Grid-Free Computation of Probabilistic Safety with Malliavin Calculus

Francesco Cosentino, Harald Oberhauser, Alessandro Abate

***Abstract*— This work concerns continuous-time, continuous-space stochastic dynamical systems described by stochastic differential equations (SDE). It presents a new approach to compute probabilistic safety regions, namely sets of initial conditions of the SDE associated to trajectories that are safe with a probability larger than a given threshold. The approach introduces a functional that is minimised at the border of the probabilistic safety region, then solves an optimisation problem using techniques from Malliavin Calculus, which computes such region. Unlike existing results in the literature, the new approach allows one to compute probabilistic safety regions without gridding the state space of the SDE.**

***Index Terms*— Formal Verification, Malliavin Calculus, Stochastic Differential Equations, Probabilistic Safety**

I. Background

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 [1]–[3]. Dually, one can express a reachability analysis problem by identifying states that are associated with trajectories entering a given target set. In the context of probabilistic models, such as stochastic differential equations (SDE), we are interested in characterising and computing the likelihood with which trajectories of the stochastic process either stay within a given set, or dually reach a target set - the former has been often studied in probability theory as the *exit time* problem. Alternatively, for stochastic models we might be interested in computing the set of initial conditions associated with dynamics that are safe with a probability at least equal to, say p - this is also known as p-safe analysis or computation of the p-safe region [4].

In this work, we present a new application of Malliavin Calculus [5] to the computation of the p-safe region borrowing ideas from Mathematical Finance: in particular, we leverage and tailor techniques for the computation of the (so called) *Greeks of a derivative* [6] for our goal. This allows one to compute probabilistic safety regions without gridding the state space of the SDE: grid-based techniques are by-and-large the standard approach in existing literature, with known limits related to precision and computational scalability.

Francesco Cosentino and Harald Oberhauser are with the University of Oxford, Mathematical Institute, and The Alan Turing Institute, UK (e-mail: name dot surname at maths dot ox dot ac dot uk)

Alessandro Abate is with the University of Oxford, Computer Science Department, and The Alan Turing Institute, UK (e-mail: name dot surname at cs dot ox dot ac dot uk)

II. Related literature on Probabilistic Safety

Safety analysis, a standard specification in Formal Verification, has been studied on dynamical models within the Hybrid Systems community [3]. Corresponding safety notions for stochastic models (and in particular for stochastic and hybrid ones - the latter feature is not under study in this work) have been explicitly introduced only over the past two decades [2], as further surveyed next.

This work, unlike [2], focuses on continuous-time models: particularly relevant for this setup, [7] has presented a new modeling framework named *stochastic hybrid system* (SHS), 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 (SDE), rather than deterministic ODEs. [7] proposes the notion of embedded Markov chain (EMC) and studies the exit probability problem, which is related to reachability analysis: it is shown that this probability over the EMC converges to its counterpart for the original SHS, as the discretisation step goes to zero. [8] blends the models from [7] with Markov models with jumps in [9], setting up *Markov strings* and thus obtaining a very general class of models for SHS. Closely related to [8], [10] introduces a general model for SHS and proposes a method based on Dirichlet forms, to study the reachability problem over SHS models. Similarly over SHS, [11] proposes a method to compute probabilistic reachability: underpinned by seminal work in [12], [11] first shows that reachability can be characterised as a viscosity solution of a system of coupled Hamilton-Jacobi-Bellman equations; second, it presents 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 ever finer. [13] builds on [11] by employing Monte Carlo (MC) techniques for estimating probabilities of events, and [13] uses multilevel splitting (MLS), a variance-reduction technique that can improve both efficiency and accuracy. Again over SHS, [14] establishes 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, namely the exit-time problem mentioned above, [14]

provides its characterisation as a solution of a partial differential equation in the sense of viscosity solutions, along with Dirichlet boundary conditions. [4] establishes an optimisation scheme for computing probabilistic safety of SHS, combining the use of barrier certificates and of potential theory.

[15] presents a method to compute *protection certificates*, which are closely related to the concept of p-safe region, elaborated later. As discussed in Remark 1, [16], [17] compute the p-safe region based on the extended generator of stochastic dynamical systems; these contributions characterize the safety problem as an optimization problem on the space of positive measures and then solve it via a moment-based method. [18] characterizes the p-safe regions using concepts from Potential Theory.

Alternative techniques leveraging randomised approaches have been presented: [19] introduces a method for estimating the probability of conflict for two-aircraft encounters at a fixed altitude - a probabilistic safety problem. 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, [20] discusses the maximum instantaneous probability of conflict: randomised algorithms are introduced to efficiently estimate this measure of criticality and to provide quantitative bounds on the level of the 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 (SMC) techniques, which have also been developed for models related to SHS in [21].

Finally, the work in [22] enables sound verification and correct-by-construction controller synthesis for stochastic models and their hybrid extensions [23]: a stochastic control model satisfying a probabilistic variant of *incremental input-to-state stability* is shown to be abstracted into a finite-state transition system, which is epsilon-approximately bisimilar to the original model.

III. Problem Statement

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

$$dX_t = \mu(X_t)dt + \sum_{k=1}^{d} \sigma_k(X_t)dW_t^k, \qquad X_0 = x. \quad (1)$$

The setup above is adopted by related literature, as surveyed above.

