1/2} + k\gamma_n$, whilst if the coefficients are linearly bounded then the r.v. Y_x, can be built s.t. $|Y_x| \leq c_{\mu,\sigma}(1+|x|)n^{-1/2} + k\gamma_n$, for some $k > 0$. Since we are interested in the finite time horizon we suppose for simplicity $T = 1$, which means $i \leq n$.

Let us define $\mathcal{Y}_i^n = \sum_{j=0}^{i-1} Y_{X_j^n} - n^{-1}\mu(X_j^n)$. By construction, to satisfy Item Lemma 4.14-(1), we build the r.v. $Y_{(\cdot)}$ s.t. $\mathbb{E}_j Y_{X_j^n} = n^{-1}\mu(X_j^n)$, which implies that $\mathcal{Y}_i^n$ is a martingale. Thus, we can apply the Burkholder-Davis-Gundy (BDG) Inequality

$$
\begin{aligned}
\mathbb{E}\left[\max_{i\leq n} |\mathcal{Y}_i^n|^{2q}\right] \leq c_q \mathbb{E}\left\langle \mathcal{Y}_n^n \right\rangle^q \leq & c_q \mathbb{E}\left[\sum_{j=0}^{n-1} \left| Y_{X_j^n} - n^{-1}\mu(X_j^n)\right|^2\right]^q \\
\leq & c_q n^{q-1} \sum_{j=0}^{n-1} \mathbb{E}\left| Y_{X_j^n} - n^{-1}\mu(X_j^n)\right|^{2q} \\
\leq & c_q n^{q-1}\left[\sum_{j=0}^{n-1} c_{\mu,\sigma}(1+\mathbb{E}|X_j^n|^{2q})n^{-q} + k\gamma_n^{2q}\right] \\
\leq & c_q n^{q-1}\left[nk\gamma_n^{2q} + c_{\mu,\sigma}n^{-q}n + c_{\mu,\sigma}n^{-q}\sum_{j=0}^{n-1}\mathbb{E}|X_j^n|^{2q}\right] \\
\leq & c_q (n\gamma_n^2)^q + c_{q,\mu,\sigma} + c_{q,\mu,\sigma} n^{-1}\sum_{j=0}^{n-1}\mathbb{E}|X_j^n|^{2q} \\
\leq & c_{\mu,\sigma,q}\left(1 + n^{-1}\sum_{j=0}^{n-1}\mathbb{E}|X_j^n|^{2q}\right),
\end{aligned}
$$

where at the 4-th step (3-rd line) we have used the fact that $|Y_{X_j^n}| \leq c_{\mu,\sigma}(1+|X_j^n|)n^{-1/2} + k\gamma_n$ and $\mathbb{E}Y_{X_j^n} = n^{-1}\mu(X_j^n)$, so $n^{-1}\mu(X_j^n) \in$ Convex Hull$\{\text{supp}(Y_{X_j^n})\}$; at the last step we have used the assumption that $n\gamma_n^2 < c$, for some $c > 0$.

We can now proceed to bound $\mathbb{E}\left[\max_{i\leq n}|X_i^n|^{2q}\right]$

$$
\begin{aligned}
\mathbb{E}\Big[\max_{i\leq n}|\, X_i^n|^{2q}\Big] = & \mathbb{E}\left[\max_{i\leq n}\left|X_0 + \sum_{j=0}^{i-1} Y_{X_j^n}\right|^{2q}\right] \\
= & \mathbb{E}\left[\max_{i\leq n}\left|X_0 + \sum_{j=0}^{i-1} n^{-1}\mu(X_j^n) + \sum_{j=0}^{i-1} Y_{X_j^n} - n^{-1}\mu(X_j^n)\right|^{2q}\right] \\
\leq & c_q\left\{|X_0|^{2q} + \mathbb{E}\left[\max_{i\leq n}\left|n^{-1}\sum_{j=0}^{i-1}\mu(X_j^n)\right|^{2q}\right] + \mathbb{E}\left[\max_{i\leq n}\left|\sum_{j=0}^{i-1} Y_{X_j^n} - n^{-1}\mu(X_j^n)\right|^{2q}\right]\right\}
\end{aligned}
$$

$$
\begin{aligned}
&\leq c_q \left\{ |X_0|^{2q} + n^{-2q} n^{2q-1} \sum_{j=0}^{n-1} \mathbb{E} \left| \mu(X_j^n) \right|^{2q} + \mathbb{E} \left[\max_{i \leq n} |\mathcal{Y}_i^n|^{2q} \right] \right\} \\
&\leq c_q \left\{ |X_0|^{2q} + n^{-1} \sum_{j=0}^{n-1} c_{q,\mu,\sigma} \left(1 + \mathbb{E}|X_j^n|^{2q} \right) + c_{\mu,\sigma,q} + c_{\mu,\sigma,q} n^{q-1} n^{-q} \sum_{j=0}^{n-1} \mathbb{E}|X_j^n|^{2q} \right\} \\
&\leq c_q \left\{ |X_0|^{2q} + c_{q,\mu,\sigma} n^{-1} \sum_{j=0}^{n-1} \mathbb{E}|X_j^n|^{2q} + 2c_{q,\mu,\sigma} + c_{q,\mu,\sigma} n^{-1} \sum_{j=0}^{n-1} \mathbb{E}|X_j^n|^{2q} \right\} \\
&\leq c_{q,\mu,\sigma} \left\{ |X_0|^{2q} + 1 + n^{-1} \sum_{j=0}^{n-1} \mathbb{E}|X_j^n|^{2q} \right\}.
\end{aligned}
$$

It remains now to bound $\mathbb{E}|X_i^n|^{2q}$: applying the discrete Grönwall's Lemma 4.26, from the previous equation we know that

$$
\begin{aligned}
\mathbb{E}|X_i^n|^{2q} &\leq \mathbb{E} \left[\max_{1 \leq i \leq n} |X_i^n|^{2q} \right] \\
&\leq c_{q,\mu,\sigma} \left\{ |X_0|^{2q} + 1 + n^{-1} \sum_{j=0}^{n-1} \mathbb{E}|X_j^n|^{2q} \right\} \\
&\leq c_{q,\mu,\sigma} \left\{ |X_0|^{2q} + 1 \right\} + c_{q,\mu,\sigma} \left\{ |X_0|^{2q} + 1 \right\} \sum_{j=0}^{i-1} n^{-1} \exp \left\{ \sum_{j=0}^{n-1} n^{-1} \right\} \\
&\leq c_{q,\mu,\sigma} \left\{ |X_0|^{2q} + 1 \right\} (1 + e).
\end{aligned}
$$

To conclude the proof we can notice that using the Jensen Inequality and the sub-additivity of the root we have that

$$
\begin{aligned}
\mathbb{E} \left[\max_{i \leq n} |X_i^n|^q \right] &\leq \sqrt{\mathbb{E} \left[\max_{i \leq n} |X_i^n|^{2q} \right]} \\
&\leq \sqrt{c_{q,\mu,\sigma} \left\{ |X_0|^{2q} + 1 \right\} (1 + e)} \leq c_{q,\mu,\sigma} \left\{ |X_0|^q + 1 \right\} (1 + e).
\end{aligned}
$$

□

We state here the discrete Grönwall's Lemma that we have used in some previous proofs of this Chapter.

Lemma 4.26. *[Discrete Grönwall's Lemma [83]] Let y_n, f_n, and g_n be non-negative*

sequences such that for $n > 0$, $y_n \leq f_n + \sum_{k=0}^{n} g_k y_k$, *then*

$$y_n \leq f_n + \sum_{k=0}^{n} f_k g_k \exp\left(\sum_{j=k}^{n} g_j\right).$$

5 Grid-Free Computation of Probabilistic Safety Regions with Malliavin Calculus

Keywords: Formal Verification, Malliavin Calculus, SDE, Probabilistic Safety

In this Chapter, we work with continuous-time, continuous-space stochastic dynamical systems described by Stochastic Differential Equations. Related to Chapter 1, we present a new approach to compute probabilistic safety regions, namely the set of initial conditions, associated with trajectories of the SDE, that are safe with a probability larger than a given threshold. We introduce a functional that is minimised at the border of the probabilistic safety region, then solve an optimisation problem that computes such region using techniques from Malliavin Calculus. Unlike existing results in the literature, this new approach allows computing probabilistic safety regions without gridding the state space of the SDE. The Chapter is supported by numerical experiments. This Chapter contains the material presented in Cosentino et al. [42].

5.1 Introduction

In control engineering and in formal verification, a fundamental and common problem is safety analysis: this concerns identifying states of a dynamical model that are safe, namely that are associated to trajectories that do not escape (whether over finite or infinite time) a given set that is deemed to be safe, Abate et al. [2, 4], Blanchini and Miani [18]. Dually, one can express a reachability analysis problem by identifying states that are associated with trajectories entering a given target set. In a probabilistic context, we are interested in characterising and computing the likelihood of the exit time. Being safe with a probability

at least equal to p is also known as p-safe analysis or computation of the p-safe region, Wisniewski and Bujorianu [183]. In this Chapter, we present an application of Malliavin Calculus to the computation of the p-safe region, borrowing ideas from Mathematical Finance and in particular leveraging techniques for the computation of the Greeks of a derivative, to this problem.

Let us consider a d-dimensional Brownian motion $W_t \in \mathbb{R}^d$ defined on a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t), \mathbb{P})$ and the following SDE

$$dX_t = \mu(X_t)dt + \sum_{k=1}^{d} \sigma_k(X_t)dW_t^k, \qquad X_0 = x, \tag{5.1}$$

where the stochastic integral is considered to be the Ito version.

Assumption 5.1. *We consider the vector fields $\mu, \sigma_k, k = 1, 2, \ldots, d$, to have linear bounded growth and to be in $C_b^\infty(\mathbb{R}^d; \mathbb{R}^d)$, the space of infinitely differentiable functions with bounded derivatives from $\mathbb{R}^d$ to $\mathbb{R}^d$. Moreover, if we call $\sigma \in C_b^\infty(\mathbb{R}^d; \mathbb{R}^{d\times d})$ the matrix whose columns are the vectors σ_k, we assume that σ satisfies the uniform ellipticity condition, i.e. $\sigma\sigma^\top$ is uniformly positive definite.*

If Assumption 5.1 holds, then it is well-known, N. Ikeda [134], that the SDE (5.1) has a unique strong solution, Definition 4.1, X_t^x and whenever clear from the context we shall omit the index x.
Let us consider a bounded and smooth region A and let ∂A denote the border of A. We recall that the exit time τ_A^x of X_t^x from the region A is defined as

$$\tau_A^x = \inf\{t \geq 0 \,:\, X_t^x \notin A\}, \qquad x \in A.$$

Whenever clear from the context we shall omit the indexes x, A.
We define $\mathbb{A}_T^p$ to be the p-safe portion of a region A, or equivalently p-safe region of A, as the initial points x in A such that if X starts from x, then it stays in A longer than T with

probability greater than p, i.e.

$$\mathbb{A}_T^p = \{x \in A : \mathbb{P}(\tau^x \geq T) \geq p\}.$$

Whenever clear from the context we shall omit the indexes p, T.

Remark 5.2. *In Wisniewski et al. [187] the authors study a more general problem, namely the probabilistic reach-avoid problem. Given a safe set S and an unsafe set U, they compute the probability to leave S before entering in U, before a pre-specified time T, i.e.* $\{x \in S : \mathbb{P}(\tau_{U^C}^x \leq \tau_S^x, \tau_{U^C}^x < T) \leq q\}$[1]. *If we consider a set U s.t. $U^C = S$, then the wanted quantity becomes $\{x \in S : \mathbb{P}(\tau_S^x < T) \leq q\}$, which is exactly the dual of $\mathbb{A}_T^p$ for $p = 1 - q$. Unfortunately, in [187] an analysis of the approximation error is not presented and, since the approach is radically different from the one presented in this Chapter, a quantitative comparison is difficult.*

The standard way to compute $\mathbb{A}_T^p$, see next Section, is to discretise the region A and to compute the value $\mathbb{P}(\tau^x \geq T)$ at any point in the grid. In the following instead, using ideas from Mathematical Finance and results from the Malliavin Calculus, we show how to compute $\mathbb{A}_T^p$ with a grid-free technique. The approach hinges on the observation that the border of $\mathbb{A}_T^p$ can be expressed as[2]

$$\partial\mathbb{A}_T^p = \arg\min_x \frac{1}{2} \left(\mathbb{P}\left(\tau^x \geq T\right) - p\right)^2. \tag{5.2}$$

The main idea of this approach is thus to solve such optimisation problem: assuming differentiability and excluding convexity issues, we know that, setting up the recursion

$$x_{j+1} = x_j - \lambda \left(\mathbb{P}\left(\tau^x \geq T\right) - p\right) D_x \mathbb{P}\left(\tau^x \geq T\right), \tag{5.3}$$

then $x_j \to x^\star$, where $x^\star \in \partial\mathbb{A}_T^p$, for $\lambda > 0$ small enough. Equation (5.3) represents a

[1] In [187] the authors want to compute $\mathbb{P}(\tau_{U^C}^x < \tau_S^x, \tau_{U^C}^x < T) \leq q$, we have slightly modified using $\leq$, because it does not change the physical meaning and greatly simplifies the comparison between [187] and this Chapter.

[2] We employ here for simplicity a quadratic function $f(x) = 0.5(x-p)^2$, however any other differentiable function f minimised in $x = p$ is also appropriate for the task.

standard Gradient Descent (GD) step. We remark that in principle other optimization algorithms can be used to solve the problem in (5.2); in this Chapter we focus on first-order gradient-based optimization procedures, of which GD is an exemplar. As an alternative instance to standard GD, in the case study we employ ADAM [96], a state-of-the-art optimization procedure, see also Chapter 3.

Using the GD in Equation (5.3) not only allows identifying the set $\mathbb{A}_T^p$; in Section 5.4 we also provide a procedure to explore its border. Furthermore, in Theorem 5.5 we prove that the "*interior*" of the obtained region delimited by $\partial\mathbb{A}_T^p$ is in $\mathbb{A}_T^p$, which implies that there is no need to check these internal points. Moreover, in Proposition 5.6 and Corollary 5.7 we show how to check if a point x is "*inside*" $\mathbb{A}_T^p$ without computing $\mathbb{P}(\tau^x \geq T)$, but only using the gradient at a specific point on $\partial\mathbb{A}_T^p$, which is generated by the optimization procedure.

Grid-based vs -free approaches. The GD step above depends on the two quantities $\mathbb{P}(\tau^x \geq T)$ and $D_x\mathbb{P}(\tau^x \geq T)$: it should be clear that if we can compute them, or approximations thereof, then the overall procedure will result in a grid-free computation of the p-safe portion of the region. An important advantage of such grid-free approach is that if the p-safe region $\mathbb{A}_T^p$ is not empty - however small, even if it was a point - then it will be found. Instead, grid-based approaches, the majority of those presented in the literature (see next Section), will find the set $\mathbb{A}_T^p$ only if it intersects with the introduced grid. If the p-safe region consists of only one point, the procedure introduced here can find it, up to a numerical approximation error related to the approximation of $\mathbb{P}(\tau^x \geq T)$ and $D_x\mathbb{P}(\tau^x \geq T)$, whilst this might not be possible for grid-based approaches unless the grid is selected to intersect such point, which is usually not known beforehand. Another extreme case can be identified when the p-safe region of interest is not bounded: in such case the approach underpinning grid-based methods can be quite inefficient, whilst the grid-free based approach presented here shall converge to its border $\partial\mathbb{A}_T^p$, and explore as much of it as computationally feasible.

In general, a formal comparison between the two approaches can be problematic: whilst grid-free strategies search for solutions within an uncountable infinite set, grid-based

procedures search for solutions within a given, finite set.

Still, we can comment on the computational complexity related to the two different approaches: suppose that we are working with a model of dimension 2, that set $\mathbb{A}_T^p$ consists of only one bounded connected region, and consider a grid over $\gamma\mathbb{Z}^2$, where γ is a scaling parameter. In the following, we will show that the procedure presented in this Chapter requires a step-exploration parameter (again called γ) that can be related to the γ parameter of the grid: they both indicate how precise we want the approximation to be, see Figure 5.1 and Algorithm 8. The points explored by the two methods can be roughly quantified as $O(\text{Area}(A)\gamma^{-2})$ for the grid discretisation - this is the number of points of $\gamma\mathbb{Z}^2$ in the overall (larger) region A - and $O(\text{Len}(\partial\mathbb{A}_T^p)\gamma^{-1}) + C$ for the method here presented - the first term represents the number of points we will explore on $\partial\mathbb{A}_T^p$, whilst C depends on how many points we explore to arrive at the border $\partial\mathbb{A}_T^p$ from the starting point $x_\star$ we choose for the GD procedure. For a model in dimension d, we would have instead $O(\text{Area}(A)\gamma^{-d})$ and $O(\text{Area}(\partial\mathbb{A}_T^p)\gamma^{-d+1}) + C$.[3] As a rough estimate, we can see that, as $\gamma \to 0$, the order of points explored is less with the method here presented.

Literature - Probabilistic Safety. Safety analysis, a standard specification in Formal Verification, has been studied on dynamical models within the Hybrid Systems community. Corresponding safety notions for stochastic models have been introduced only over the past two decades. In particular, in Hu et al. [86] a new modeling framework named stochastic hybrid system (SHS) is presented, which extends with randomness the deterministic framework of hybrid models by allowing the continuous flow inside each invariant set of the discrete state variables to be governed by Stochastic Differential Equations, rather than deterministic ordinary differential equations. In Hu et al. [86], the notion of embedded Markov chain is proposed and the exit probability problem is studied, which is related to reachability analysis: it is shown that this probability converges to its counterpart for the original SHS, as the discretisation step goes to zero. In Bujorianu and Lygeros [31] the models from [86] are extended, setting up Markov strings to obtain a very general class of models for SHS. Closely related to [31], in Bujorianu [32] is introduced a general

[3] This is a slight abuse of notation, indeed $\text{Area}(\partial\mathbb{A}_T^p)$ now represents the Lebesgue measure in $\mathbb{R}^{d-1}$, whist $\text{Area}(A)$ represents the Lebesgue measure in $\mathbb{R}^d$.

model for SHS and proposed a method based on Dirichlet forms, to study the reachability problem over SHS models. Similarly, in Koutsoukos and Riley [105] a method to compute probabilistic reachability is proposed: [105] first show that reachability can be characterised as a viscosity solution of a system of coupled Hamilton-Jacobi-Bellman equations; second, they present a numerical method for computing the solution based on discrete approximations, showing that this solution converges to the one for the original SHS model as the discretisation becomes finer. Riley et al. [155] builds on [105] by employing Monte Carlo (MC) techniques for estimating probabilities of events, and [155] uses multilevel splitting, a variance-reduction technique that can improve both efficiency and accuracy. Again for SHS, in Mohajerin Esfahani et al. [131] is established a connection between stochastic reach-avoid problems - problems encompassing both reachability and safety, also known as constrained reachability problems - and optimal control problems involving discontinuous payoff functions. Focussing on a particular stochastic optimal control problem, the exit-time problem mentioned above, [131] provide its characterisation as a solution of a partial differential equation in the sense of viscosity solutions, along with Dirichlet boundary conditions. In Wisniewski and Bujorianu [183] is established an optimisation scheme for computing probabilistic safety of SHS, combining the use of barrier certificates and of potential theory.

In Wisniewski et al. [186] is presented a method to compute protection certificates that are closely related to the concept of p-safe region. As discussed in Remark 5.2, Wisniewski and Sloth [185], Wisniewski et al. [187] compute the p-safe region based on the extended generator of stochastic dynamical systems; they characterize the safety problem as an optimization problem on the space of positive measures and then solve it via a moment method. In Bujorianu and Wisniewski [33] the p-safe regions are characterized using concepts from Potential Theory.

Alternative techniques have been presented in Hu and Prandini [85], where a real-world probabilistic safety problem has been introduced: the probability of conflict for two aircraft at a fixed altitude. The procedure is based on the introduction of a Markov chain approximation of the stochastic process describing the relative position of the aircraft. Along similar lines, in Prandini et al. [149] the probability of conflict is studied: randomised

algorithms are introduced to efficiently estimate this measure of criticality and to provide quantitative bounds on the level of approximation introduced. Also, approximate closed-form analytical expressions for the probability of conflict are obtained. These randomised approaches can be related to statistical model checking techniques, which have also been developed for models related to SHS in Shmarov and Zuliani [165].

Literature - Mathematical Finance. In Pricing Theory, a branch of Mathematical Finance, a typical problem is to evaluate the variation of the price of a derivative, in response to a change of the underlying asset price or volatility, e.g. Hull [90]. These quantities are known by the name of Greeks and they play a core role in hedging theory. More precisely, given an underlying asset, whose price P_t is the solution of an SDE starting in p_0, the price D of a derivative is given as the expectation of a functional of P_t, i.e.

$$D_{p_0} = \mathbb{E}[f(P_T^{p_0})], \qquad T > 0.$$

If p_0 is the initial price of the underlying asset, the Greek representing the sensitivity with respect to the initial price is called *Delta*, Δ, and is formally defined as

$$\Delta = \frac{\partial D_{p_0}}{\partial p_0} = \frac{\partial \mathbb{E}[f(P_T^{p_0})]}{\partial p_0}.$$

In our problem setup, we are interested in computing the quantity $D_x\mathbb{P}(\tau^x \geq T)$, where τ^x is a specific exit time related to the probabilistic safety property: this is challenging because it involves the derivative of a non-smooth functional $\mathbb{1}$ of the exit time. Through Malliavin Calculus it is possible to provide explicit formulae for the Greeks, Fournié et al. [60, 61], Malliavin and Thalmaier [127].

5.2 Malliavin Calculus for stopping times

We leverage a result from Malliavin Calculus to provide the derivative term $D_x\mathbb{P}(\tau^x \geq T)$ used in (5.3), which otherwise we should estimate numerically, with associated unavoidable

imprecisions.

Under Assumption 5.1, it is possible to show that X_t^x is a.s. differentiable with respect to the starting point x Kunita [107]. Following the notations in Malliavin and Thalmaier [127], let us introduce $J_t = D_x X_t$ and $\boldsymbol{\mu} = D_x[\mu(x)]$, $\boldsymbol{\sigma}_k = D_x[\sigma_k(x)] \in C_b^\infty(\mathbb{R}^d; \mathbb{R}^{d\times d})$; then J_t solves

$$dJ_t = \boldsymbol{\mu}(X_t)J_t dt + \sum_{k=1}^{d} \boldsymbol{\sigma}_k(X_t)J_t dW_t^k, \qquad J_0 = I_d.$$

Theorem 5.3. *Malliavin and Thalmaier [127, Theorem 2.18] If Assumption 5.1*[4] *holds true, calling τ^1 the time when $\int_0^{\tau^1} dist(X_t, \partial A)^{-2} dt = 1$, then*

$$\frac{\partial}{\partial \varepsilon}\mathbb{P}\left[\tau^{x+\varepsilon\varsigma} \geq T\right]\Big|_{\varepsilon=0} = \mathbb{E}\left[\mathbb{1}_{\{\tau^x \geq T\}} H_{\varsigma,T}\right],$$
$$H_{\varsigma,T} = \sum_{k=1}^{d} \int_0^T \beta_t^k \frac{\mathbb{1}_{\{t<\tau^1\}}}{dist(X_t, \partial A)^2} dW_t^k,$$

where $\beta_t^k \in \mathbb{R}^d$ is the stochastic process satisfying $\sum_{k=1}^d \beta_t^k \sigma_k(X_t) = J_t \cdot \varsigma$.

Selecting the directions $\varsigma = e_i, i = 1, \ldots, d$, we can obtain the gradient $D_x\mathbb{P}(\tau^x \geq T)$, which lies at the core of our procedure in Equation (5.3).

Therefore, we have that $D_x\mathbb{P}(\tau^x \geq T) = \mathbb{E}\left[\mathbb{1}_{\{\tau^x \geq T\}} H_T\right]$, where

$$H_T = \int_0^T \frac{\mathbb{1}_{\{t<\tau^1\}}}{\text{dist}(X_t, \partial A)^2} \beta_t \cdot dW_t, \tag{5.4}$$
$$\beta_t = \sigma^{-1}(X_t) \cdot J_t,$$

and σ is the matrix whose columns are the vectors σ_k. Please note that the dimensions of β_t, H_t in Equation (5.4) and Theorem 5.3 are different.

[4] Notice that these are not minimal assumptions: see Malliavin and Thalmaier [127] for weaker hypotheses and a more general theorem.

5.3 Properties of the region $\mathbb{A}_T^p$

Whilst Theorem 5.3 can be useful for the problem at hand, from an algorithmic point of view there are still a few subtle points to be handled.

Firstly, we do not know whether the quantity $\mathbb{P}(\tau^x \geq T)$ is convex or not. Nevertheless, we know that x is in $\partial\mathbb{A}_T^p$ if $\mathbb{P}(\tau^x \geq T) = p$, which implies that the quantity in (5.3) $\left(\mathbb{P}(\tau^x \geq T) - p\right) D_x\mathbb{P}(\tau^x \geq T) = 0$, regardless of the value of the gradient $D_x\mathbb{P}(\tau^x \geq T)$. Therefore, if we end at a point x where

$$\mathbb{P}(\tau^x \geq T) \neq p \text{ and } \left(\mathbb{P}(\tau^x \geq T) - p\right) D_x\mathbb{P}(\tau^x \geq T) = 0,$$

then we know that $x \notin \partial\mathbb{A}_T^p$, thus we are in a local saddle or local maximum point.

Secondly, we observe that the GD scheme in (5.3) converges to a point, however in general it does not "discover" the entire border $\partial\mathbb{A}_T^p$. Besides, if $\mathbb{A}_T^p$ is the union of two (or more) disconnected regions, then the GD scheme will converge solely to one of them. The former issue can be mitigated algorithmically, by finding a way to "explore" the border defined by the condition $\{\mathbb{P}(\tau^x \geq T) = p\}$: this is discussed in the next Section. However, we cannot in general solve the latter problem, which is related to the issue of convergence to local-vs-global optima, which is intrinsic to GD schemes.

Still, we shall shed some further light on the shape of $\mathbb{A}_T^p$. Let us start noticing that if $x \in A$, then $\mathbb{P}(\tau^x \geq T) \geq 0$, therefore for any $p > 0, T > 0$ s.t. $\mathbb{A}_T^p \neq \emptyset$,

$$\mathbb{A}_T^p \subseteq A = \mathbb{A}_T^0.$$

However, we cannot be sure that the p-safe region $\mathbb{A}_T^p$ is a connected set, as we can, in general, argue that $\mathbb{A}_T^p = \cup_i \mathbb{A}_i$, namely $\mathbb{A}_T^p$ consists possibly of a countable union of sets, wherein any $\mathbb{A}_i$ is a bounded connected set. Each component $\mathbb{A}_i$ is endowed with interesting properties.

Definition 5.4. *We say that a surface*[5] *is closed if it partitions the space* $\mathbb{R}^d$ *into one*

[5] See e.g. [97] for a formal definition of surface.

bounded connected region and one unbounded region. We denote this bounded region as the interior of the surface.

Theorem 5.5 (No holes)**.** *Let the $\partial\mathbb{A}_i$ be a closed surface such that $\partial\mathbb{A}_i \subseteq \partial\mathbb{A}_T^p$. Then, the interior of $\partial\mathbb{A}_i$ is in $\mathbb{A}_T^p$.*

Proof. Let us indicate with $\mathbb{A}_i$ the interior of $\partial\mathbb{A}_i$. We prove the thesis if for any $x \in \mathbb{A}_i$, $\mathbb{P}(\tau_A^x \geq T) \geq p$ – we omit the index A in the next steps. If we define θ to be the exit time from $\mathbb{A}_i$, then

$$\begin{aligned}\mathbb{P}\left(\tau^x \geq T\right) =& \mathbb{P}\left(\tau^{X_\theta^x} \geq T - \theta^x \middle| \theta^x \leq T\right)\mathbb{P}\left(\theta^x \leq T\right) + \mathbb{P}\left(\tau^x \geq T \middle| \theta^x \geq T\right)\mathbb{P}\left(\theta^x \geq T\right) \\ =& \mathbb{P}\left(\tau^{X_\theta^x} \geq T - \theta^x \middle| \theta^x \leq T\right)\mathbb{P}\left(\theta^x \leq T\right) + \mathbb{P}\left(\theta^x \geq T\right),\end{aligned}$$

where thanks to the definition of θ^x, τ^x we have that $\mathbb{P}\left(\tau^x \geq T \,|\, \theta^x \geq T\right) = 1$, indeed $\theta^x \leq \tau^x$ a.s. since $\mathbb{A}_i \subseteq A$ and by definition of exit time.
Since $\theta^x \geq 0$ a.s., $\mathbb{P}(\tau^{X_\theta^x} \geq T - \theta^x \,|\, \theta^x \leq T) \geq \mathbb{P}(\tau^{X_\theta^x} \geq T \,|\, \theta^x \leq T)$ then

$$\begin{aligned}\mathbb{P}\left(\tau^x \geq T\right) \geq& \mathbb{P}\left(\tau^{X_\theta^x} \geq T \middle| \theta^x \leq T\right)\mathbb{P}\left(\theta^x \leq T\right) + \mathbb{P}\left(\theta^x \geq T\right) \\ \geq& p\mathbb{P}\left(\theta^x \leq T\right) + p\mathbb{P}\left(\theta^x \geq T\right) \\ =& p,\end{aligned}$$

because $X_\theta^x \in \partial\mathbb{A}_T^p$. □

From an algorithmic point of view, Theorem 5.5 is remarkable: once the algorithm has obtained a closed surface for $\{x : \mathbb{P}\left(\tau^x \geq t\right) = p\}$ we know that all the points inside are in $\mathbb{A}_t^p$ without the need to check any further. Nevertheless, let us recall that we cannot know if this is the only part of $\mathbb{A}_T^p$ as there could be other sets in A, not connected with the one just found.

Once we have identified (part of) $\mathbb{A}_T^p$, an important question is how to check if a point lies inside $\mathbb{A}_T^p$. There are different ways to check if a point is inside a region, such as the winding number, or the Point-in-Polygon algorithm, e.g. [87, 106] , but computationally

these methods are quite expensive and generalizations to dimensions greater than 3 do not seem to be treated in the literature, at least from an algorithmic point of view.

Remember that to compute $\partial\mathbb{A}_T^p$, we use a gradient-based optimization algorithm, requiring the computation of the quantity $D_x\mathbb{P}(\tau^x \geq T)$ for any point in the sequence (5.3). Hence, it would be useful to understand if one point is inside the safety region using the information given by $D_x\mathbb{P}(\tau^x \geq T)$: this is handled by the next result.

Proposition 5.6. *Let us suppose that a region $A \in \mathbb{R}^d$ is defined by a differentiable function α, i.e. $A := \{x : \alpha(x) \leq 0\}$ and $\partial A := \{x : \alpha(x) = 0\}$. Moreover, let us suppose that A is connected. Then, a point x is inside A if*

$$x = x^\star - \|x - x^\star\| \frac{D_x\alpha(x)\Big|_{x=x^\star}}{\left\|D_x\alpha(x)\Big|_{x=x^\star}\right\|},$$

where $x^\star := \arg\min_{y\in\partial A} \|x - y\|$. If instead

$$x = x^\star + \|x - x^\star\| \frac{D_x\alpha(x)\Big|_{x=x^\star}}{\left\|D_x\alpha(x)\Big|_{x=x^\star}\right\|},$$

then x is outside.

Proof. Let us consider $S = S(x, \|x - x^\star\|)$ the open sphere with center x and radius $\|x - x^\star\|$; we know that if x is in A then $S \subset A$, vice versa $S \subset A^C$ if x is outside A.
Note that $x - x^\star$ is perpendicular to the tangential plane to α in $x^\star$, as it is also the gradient $D_x\alpha(x)\Big|_{x=x^\star}$, therefore

$$x = x^\star \pm \|x - x^\star\| \frac{D_x\alpha(x)\Big|_{x=x^\star}}{\left\|D_x\alpha(x)\Big|_{x=x^\star}\right\|}.$$

Since A is connected, sign $\{\alpha(x)\}$ is the same for any $x \in S$ and given that $\alpha(x^\star) = 0$ the sign can be deduced by the direction of the gradient, which means that if the $D_x\alpha(x)\Big|_{x=x^\star}$ points to x than $\alpha(x) \geq 0$ and $x \notin A$, if the $-D_x\alpha(x)\Big|_{x=x^\star}$ points to x than $\alpha(x) \geq 0$ and $x \in A$. □

Since we know from Theorem 5.5 that any portion $\mathbb{A}_i$ of $\mathbb{A}_T^p$ is connected, once we have found a closed surface bordering $\mathbb{A}_i$, then thanks to Proposition 5.6 we know how to check if a point x is inside $\mathbb{A}_i$ by estimating the gradient in $\partial\mathbb{A}_i$, which we compute during the optimization procedure. This means that we do not have to compute $\mathbb{P}(\tau^x \geq T)$. Unfortunately, we cannot know a-priori if it is outside because we do not know beforehand whether $\mathbb{A}_T^p$ is connected or not.

Corollary 5.7. *Let the $\partial\mathbb{A}_i$ be a closed surface such that $\partial\mathbb{A}_i \subseteq \partial\mathbb{A}_T^p$ and $\mathbb{A}_i$ its interior. Denoting by $x^\star := \arg\min_{y\in\partial\mathbb{A}_i} \|x-y\|$, then a point x is inside $\mathbb{A}_i$ if*

$$x = x^\star - \|x - x^\star\| \frac{D_x\mathbb{P}(\tau^x \geq T)\Big|_{x=x^\star}}{\left\|D_x\mathbb{P}(\tau^x \geq T)\Big|_{x=x^\star}\right\|}.$$

Proof. The proof follows closely Proposition 5.6 considering $\mathbb{A}_i$ in place of A. The difference is that the sign of the points in S is the same for the points inside, whilst we cannot say the same if x is outside the region $\mathbb{A}_i$. It could be that x is outside $\mathbb{A}_i$, but $S \cap \mathbb{A}_j \neq \emptyset$ for some $j \neq i$. □

5.4 Exploration of the border $\partial\mathbb{A}_T^p$

In this Section we provide an algorithmic solution to the problem discussed above, namely how to explore points on the border characterised by

$$\arg\min_x \frac{1}{2}\left(\mathbb{P}\left(\tau^x \geq t\right) - p\right)^2.$$

In the following we treat the case of models with dimension $d = 2, 3$; if the model dimension is greater than 3, the procedure can be iterated.

Dimension 2. If $d = 2$ then Algorithm 8 explains how to proceed quite easily.
Let us discuss the main steps of Algorithm 8:

(i) Line 5. If we move along the border of the region $\mathbb{A}_i$ just found, in a, say, clockwise manner, thanks to Theorem 5.5 we know that we can stop when we have found a

Algorithm 8 Exploration of the border $\partial\mathbb{A}_T^p$

1: Initialize x_0
2: Run the GD from x_0 up to a point $x_\star$ such that $\mathbb{P}(\tau^{x_\star} \geq T) \approx p$
3: $x \leftarrow x_\star$
4: step $\leftarrow 1$
5: **while not** ($x \approx x_\star$ **and** step>step_min) **do**
6: Append x to $\partial\mathbb{A}_T^p$
7: Move in a direction dir perpendicular to $D_x\mathbb{P}(\tau^x \geq t)$
8: $x \leftarrow x + \gamma \times \frac{dir}{\|dir\|}$
9: step $\leftarrow$ step $+1$
10: **end while**

closed point ($x \approx x_\star$) and all the points inside are in $\mathbb{A}_i$. Given that the set could be the union of different disjoint sets, we should still explore the rest of the region A, i.e. $A \setminus \mathbb{A}_i$.

(ii) Line 8. Defining $x_0 = x + \gamma \times dir$ we know that

$$\begin{aligned}\mathbb{P}(\tau^{x_0} \geq t) &\approx \mathbb{P}(\tau^x \geq t) + D_x\mathbb{P}(\tau^x \geq t)(\gamma \times dir) + \|\gamma \times dir\|^2 \mathrm{ERR} \\ &\approx p + \|\gamma \times dir\|^2 \mathrm{ERR},\end{aligned}$$

which means that for small γ we do not go far from the border $\partial\mathbb{A}_t^p$. This seems the best we can do without computing further derivatives (other than the gradient). ERR represents the error term of a Taylor expansion. It is important to remark that γ and λ are two different parameters, which can be chosen independently, however for more insights see Section 5.5.

Dimension 3 In dimension 3 we can explore the desired border along its "sections". Without loss of generality, let us suppose that the region A is the sphere of center 0 and radius 1. Let us fix the discretization parameter $\delta > 0$, which is related to the error we can tolerate. We can discretize the first directions x_1 to create the planes $x_1 = \pm i\delta, i \in \mathbb{N}, 0 \leq i \leq \delta^{-1}$. The sections of the border are therefore the curves resulting from the intersections between the border and the considered planes. We thus run Algorithm 8 constrained on any given plane $x_1 = \pm i\delta$ that we are considering, see Figure 5.2. Then, the same must be done for the other directions x_2 and x_3. Note that, if we have already computed a "section", e.g.

for the plane $x_1 = 0$, then this information can be very useful for the computation of the close sections, e.g. $x_1 = \pm\delta$.

There are two generalizations to this procedure. Firstly, we can consider alternative directions: instead of selecting directions x_1, x_2, x_3 corresponding to the vectors in the canonical basis (e_1, e_2, e_3), we can consider a general basis of $\mathbb{R}^3$ and derive directions therefrom. Secondly, in order to obtain a grid-free approach to safety analysis if the dimension is beyond 2, instead of constraining the GD on planes, we can constrain the GD on more general regions, e.g. on the regions $x_j \in [i\delta, (i+1)\delta]$, $i \in \mathbb{N}, 0 \leq i \leq \delta^{-1}, j \in \{1, 2, 3\}$.

Note that once we select a plane (say ϕ, or a region) it could happen that $\min_{x\in\phi} \mathbb{P}(\tau^x \geq T) < p$, which means that there is no intersection between ϕ and $\mathbb{A}_t^p$ and we must pass on to examining another plane (or region).

Recursion for higher dimensions We can apply the same reasoning on models with any dimension: namely, if we are in $\mathbb{R}^d$, then we can partition the considered region A in sets of dimension $d-1$. Continuing this procedure we can go back recursively to the case $d = 2$.

Alternative approaches for higher dimensions An alternative grid-free approach is to "explore the border" without constraints relatively to some sections, i.e. generalize directly the case $d = 2$. Let us suppose that $x_\star$ is a point on the border; then we can compute $d-1$ orthonormal vectors $\{g_1^\star, \ldots, g_{d-1}^\star\}$ to $D_x\mathbb{P}(\tau^x \geq T)|_{x=x_\star}$, thus running the procedure recursively from any new point $x_\star \pm \gamma g_j^\star, j = 1, \ldots, d-1$, until we obtain a closed surface. However, attention is needed with the selection of the orthonormal points $\{g_1^\star, \ldots, g_{d-1}^\star\}$: indeed, when $d > 2$ there are infinitely many possibilities, but it would be convenient to find a possible "orientation" such that the exploration of the border is done in an orderly way - see the discussion relative to Line 5 of Algorithm 8 above.

5.5 Experiments

In this Section, we present a toy example. For the experiments, we use a simulation-based approach for its simplicity, i.e. we use Monte Carlo (MC) techniques, and to reduce the

variance we use antithetic Brownian paths, Glasserman [69], Huynh et al. [91]. The code can be found at github.com/FraCose/Grid-free_prob_safety.

Remark 5.8. *We remark that the way* $\mathbb{P}(\tau^x \geq T)$ *and* $D_x\mathbb{P}(\tau^x \geq T)$ *are computed it is not relevant for the idea presented in this Chapter. Indeed, it is enough to be able to compute the quantities* $\mathbb{E}[\mathbb{1}_{\{\tau \geq T\}}]$ *and* $\mathbb{E}[\mathbb{1}_{\{\tau \geq T\}} H_T]$ *and plug them into the GD procedure. We refer to Andersson and Kohatsu-Higa [7], Frikha et al. [64], Henry-Labordère et al. [78] for the exposition of unbiased simulation methods. Other methods to compute these quantities are PDE techniques, which we expect to be computationally heavier.* □

Before presenting the case study, it is important to draw some considerations.

Complexity. Let us recall the definition of H_t and β_t:

$$H_T = \int_0^T \frac{\mathbb{1}_{\{t<\tau^1\}}}{\text{dist}(X_t, \partial A)^2} \beta_t \cdot dW_t,$$
$$\beta_t = \sigma^{-1}(X_t) \cdot J_t.$$

Computing H_t can be expensive. To estimate the expectation via MC methods we use N simulations and a time discretization step of n^{-1}, i.e. we split the time interval [0,T] in n steps. The stochastic processes to be simulated are X_t, J_t, β_t, H_t and $\text{dist}(X_t, \partial A)$. The realization of the stochastic process β has a total cost of $O(Nnd^3)$, where d^3 is the cost related to the matrix inversion σ^{-1}, plus matrix multiplications. Moreover, an optimization problem to compute $\text{dist}(X_t, \partial A)$ has to be solved $O(Nn)$ times. Nevertheless, we have to simulate H_t, J_t only if $t < \tau_1$. It is important to remark that we have analysed the computational cost of computing only one step of the gradient descent procedure, but many are necessary to converge and explore the space.

If we are interested in a relatively low-dimensional problem the matrix inversion can be solved analytically, or leveraging special forms of σ, e.g. tri-diagonal, upper(lower)-triangular. This increases the stability of the procedure and reduces in part its complexity, although the overall complexity remains $O(Nnd^3)$, being dominated by matrix multiplications. A second improvement is to consider particular forms for the region A that can be advantageous for computing $\text{dist}(X_t, \partial A)$, e.g. a sphere, a parallelepiped or a

simplex – although non-smooth regions are not covered by the assumptions of this Chapter. Furthermore, both the arguments just discussed allow the usage of GPU acceleration more easily, which "artificially" reduces the complexity in N.

Bias. It is important to remark that the steps done by the Gradient Descent algorithm are stochastic and biased. Indeed, we do not compute the exact probability $\mathbb{P}(\tau^x \geq T)$, but we discretize the time, therefore computing $\mathbb{P}(\tau_n^x \geq T)$; recall that in [70] it is shown that

$$|\mathbb{P}(\tau^x \geq T) - \mathbb{P}(\tau_n^x \geq T)| \leq O(n^{-1/2}),$$

where τ_n represents the discrete stopping time of the Euler Scheme associated with Equation (5.1). Moreover, the algorithm is stochastic since we approximate $\mathbb{P}(\tau_n^x \geq T)$ using MC techniques. Therefore we have to consider that [70]

$$\left|\mathbb{P}\left(\tau^x \geq T\right) - \mathbb{P}\left(\widehat{\tau_n^x \geq T}\right)\right| \leq O(n^{-1/2}) + O\left(\frac{1}{\sqrt{N}}\right) Z,$$

where the hat denotes an MC estimator of the quantity of interest, Z represents a standard normal random variable, and N is the number of simulations. A similar error bound might be derived for the other term $\left|D_x\mathbb{P}\left(\tau^x \geq T\right) - D_x\widehat{\mathbb{P}\left(\tau_n^x \geq T\right)}\right|$ [70, 72, 73], though an adaptation is needed due to the presence of τ_1 in the definition of H in Theorem 5.3. Due to these biases, we have noticed that reducing the variance helps the GD to converge better (cf. use of antithetic Brownian paths mentioned above): for instance, when the (norm of the) gradient $D_x\mathbb{P}\left(\tau^x \geq T\right)$ becomes small, the error could dominate and the gradient descent step could not work properly; this is especially the case when we simulate paths starting from points close to the border of A.

Hyper-parameters. The hyper-parameters to be chosen for the procedure are the following:

(i) n, time discretisation step - in principle the higher the better, but n has a big impact on the computational time since it cannot be parallelised. Through experiments, we have learnt to start with a relatively fine time discretisation step.

(ii) N, Monte Carlo simulations - increasing N reduces the variance of the MC methods. N has a relatively low impact since the number of samples can be parallelised using a GPU.

(iii) λ, the "learning rate" of the GD procedure in Equation (5.3) - λ must be chosen carefully. While we are doing the first minimization, i.e. while we are searching for a first point on $\partial\mathbb{A}_T^p$ (exploration phase), λ can be quite high (more than 1×10^{-3}, as suggested in Kingma and Ba [96]). Instead, if we are considering the minimization procedure in Algorithm 8, since we should be already close to the border we should select a small λ.

(iv) γ, the "border exploration" parameter in Algorithm 8 - γ indicates how fine-grained we wish the approximation of $\partial\mathbb{A}_T^p$ to be. If it is selected to be excessively small, the walk on the border $\partial\mathbb{A}_T^p$ will be quite slow.

Accelerating the walk. Algorithm 8 is a good starting point to walk on the border $\partial\mathbb{A}_T^p$, however in practice care must be taken. For the following discussions, we consider the problem to be in a 2-dimensional space as a base case.

First, we would like to walk with an orientation, e.g. clockwise, such that we do not go back to a region already explored. This can be done in principle, but sometimes the gradient approximation can be (quite) wrong, especially close to the border of the considered region A or because the chosen discretization time step n is too coarse. To solve this problem, we check if there are points already computed in the direction we are going to explore. However, selecting an "optimal" number of points is an open question that depends on the curvature of $\partial\mathbb{A}_T^p$, which a-priori is unknown. Another heuristic is to constrain the algorithm to search the new point on $\partial\mathbb{A}_T^p$ in a given region, see below and Figure 5.1. A more sophisticated alternative is to split the region into subspaces and search the border $\partial\mathbb{A}_T^p$ locally. This technique would also increase the level of parallelisation, Suffern [172].

Secondly, going towards the direction perpendicular to the gradient, see Section 5.4, is only an approximation and sometimes, depending on the local curvature, can be quite imprecise. To improve this approximation we have considered the following procedure.

Let us imagine that we have computed a certain number of points on $\partial\mathbb{A}_T^p$, in order $\{x_1, x_2, ..., x_m\}$. We can compute the parabola equation (since the plane is fixed) that approximates the points $\{x_1, x_2, ..., x_m\}$, and use this equation as a second possible approximation. This can be thought an approximation of the second-order information of the curve $\mathbb{P}(\tau_n^x \geq T) - p = 0$ in x. Later, we can choose the new direction as a weighted average of the perpendicular to $D_x\mathbb{P}(\tau_n^x \geq T)$ and the value of the approximated parabola $p(x)$ in $x = 2x_m - x_{m-1}$. As there are several ways to compute the weights, we use the past distances between the points found on $\partial\mathbb{A}_T^p$ and the forecasts relative to the gradient and the parabola approximation, see the code for more insight. In this way, when the curvature of $\partial\mathbb{A}_T^p$ "changes" the algorithm starts following more closely the gradient (if the approximation error is low), otherwise it follows an average which experimentally is closer to the parabola forecast. Experimentally, this procedure accelerates the walk, since it reduces the approximation error relative to the gradient.

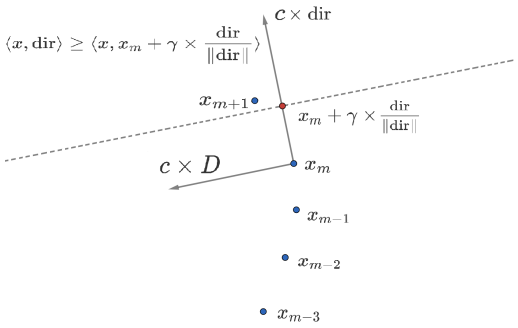


Figure 5.1: In the picture, it is shown a representation of how the algorithm walks on the border: given the points $\{x_m, x_{m-1}, \ldots\}$ already found by the algorithm, it proposes the red point $x_m + \gamma \times \text{dir}/\|\text{dir}\|$ as the new point of the region, and from there it runs the GD to find the new point x_{m+1}. It is possible that on the half-plane where the algorithm looks for the new point does not exist a point of $\partial\mathbb{A}_T^p$, therefore it is necessary to update the constraint, see Algorithm 9.

Finally, it is better to constraint the space where the algorithm searches for the next point of the border. In Figure 5.1 it is shown how we proceed. Once one point x_m on the border is found, i.e. $\mathbb{P}(\tau_n^{x_m} \geq T) \approx p$, we compute the gradient (D) and the direction to follows[6] (dir). Given dir and γ, it is possible to search the new point only in the part

[6]Possibly as a weighted average of the gradient and some local approximation of the curvature as

of plane where there are not "recent points" considering the line perpendicular to the direction passing through the point guess $x_m + \gamma \times \text{dir}/\|\text{dir}\|$.

It is possible that the constraint does not allow the optimization procedure to find a point $\mathbb{P}(\tau_n^{x_\star} \geq T) \approx p$, therefore if the solution of the GD returns, after a certain number of iterations, a point $x_\star$ s.t. $\mathbb{P}(\tau_n^{x_\star} \geq T) \not\approx p$, then it is important to update the direction dir and the corresponding constraint. The candidate we have chosen for the updated direction is $2x_\star - x_m$, up to some rescaling, but other choices are available. For example, we have experimented that selecting $2x_\star - x_m$ accelerates the procedure over the choice $x_\star - x_m$. Moreover, it is necessary to reduce the step exploration γ, such that we get closer to the point x_m and by continuity of $\partial \mathbb{A}_T^p$ we will find the point sooner or later. In Algorithm 9 we present a pseudo-code of the procedure.

Algorithm 9 Adaptive constraint for the GD procedure

Given x_m
Run the constrained GD from $x_m + \gamma \frac{dir}{\|dir\|}$ up to a point $x_\star$
step $\leftarrow 1$
while $\mathbb{P}(\tau^{x_\star} \geq T) \not\approx p$ **do**
    dir $\leftarrow 2x_\star - x_m$
    $\bar{x} \leftarrow x_m + \frac{\gamma}{2 \cdot \text{step}} \cdot \frac{dir}{\|dir\|}$
    Update the plane using the new dir and $\bar{x}$
    Run the constrained GD from $\bar{x}$ up to a point $x_\star$
    step $\leftarrow$ step $+ 1$
end while

If the direction guides towards points already explored recently, because for instance the discretization error is too high or due to the constrained updates in Algorithm 9, given that on a plane the perpendicular vectors to a vector are two, it is enough to invert the direction.

Case study. The model considered for the experiment is:

$$\begin{pmatrix} dX_t^{(1)} \\ dX_t^{(2)} \\ dX_t^{(3)} \end{pmatrix} = \begin{pmatrix} X_t^{(1)} \\ \frac{1}{2}X_t^{(1)} + \frac{1}{2}X_t^{(2)} \\ \frac{1}{3}X_t^{(1)} + \frac{1}{3}X_t^{(2)} + \frac{1}{3}X_t^{(3)} \end{pmatrix} dt + \frac{1}{3} \begin{pmatrix} \omega_1, & \omega_2, & \omega_2 \\ \omega_2, & \omega_1, & \omega_2 \\ \omega_2, & \omega_2, & \omega_1 \end{pmatrix} dW_t \tag{5.5}$$

$$\omega_1 = 2\sqrt{1-\rho} + \sqrt{1+2\rho},$$

explained before.

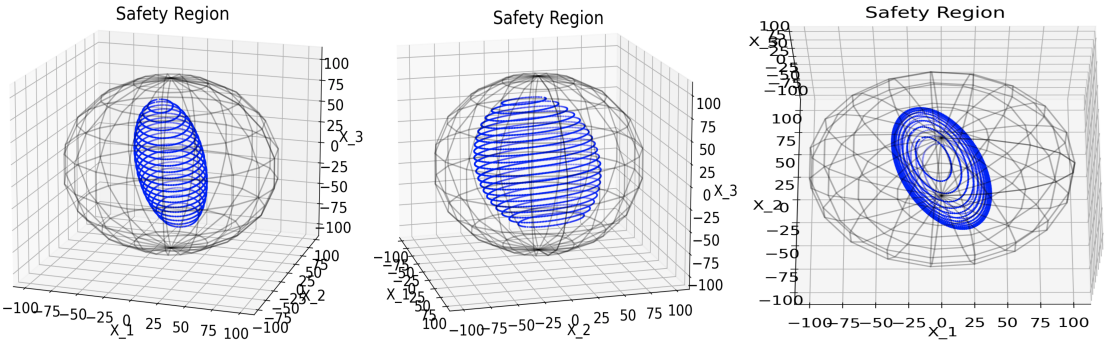


Figure 5.2: The plots show the surface $\partial\mathbb{A}_1^{0.5}$ (in blue) when A is a sphere. We have computed it decomposing the region A (the sphere in black) in sections. However, in the future we aim to exploit the grid-free approach, as explained at the end of Section 5.4.

$$\omega_2 = -\sqrt{1-\rho} + \sqrt{1+2\rho}$$

If we define $d\tilde{W}_t = \sigma dW_t$, where σ is the diffusion matrix in Equation (5.5), then we have that $\text{Corr}(d\tilde{W}_t^{(i)}, d\tilde{W}_t^{(j)}) = \rho$, $i \neq j$ and $i, j \in \{1, 2, 3\}$. In the experiment we have used $\rho = 0.5$. For the region A, we have considered two cases: a sphere with center at the origin and radius equal to 100 and a cube with vertices between $(-100, -100, -100)$ and $(100, 100, 100)$. Note that in the second experiment (A being a cube) the assumptions of the theoretical part of this Chapter are not satisfied. Nevertheless, the procedure is still able to walk on the border.

We consider the problem of computing $\mathbb{A}_1^{0.5} = \{x \in A : \mathbb{P}(\tau^x \geq 1) \geq 0.5\}$. We start at a point x_0 where $\mathbb{P}(\tau^{x_0} \geq 1) \neq 0.5$, then we minimize $\frac{1}{2}(\mathbb{P}(\tau^x \geq 1) - 0.5)^2$ until we obtain a point $x_\star$ s.t. $|\mathbb{P}(\tau^{x_\star} \geq 1) - 0.5| < \text{err}$ – in this case err represents the approximation errors due to the computation of $\mathbb{P}(\tau^{x_\star} \geq 1)$. From $x_\star$ we fix $x_\star^{(3)}$ and start Algorithm 8, i.e. we fix the plane $x^{(3)} = x_\star^{(3)}$, see Figure 5.3 for the results of the experiments. See Figure 5.2 for a 3-dimensional representation of (possibly a portion of) $\partial\mathbb{A}_1^{0.5}$ in the case A is a sphere. Instead of using plain Gradient Descent, we use ADAM, [96], a version of GD with momentum and adaptive learning rate. In particular, we prefer to include momentum, because we do not know whether $\frac{1}{2}(\mathbb{P}(\tau^x \geq 1) - 0.5)^2$ is convex.

The hyper-parameters chosen are $N = 10\,000$, $n = 200$, *maximum iteration of the GD* (any time we use it) $= 50$, $\lambda = 5 \times 10^{-2}$, $\gamma = 1.5$. With reference to the previous discussion on the approximation of the second order information of $\partial\mathbb{A}_T^p$, in order to compute the new

direction, i.e. dir in Figure 5.1, in addition to the gradient information, we use also the parabola approximating the previous 4 points found on the border. For more information, we refer to the code at github.com/FraCose/Grid-free_prob_safety.

5.6 Summary

We have presented a new approach to estimate the safety region for SDEs without discretizing the space. We have proved two theorems that allow us to explore only some parts of the region, diminishing the computational load for continuous space problems. To solve the optimization problem we have borrowed techniques and ideas from Malliavin Calculus and Mathematical Finance. We have discussed some problems arising algorithmically and given some possible solutions. However, some work must be done to complete the task in a completely automatic fashion especially in high dimensions.

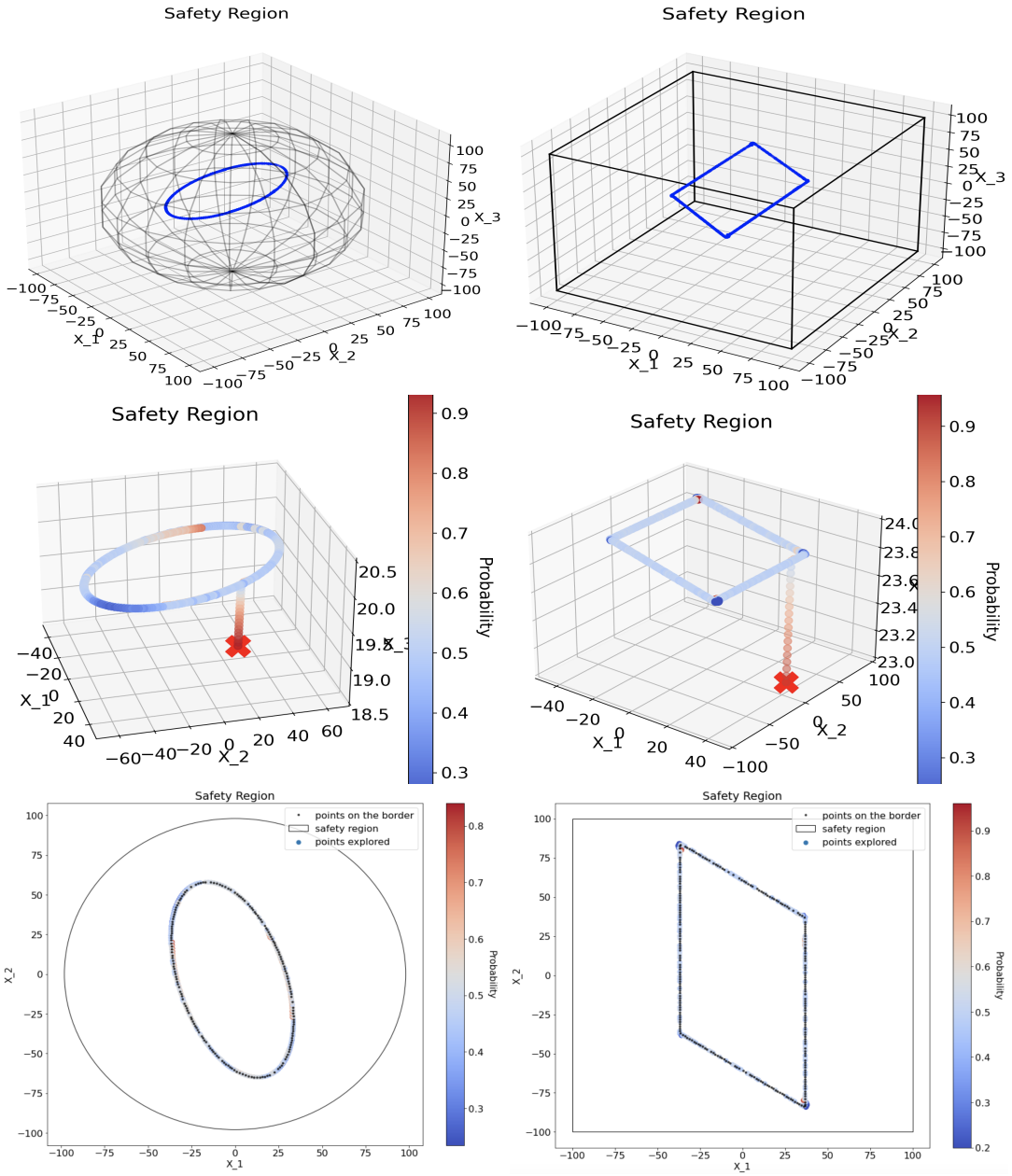


Figure 5.3: The plots show the points found while seeking and exploring the border of the desired 3D regions $\mathbb{A}_1^{0.5}$, with respect to one plane (or section). Left: spherical region A; right: cubic region A. Top: full-view of the problem. Middle: same plot but zoomed, moreover there is a colour scale, showing the safety probabilities of the points explored during the walk. It can be seen how starting from the red-cross point, with a probability of approximately 0.8 (in both cases), we arrive at the points with the desired probability 0.5. The presence of points with different colours to the one corresponding to the desired probability 0.5 means that in those regions, before finding the points with probability 0.5, we have explored via the GD other adjacent points. Bottom: planar section of the space considered, fixing direction x_3. The black points represent the points we consider to be part of the border (up to an approximation error), whilst the circle points are the points explored via the GD method. It can be seen that we explore points associated with probabilities between 0.2 and 0.8.

Bibliography

[1] A. Abate. Probabilistic bisimulations of switching and resetting diffusions. In *49th IEEE Conference on Decision and Control (CDC)*. IEEE, dec 2010.

[2] A. Abate, S. Amin, M. Prandini, J. Lygeros, and S. Sastry. Computational approaches to reachability analysis of stochastic hybrid systems. In *Hybrid Systems: Computation and Control*, pages 4–17. Springer Berlin Heidelberg, 2007.

[3] A. Abate, S. Amin, M. Prandini, J. Lygeros, and S. Sastry. Computational approaches to reachability analysis of stochastic hybrid systems. In *Hybrid Systems: Computation and Control*, pages 4–17. Springer Berlin Heidelberg, 2007.

[4] A. Abate, M. Prandini, J. Lygeros, and S. Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica*, 44(11): 2724–2734, nov 2008.

[5] A. Abate, J.-P. Katoen, J. Lygeros, and M. Prandini. Approximate model checking of stochastic hybrid systems. *European Journal of Control*, 16(6):624–641, jan 2010.

[6] E. Akyıldırım, Y. Dolinsky, and H. M. Soner. Approximating stochastic volatility by recombinant trees. *The Annals of Applied Probability*, 24(5), oct 2014.

[7] P. Andersson and A. Kohatsu-Higa. Unbiased simulation of stochastic differential equations using parametrix expansions. *Bernoulli*, 23:2028–2057, 2017.

[8] archive.ics.uci.edu. 3d road network, north jutland, denmark, . `https://archive.ics.uci.edu/ml/datasets/3D+Road+Network+(North+Jutland,+Denmark)`.

[9] archive.ics.uci.edu. Individual household electric power consumption, . `https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption`.

[10] C. Baier and J.-P. Katoen. *Principles of Model Checking*. The MIT Press, 2008.

[11] Bally. An elementary introduction to Malliavin Calculus. 2003. `https://hal.inria.fr/inria-00071868/document`.

[12] V. Bally and D. Talay. The law of the Euler scheme for stochastic differential equations. I. Convergence rate of the distribution function. *Probability Theory and Related Fields*, 104(1):43–60, 1996.

[13] V. Bally and D. Talay. The law of the Euler scheme for stochastic differential equations. II. Convergence rate of the density. *Monte Carlo Methods and Applications*, 2(2): 93–128, 1996.

[14] C. Bayer and J. Teichmann. The proof of tchakaloff's theorem. *Proceedings of the American Mathematical Society*, 134(10):3035–3041, oct 2006.

[15] C. Bender and P. Parczewski. Discretizing malliavin calculus. *Stochastic Processes and their Applications*, 128(8):2489–2537, aug 2018.

[16] A. Beskos and G. O. Roberts. Exact simulation of diffusions. *The Annals of Applied Probability*, 15(4):2422–2444, nov 2005.

[17] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, may 1973.

[18] F. Blanchini and S. Miani. *Set-Theoretic Methods in Control*. Birkhäuser Basel, 1st edition, 2007.

[19] D. Blatt, A. O. Hero, and H. Gauchman. A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, 18(1):29–51, jan 2007.

[20] L. Bo and C. Sminchisescu. Greedy block coordinate descent for large scale gaussian process regression. *arXiv preprint arXiv:1206.3238*, 2012.

[21] B. Bollobás. Volume estimates and rapid mixing. In *Flavors of geometry*, volume 31 of *Math. Sci. Res. Inst. Publ.*, pages 151–182. Cambridge Univ. Press, Cambridge, 1997.

[22] A. Borodin and P. Salminen. *Handbook on Brownian motion. Facts and formulae.* Birkhauser, Basel, 2002.

[23] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, jan 2018.

[24] B. Bouchard and N. Touzi. Weak dynamic programming principle for viscosity solutions. *SIAM Journal on Control and Optimization*, 49(3):948–962, jan 2011.

[25] P. P. Boyle. A lattice framework for option pricing with two state variables. *The Journal of Financial and Quantitative Analysis*, 23(1):1, mar 1988.

[26] P. P. Boyle, J. Evnine, and S. Gibbs. Numerical evaluation of multivariate contingent claims. *Review of Financial Studies*, 2(2):241–250, apr 1989.

[27] J. K. Bradley, A. Kyrola, D. Bickson, and C. Guestrin. Parallel coordinate descent for l1-regularized loss minimization. *In the 28th International Conference on Machine Learning, July 2011, Washington, USA*, 2011.

[28] M. J. Brennan and E. S. Schwartz. Finite difference methods and jump processes arising in the pricing of contingent claims: A synthesis. *The Journal of Financial and Quantitative Analysis*, 13(3):461, sep 1978.

[29] M. Broucke, M. D. D. Benedetto, S. D. Gennaro, and A. Sangiovanni-Vincentelli. Optimal control using bisimulations: Implementation. In *Hybrid Systems: Computation and Control*, pages 175–188. Springer Berlin Heidelberg, 2001.

[30] M. Bujorianu and J. Lygeros. General stochastic hybrid systems: modelling and optimal control. In *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*. IEEE, 2004.

[31] M. Bujorianu and J. Lygeros. Toward a general theory of stochastic hybrid systems. In *Stochastic Hybrid Systems*, volume 337 of *Lecture Notes in Control and Information Science*. Springer, 2006.

[32] M. L. Bujorianu. Extended stochastic hybrid systems and their reachability problem. In *Hybrid Systems: Computation and Control, 7th International Workshop, HSCC, Philadelphia, PA, USA*, volume 2993 of *Lecture Notes in Computer Science*, pages 234–249. Springer, 2004.

[33] M. L. Bujorianu and R. Wisniewski. New insights on p-safety of stochastic systems. In *58th IEEE Conference on Decision and Control, CDC 2019, Nice, France, December 11-13, 2019*, pages 4433–4438. IEEE, 2019.

[34] B. Casella and G. O. Roberts. Exact monte carlo simulation of killed diffusions. *Advances in Applied Probability*, 40(1):273–291, mar 2008.

[35] D. Chatterjee, E. Cinquemani, and J. Lygeros. Maximizing the probability of attaining a target prior to extinction. *Nonlinear Analysis: Hybrid Systems*, 5(2): 367–381, may 2011.

[36] R.-R. Chen and T. T. Yang. A universal lattice. *Review of Derivatives Research*, 3 (2):115–133, 1999.

[37] Z. Ciesielski and S. J. Taylor. First passage times and sojourn times for Brownian motion in space and the exact Hausdorff measure of the sample path. *Transactions of the American Mathematical Society*, 103(3):434–434, mar 1962.

[38] P. Comon. Tensor decompositions, state of the art and applications. *Mathematics in Signal Processing, McWhirter and Proudler (Ed.) (2002) 1-24*, 2009.

[39] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain. Symmetric tensors and symmetric tensor rank. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1254–1279, jan 2008.

[40] F. Cosentino, H. Oberhauser, and A. Abate. A randomized algorithm to reduce the support of discrete measures. In *Advances in Neural Information Processing Systems*, 2020.

[41] F. Cosentino, H. Oberhauser, and A. Abate. Carathéodory sampling for stochastic gradient descent. *arXiv preprint arXiv:2006.01819v2*, 2020.

[42] F. Cosentino, H. Oberhauser, and A. Abate. Grid-free computation of probabilistic safety with malliavin calculus. *arXiv preprint arXiv:2104.14691*, Apr. 2021.

[43] F. Cosentino, H. Oberhauser, and A. Abate. Markov chain approximations to stochastic differential equations by recombination on lattice trees. *arXiv:2111.03497*, Nov. 2021.

[44] M. Costabile, I. Massabó, and E. Russo. A forward shooting grid method for option pricing with stochastic volatility. *The Journal of Derivatives*, 20(2):67–78, nov 2012.

[45] M. Costabile, I. Massabò, and E. Russo. A shifted tree model for the efficient evaluation of options with fixed dividends. *IMA Journal of Management Mathematics*, page dpw002, jun 2016.

[46] J. C. Cox, S. A. Ross, and M. Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7(3):229–263, sep 1979.

[47] J. C. Cox, J. E. Ingersoll, and S. A. Ross. A theory of the term structure of interest rates. *Econometrica*, 53(2):385, mar 1985.

[48] H. Cramér. On the composition of elementary errors. *Scandinavian Actuarial Journal*, 1928(1):13–74, jan 1928.

[49] D. Csiba and P. Richtárik. Global convergence of arbitrary-block gradient methods for generalized polyak-łojasiewicz functions. *arXiv preprint arXiv:1709.03014*, 2017.

[50] P. J. Davis. A construction of nonnegative approximate quadratures. *Mathematics of Computation*, 21:578–582, 1967.

[51] J. A. De Loera, R. Hemmecke, K. Matthias, et al. *Algebraic and geometric ideas in the theory of discrete optimization*, volume 14. SIAM, 2013.

[52] A. Dzougoutov, K.-S. Moon, E. von Schwerin, A. Szepessy, and R. Tempone. Adaptive monte carlo algorithms for stopped diffusion. In *Lecture Notes in Computational Science and Engineering*, pages 59–88. Springer Berlin Heidelberg, 2005.

[53] N. Ekvall. A lattice approach for pricing of multivariate contingent claims. *European Journal of Operational Research*, 91(2):214–228, jun 1996.

[54] P. M. Esfahani, D. Chatterjee, and J. Lygeros. The stochastic reach-avoid problem and set characterization for diffusions. *Automatica*, 70:43–56, aug 2016.

[55] S. N. Ethier and T. G. Kurtz. *Markov processes: characterization and convergence*, volume 282. John Wiley & Sons, 2009.

[56] D. Feldman, M. Schmidt, and C. Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering. In *SODA*, 2013.

[57] O. Fercoq and P. Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

[58] T. J. Finucane and M. J. Tomas. American stochastic volatility call option pricing: A lattice based approach. *Review of Derivatives Research*, 1(2):183–201, jun 1996.

[59] I. Florescu and F. G. Viens. Stochastic volatility: Option pricing using a multinomial recombining tree. *Applied Mathematical Finance*, 15(2):151–181, apr 2008.

[60] E. Fournié, J.-M. Lasry, J. Lebuchoux, P.-L. Lions, and N. Touzi. Applications of Malliavin calculus to Monte Carlo methods in finance. *Finance and Stochastics*, 3(4): 391–412, 1999.

[61] E. Fournié, J.-M. Lasry, J. Lebuchoux, and P.-L. Lions. Applications of Malliavin calculus to Monte-Carlo methods in finance. II. *Finance and Stochastics*, 5(2):201–236, 2001.

[62] A. Friedman. *Partial Differential Equations of Parabolic Type*. Dover Pubn Inc, 2008.

[63] N. Frikha, A. Kohatsu-Higa, and L. Li. On the first hitting times of one dimensional elliptic diffusions. *ArXiv preprint, ArXiv:1609.09327*, 2016.

[64] N. Frikha, A. Kohatsu-Higa, and L. Li. Integration by parts formula for killed processes: a point of view from approximation theory. *Electronic Journal of Probability*, 24(0), 2019.

[65] A. Gamba and L. Trigeorgis. A log-transformed binomial lattice extension for multi-dimensional option problems. In *Real Options*, 2001.

[66] A. Gamba and L. Trigeorgis. An improved binomial lattice method for multi-dimensional options. *Applied Mathematical Finance*, 14(5):453–475, dec 2007.

[67] S. Gerschgorin. Uber die abgrenzung der eigenwerte einer matrix. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na*, 6: 749–754, 1931.

[68] T. Glasmachers and U. Dogan. Accelerated coordinate descent with adaptive coordinate frequencies. In *Proceedings of the 5th Asian Conference on Machine Learning*, pages 72–86, Australian National University, Canberra, Australia, 2013. PMLR.

[69] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer New York, New York, NY, 2003.

[70] E. Gobet. Weak approximation of killed diffusion using Euler schemes. *Stochastic Processes and their Applications*, 87(2):167–197, 2000.

[71] E. Gobet and S. Menozzi. Exact approximation rate of killed hypoelliptic diffusions using the discrete euler scheme. *Stochastic Processes and their Applications*, 112(2): 201–223, aug 2004.

[72] E. Gobet and S. Menozzi. Discrete sampling of functionals of ito processes. In *Lecture Notes in Mathematics*, pages 355–374. Springer Berlin Heidelberg, 2007.

[73] E. Gobet and S. Menozzi. Stopped diffusion processes: Boundary corrections and overshoot. *Stochastic Processes and their Applications*, 120(2):130–162, feb 2010.

[74] I. Gyöngy and M. Rásonyi. A note on euler approximations for SDEs with hölder

continuous diffusion coefficients. *Stochastic Processes and their Applications*, 121(10): 2189–2200, oct 2011.

[75] T. J. Haahtela. Recombining trinomial tree for real option valuation with changing volatility. *SSRN Electronic Journal*, 2010.

[76] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, sep 1994.

[77] S. Hayakawa. Monte carlo cubature construction. *arXiv preprint arXiv:2001.00843*, 2020.

[78] P. Henry-Labordère, X. Tan, and N. Touzi. Unbiased simulation of stochastic differential equations. *The Annals of Applied Probability*, 27(6), dec 2017.

[79] S. Herrmann and C. Zucca. Exact simulation of the first-passage time of diffusions. *Journal of Scientific Computing*, 79(3):1477–1504, jan 2019.

[80] S. L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6(2):327–343, apr 1993.

[81] T.-S. Ho, R. C. Stapleton, and M. G. Subrahmanyam. Multivariate binomial approximations for asset prices with nonstationary variance and covariance characteristics. *Review of Financial Studies*, 8(4):1125–1152, oct 1995.

[82] A. J. Hoffman and H. W. Wielandt. The variation of the spectrum of a normal matrix. *Duke Mathematical Journal*, 20(1):37–39, mar 1953.

[83] J. M. Holte. Discrete gronwall lemma and applications. MAA North Central Section Meeting, 2009. `http://homepages.gac.edu/~holte/publications/GronwallLemma.pdf`.

[84] L. Hörmander. Hypoelliptic second order differential equations. *Acta Mathematica*, 119(0):147–171, 1967.

[85] J. Hu and M. Prandini. Aircraft conflict detection: A method for computing the probability of conflict based on markov chain approximation. In *7th European Control Conference, ECC, Cambridge, UK, September 1-4*, pages 2225–2230. IEEE, 2003.

[86] J. Hu, J. Lygeros, and S. Sastry. Towards a theory of stochastic hybrid systems. In *Hybrid Systems: Computation and Control, Third International Workshop, HSCC, Pittsburgh, PA, USA*, volume 1790 of *Lecture Notes in Computer Science*, pages 160–173. Springer, 2000.

[87] C.-W. Huang and T.-Y. Shih. On the complexity of point-in-polygon algorithms. *Computers & Geosciences*, 23(1):109–118, feb 1997.

[88] J. H. Huggins, T. Campbell, and T. Broderick. Coresets for scalable bayesian logistic regression. *arXiv preprint arXiv:1605.06423*, 2016.

[89] J. Hull and A. White. Valuing derivative securities using the explicit finite difference method. *The Journal of Financial and Quantitative Analysis*, 25(1):87, mar 1990.

[90] J. C. Hull. *Options, Futures, and Other Derivatives, Global Edition*. Pearson, 2017.

[91] H. T. Huynh, V. S. Lai, and I. Soumaré. Solution of stochastic differential equations. In *Stochastic Simulation and Applications in Finance with MATLAB® Programs*, pages 123–148. John Wiley & Sons, Ltd., may 2012.

[92] D. Ji and B. W. Brorsen. A recombining lattice option pricing model that relaxes the assumption of lognormality. *Review of Derivatives Research*, 14(3):349–367, dec 2010.

[93] A. Julius and G. Pappas. Approximations of stochastic hybrid systems. *IEEE Transactions on Automatic Control*, 54(6):1193–1203, jun 2009.

[94] Kaggle. House sales in king county, usa, . `https://www.kaggle.com/harlfoxem/housesalesprediction`.

[95] Kaggle. New york city taxi trip duration, . `https://www.kaggle.com/c/nyc-taxi-trip-duration/data`.

[96] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980v9*, 2014.

[97] L. C. Kinsey. *Topology of Surfaces*. Springer New York, 1993.

[98] P. E. Kloeden and E. Platen. *Numerical solution of stochastic differential equations*, volume 23 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin, 1992.

[99] A. Kohatsu-Higa, A. Lejay, and K. Yasuda. Weak rate of convergence of the euler–maruyama scheme for stochastic differential equations with non-regular drift. *Journal of Computational and Applied Mathematics*, 326:138–158, dec 2017.

[100] V. Konakov and E. Mammen. Local limit theorems for transition densities of Markov chains converging to diffusions. *Probability Theory and Related Fields*, 117(4):551–587, 2000.

[101] V. Konakov and E. Mammen. Local approximations of Markov random walks by diffusions. *Stochastic Processes and their Applications*, 96(1):73–98, 2001.

[102] V. Konakov and E. Mammen. Edgeworth type expansions for Euler schemes for stochastic differential equations. *Monte Carlo Methods and Applications*, 8(3):271–285, 2002.

[103] V. Konakov and E. Mammen. Edgeworth-type expansions for transition densities of Markov chains converging to diffusions. *Bernoulli. Official Journal of the Bernoulli Society for Mathematical Statistics and Probability*, 11(4):591–641, 2005.

[104] V. Konakov and S. Menozzi. Weak error for the euler scheme approximation of diffusions with non-smooth coefficients. *Electronic Journal of Probability*, 22(0), 2017.

[105] X. D. Koutsoukos and D. Riley. Computational methods for reachability analysis of stochastic hybrid systems. In *Hybrid Systems: Computation and Control, 9th International Workshop, HSCC, Santa Barbara, CA, USA*, volume 3927 of *Lecture Notes in Computer Science*, pages 377–391. Springer, 2006.

[106] G. N. Kumar and M. Bangi. An extension to winding number and point-in-polygon algorithm. *IFAC-PapersOnLine*, 51(1):548–553, 2018.

[107] H. Kunita. Stochastic differential equations and stochastic flows of diffeomorphisms. In *Lecture Notes in Mathematics*, pages 143–303. Springer Berlin Heidelberg, 1984.

[108] H. J. Kushner. Numerical methods for stochastic control problems in continuous time. *SIAM Journal on Control and Optimization*, 28(5):999–1048, 1990.

[109] H. J. Kushner and P. Dupuis. *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer New York, 2001.

[110] S. Kusuoka. Approximation of expectation of diffusion process and mathematical finance. In *Taniguchi Conference on Mathematics Nara '98*. Mathematical Society of Japan, 2001.

[111] M. Kwiatkowska, G. Norman, and D. Parker. Probabilistic symbolic model checking with PRISM: A hybrid approach. *International Journal on Software Tools for Technology Transfer*, 6(2):128–142, 2004.

[112] M. Kwiatkowska, G. Norman, and D. Parker. Stochastic model checking. In *Proceedings of the 7th International Conference on Formal Methods for Performance Evaluation*, SFM'07, pages 220–270, Berlin, Heidelberg, 2007. Springer-Verlag.

[113] L. Laurenti, M. Lahijanian, A. Abate, L. Cardelli, and M. Kwiatkowska. Formal and efficient synthesis for continuous-time linear stochastic hybrid processes. *IEEE Transactions on Automatic Control*, pages 1–1, 2020.

[114] A. Leccadito, P. Toscano, and R. S. Tunaru. Hermite binomail trees: a novel technique for derivatives pricing. *International Journal of Theoretical and Applied Finance*, 15 (08):1250058, dec 2012.

[115] C.-P. Lee and S. J. Wright. Random permutations fix a worst case for cyclic coordinate descent. *IMA Journal of Numerical Analysis*, 39(3), 2018.

[116] D. P. Leisen. A stochastic volatility lattice. *IFAC Proceedings Volumes*, 31(16):75–80, jun 1998.

[117] A. Li, P. Ritchken, and L. Sankarasubramanian. Lattice models for pricing american interest rate claims. *The Journal of Finance*, 50(2):719–737, jun 1995.

[118] X. Li, T. Zhao, R. Arora, H. Liu, and M. Hong. On faster convergence of cyclic block coordinate descent-type methods for strongly convex minimization. *Journal Of Machine Learning Research*, 18, 2018.

[119] C. Litterer and T. Lyons. High order recombination and an application to cubature on Wiener space. *The Annals of Applied Probability*, 22(4):1301–1327, 2012.

[120] R. Liu and J. Zhao. A lattice method for option pricing with two underlying assets in the regime-switching model. *Journal of Computational and Applied Mathematics*, 250:96–106, oct 2013.

[121] R. H. Liu. Regime-switching recombining tree for option pricing. *International Journal of Theoretical and Applied Finance*, 13(03):479–499, may 2010.

[122] U. H. Lok and Y.-D. Lyuu. Efficient trinomial trees for local-volatility models in pricing double-barrier options. *Journal of Futures Markets*, 40(4):556–574, dec 2019.

[123] M. Luby, A. Sinclair, and D. Zuckerman. Optimal speedup of Las Vegas algorithms. *Information Processing Letters*, 47(4):173–180, 1993.

[124] K. Lundengård, C. Ogutu, S. Silvestrov, Y. Ni, and P. Weke. Construction of moment-matching multinomial lattices using vandermonde matrices and gröbner bases. In *ICNPAA 2016 World Congress: 11th International Conference on Mathematical Problems in Engineering, Aerospace and Sciences*. Author(s), 2017.

[125] T. Lyons and N. Victoir. Cubature on Wiener space. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 460(2041):169–198, jan 2004.

[126] A. Maalouf, I. Jubran, and D. Feldman. Fast and accurate least-mean-squares solvers. In *Advances in Neural Information Processing Systems*, pages 8305–8316, 2019.

[127] P. Malliavin and A. Thalmaier. *Stochastic calculus of variations in mathematical finance*. Springer Finance. Springer-Verlag, Berlin, 2006.

[128] L. Meier, S. van de Geer, and P. Buhlmann. The group lasso for logistic regression. *Journal Of The Royal Statistical Society Series B-Statistical Methodology*, 70:53–71, 2008.

[129] G. N. Milstein. Weak approximation of solutions of systems of stochastic differential equations. In *Numerical Integration of Stochastic Differential Equations*, pages 101–134. Springer Netherlands, 1995.

[130] I. M. Mitchell. A toolbox of level set methods version 1.0, 2004.

[131] P. Mohajerin Esfahani, D. Chatterjee, and J. Lygeros. The stochastic reach-avoid problem and set characterization for diffusions. *Automatica*, 70:43–56, 2016.

[132] E. Moretto, S. Pasquali, and B. Trivellato. Derivative evaluation using recombining trees under stochastic volatility. *Advances and Applications in Statistical Science*, 1: 453–480, Aug. 2009.

[133] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[134] S. W. N. Ikeda. *Stochastic Differential Equations and Diffusion Processes*. Elsevier S&T, 2014.

[135] S. K. Nawalkha and N. Beliaeva. Efficient trees for CIR and CEV short rate models. *SSRN Electronic Journal*, 2007.

[136] D. B. Nelson and K. Ramaswamy. Simple binomial processes as diffusion approximations in financial models. *Review of Financial Studies*, 3(3):393–430, jul 1990.

[137] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[138] Y. Nesterov. *Lectures on convex optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer, Cham, 2018. Second edition of [MR2142598].

[139] Y. E. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Doklady Akademii Nauk SSSR*, 269(3):543–547, 1983.

[140] M. Ninomiya and S. Ninomiya. A new higher-order weak approximation scheme for stochastic differential equations and the runge–kutta method. *Finance and Stochastics*, 13(3):415–443, may 2009.

[141] J. Nutini, M. Schmidt, I. H. Laradji, M. P. Friedlander, and H. A. Koepke. Coordinate descent converges faster with the gauss-southwell rule than random selection. In *Proceedings of the 32nd International Conference on Machine Learning, ICML, Lille, France*, pages 1632–1641, 2015.

[142] J. Nutini, I. Laradji, and M. Schmidt. Let's make block coordinate descent go fast: Faster greedy rules, message-passing, active-set complexity, and superlinear convergence. *arXiv preprint arXiv:1712.08859*, 2017.

[143] S. Osher. A level set formulation for the solution of the Dirichlet problem for Hamilton-Jacobi equations. *SIAM Journal on Mathematical Analysis*, 24(5):1145–1152, sep 1993.

[144] J. M. Phillips. Coresets and sketches. *arXiv preprint arXiv:1601.00617*, 2016.

[145] F. Piazzon, A. Sommariva, and M. Vianello. Caratheodory-tchakaloff subsampling. *Dolomites Research Notes on Approximation*, 10(1), 2017.

[146] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research, April 1998.

[147] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, jul 1992.

[148] S. Prajna, A. Jadbabaie, and G. J. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8):1415–1428, aug 2007.

[149] M. Prandini, J. Hu, J. Lygeros, and S. Sastry. A probabilistic approach to aircraft conflict detection. *IEEE Trans. Intell. Transp. Syst.*, 1(4):199–220, 2000.

[150] J. A. Primbs, M. Rathinam, and Y. Yamada. Option pricing with a pentanomial lattice model that incorporates skewness and kurtosis. *Applied Mathematical Finance*, 14(1):1–17, feb 2007.

[151] Z. Qu and P. Richtárik. Coordinate descent with arbitrary sampling i: algorithms and complexity. *Optimization Methods and Software*, 31(5):829–857, jul 2016.

[152] R. R. Rao and R. N. Bhattacharya. *Normal approximation and asymptotic expansions*. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104, 2010.

[153] R. G. Regis. On the properties of positive spanning sets and positive bases. *Optimization and Engineering. International Multidisciplinary Journal to Promote Optimization Theory & Applications in Engineering Sciences*, 17(1):229–262, 2016.

[154] P. Richtárik and M. Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1-2):433–484, 2016.

[155] D. Riley, X. D. Koutsoukos, and K. Riley. Reachability analysis for stochastic hybrid systems using multilevel splitting. In *Hybrid Systems: Computation and Control, 12th International Conference, HSCC, San Francisco, CA, USA*, volume 5469 of *Lecture Notes in Computer Science*, pages 460–464. Springer, 2009.

[156] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, sep 1951.

[157] M. Romano and N. Touzi. Contingent claims and market completeness in a stochastic volatility model. *Mathematical Finance*, 7(4):399–412, oct 1997.

[158] N. L. Roux, M. Schmidt, and F. R. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in neural information processing systems*, pages 2663–2671, 2012.

[159] M. Rubinstein. Implied binomial trees. *The Journal of Finance*, 49(3):771–818, jul 1994.

[160] M. Rubinstein. Edgeworth binomial trees. *The Journal of Derivatives*, 5(3):20–27, feb 1998.

[161] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[162] M. D. Santis, S. Lucidi, and F. Rinaldi. A fast active set block coordinate descent algorithm for l1-regularized least squares. *SIAM Journal on Optimization*, 26(1): 781–809, jan 2016.

[163] C. Scherrer, A. Tewari, M. Halappanavar, and D. Haglin. Feature clustering for accelerating parallel coordinate descent. In *Advances in Neural Information Processing Systems*, volume 25, pages 28–36. Curran Associates, Inc., 2012.

[164] R. Schneider and W. Weil. *Stochastic and integral geometry*. Probability and its Applications (New York). Springer-Verlag, Berlin, 2008.

[165] F. Shmarov and P. Zuliani. Probabilistic hybrid systems verification via smt and monte carlo techniques. In *12th Haifa Verification Conference HVC*, volume 10028 of *Lecture Notes in Computer Science*, pages 152–168. Springer, 2006.

[166] D. Siegmund and Y. S. Yuh. Brownian approximations to first passage probabilities. *Zeitschrift fur Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 59(2):239–248, 1982.

[167] J.-G. Simonato. Johnson binomial trees. *SSRN Electronic Journal*, 2008.

[168] D. M. Y. Sommerville. The relations connecting the angle-sums and volume of a polytope in space of n dimensions. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 115(770):103–119, 1927.

[169] S. E. Z. Soudjani and A. Abate. Adaptive and sequential gridding procedures for the abstraction and verification of stochastic processes. *SIAM Journal on Applied Dynamical Systems*, 12(2):921–956, jan 2013.

[170] S. E. Z. Soudjani, C. Gevaerts, and A. Abate. FAUST^2: Formal abstractions of uncountable-STate STochastic processes. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 272–286. Springer Berlin Heidelberg, 2015.

[171] D. W. Stroock and S. R. S. Varadhan. *Multidimensional Diffusion Processes*. Springer Berlin Heidelberg, 2006.

[172] K. G. Suffern. Quadtree algorithms for contouring functions of two variables. *The Computer Journal*, 33(5):402–407, may 1990.

[173] S. Summers and J. Lygeros. A probabilistic reach-avoid problem for controlled discrete time stochastic hybrid systems. *IFAC Proceedings Volumes*, 42(17):150–155, 2009.

[174] D. Talay. Discrétisation d'une équation différentielle stochastique et calcul approché d'espérances de fonctionnelles de la solution. *RAIRO Modélisation Mathématique et Analyse Numérique*, 20(1):141–179, 1986.

[175] D. Talay and L. Tubaro. Expansion of the global error for numerical schemes solving stochastic differential equations. *Stochastic Analysis and Applications*, 8(4):483–509, jan 1990.

[176] V. Tchakaloff. Formules de cubature mécanique à coefficients non négatifs. *Bulletin des Sciences Mathématiques*, 81:123–134, 1957.

[177] M. Tchernychova. *Caratheodory cubature measures*. PhD thesis, University of Oxford, 2016.

[178] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, jan 1996.

[179] C. Tomlin, J. Lygeros, and S. Sastry. Computing controllers for nonlinear hybrid systems. In *Hybrid Systems: Computation and Control*, pages 238–255. Springer Berlin Heidelberg, 1999.

[180] C. Tomlin, I. Mitchell, A. Bayen, and M. Oishi. Computational techniques for the verification of hybrid systems. *Proceedings of the IEEE*, 91(7):986–1001, jul 2003.

[181] M. Wahab and C.-G. Lee. A lattice approach to pricing of multivariate contingent claims with regime switching. *The Journal of Derivatives*, 17(1):49–61, aug 2009.

[182] J. G. Wendel. A problem in geometric probability. *Math. Scand*, 11:109–111, 1962.

[183] R. Wisniewski and M. L. Bujorianu. Stochastic safety analysis of stochastic hybrid systems. In *56th IEEE Annual Conference on Decision and Control, CDC 2017, Melbourne, Australia, December 12-15, 2017*, pages 2390–2395. IEEE, 2017.

[184] R. Wisniewski and M. L. Bujorianu. Stochastic safety analysis of stochastic hybrid systems. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, dec 2017.

[185] R. Wisniewski and C. Sloth. Safety analysis of stochastic dynamical systems. In *IFAC Conference on Analysis and Design of Hybrid Systems, ADHS*. IFAC, 2015.

[186] R. Wisniewski, M. Svenstrup, A. Pedersen, and C. Steiniche. Certificate for safe emergency shutdown of wind turbines. In *Proceedings of American Control Conference*, 2013.

[187] R. Wisniewski, M. L. Bujorianu, and C. Sloth. p-safe analysis of stochastic hybrid processes. *IEEE Trans. Autom. Control.*, 65(12):5220–5235, 2020.

[188] R. Wisniewski, M. L. Bujorianu, and C. Sloth. p-safe analysis of stochastic hybrid processes. *IEEE Transactions on Automatic Control*, pages 1–1, 2020.

[189] S. J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1, Ser. B):3–34, 2015.

[190] R. Xu and S. Li. A tree model for pricing credit default swaps with equity, market and default risk. In *2009 International Conference on Management and Service Science*. IEEE, sep 2009.

[191] Y. Xu and W. Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences*, 6(3):1758–1789, jan 2013.

[192] M. Zamani, P. M. Esfahani, R. Majumdar, A. Abate, and J. Lygeros. Symbolic control of stochastic systems via approximately bisimilar finite abstractions. *IEEE Transactions on Automatic Control*, 59(12):3135–3150, dec 2014.

[193] X.-C. Zeng and S.-P. Zhu. A new simple tree approach for the heston's stochastic volatility model. *Computers & Mathematics with Applications*, 78(6):1993–2010, sep 2019.

[194] G. M. Ziegler. *Lectures on polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995.