We introduce the following requirements, which are used in [5] and in particular are sufficient for all the results and algorithms proposed in this work. Obtaining weaker requirements, and thus generalising our setup, would require modifying the technical results from Malliavin calculus, which is not core to our contributions.

Assumption 1. *We suppose that the vector fields $\mu, \sigma_k \in C^\infty_{l,b}(\mathbb{R}^d; \mathbb{R}^d), k = 1, 2, \dots, d$, where $C^\infty_{l,b}(\mathbb{R}^d; \mathbb{R}^d)$ indicates the space of infinitely differentiable functions with bounded derivatives and bounded linear growth from $\mathbb{R}^d$ to $\mathbb{R}^d$. Moreover, if we call $\sigma \in C^\infty_{l,b}(\mathbb{R}^d; \mathbb{R}^{d\times d})$ the matrix whose columns are the vectors $\sigma_k, k \geq 1$, we assume that σ satisfies the uniform ellipticity condition, i.e. $\sigma\sigma^\top$ is uniformly positive definite.*

If Assumption 1 holds, then it is well-known that the SDE (1) has a unique strong solution X_t^x [24], 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 call τ_A^x the exit time of X_t^x from the region A, i.e.

$$\tau_A^x = \inf\{t \geq 0 \,:\, X_t^x \notin A\}, \qquad x \in A.$$

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_t starts from x, then it stays in A longer than T with probability greater than p, i.e.

$$\mathbb{A}_T^p = \{x \in A : \mathbb{P}(\tau^x \geq T) \geq p\}.$$

Again, whenever clear from the context we shall omit the indexes p, T.

Remark 1. *In [16] the authors study a more general problem, namely the probabilistic reach-avoid problem, defined next. 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 $p = 1 - q$. An analysis of the approximation error is not presented and, since the approach is radically different from the one presented in this paper (cf. discussion in Related Literature and in the next section), a quantitative comparison between the two approaches is questionable.*

The standard way to compute $\mathbb{A}_T^p$ is to discretise the region A and to compute the value $\mathbb{P}(\tau^x \geq T)$ at any point in the introduced grid (cf. Related Literature). 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]

$$\partial\mathbb{A}_T^p = \arg\min_x \frac{1}{2}\left(\mathbb{P}\left(\tau^x \geq T\right) - p\right)^2. \quad (2)$$

The main idea of this approach is thus to solve such optimisation problem: indeed, assuming differentiability and excluding convexity issues, we know that, setting up the recursion

$$x_{j+1} = x_j - \lambda\left(\mathbb{P}\left(\tau^x \geq T\right) - p\right) D_x\mathbb{P}\left(\tau^x \geq T\right), \quad (3)$$

[1] In [16] the authors compute $\mathbb{P}(\tau_{U^C}^x < \tau_S^x, \tau_{U^C}^x < T) \leq q$, whereas here we use $\leq$, as it does not change the outcome, whilst greatly simplifying the comparison between [16] and this work.

[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.

then $x_j \to x^\star$, where $x^\star \in \partial\mathbb{A}_T^p$, for $\lambda > 0$ small enough. Equation (3) represents a standard Gradient Descent (GD) step. We remark that in principle other optimization algorithms can be used to solve the problem in (2); in this work 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 [25], a state-of-the-art optimization procedure.

Using the GD in Equation (3) not only allows identifying the set $\mathbb{A}_T^p$: in Section VII we also provide a procedure to explore its border. Furthermore, in Theorem 2 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 1 and Corollary 1 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.

IV. Grid-based vs -free approaches

Evidently, the GD step in Equation (3) 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) with a grid-free method, 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 zero-measure set - then it will be found. Instead, grid-based approaches (broadly all those presented in the previous section on related work) will find the set $\mathbb{A}_T^p$ only if it intersects with the introduced grid. As an extreme instance, if the p-safe region consists of only one point, the procedure introduced here shall find it, up to a numerical precision related to the approximation of $\mathbb{P}(\tau^x \geq T)$ and $D_x\mathbb{P}(\tau^x \geq T)$; on the contrary, this might not be possible for grid-based approaches, unless the grid is selected to intersect such point (which is usually not known beforehand) - and this is a limit holding regardless of their numerical implementation. 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 uncountably infinite set, grid-based procedures search for solutions within a pre-defined, possibly 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 work 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 1 and Algorithm 1. The points explored by the two methods can be quantified as $O(\mathrm{Area}(A)\gamma^{-2})$ for the grid discretisation - this is the number of points of $\gamma\mathbb{Z}^2$ in the overall (larger) region A - and $O(\mathrm{Len}(\partial\mathbb{A}_T^p)\gamma^{-1})+C$ 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 $O(\mathrm{Area}(A)\gamma^{-d})$ and $O(\mathrm{Area}(\partial\mathbb{A}_T^p)\gamma^{-d+1}) + C$.[3] As an estimate, we can see that, as $\gamma \to 0$, the order of points explored is much less with the method presented here.

V. Malliavin Calculus for stopping times

In Pricing Theory, a branch of Mathematical Finance, a classical problem is to evaluate the variation of the price of a derivative, in response to a change of the underlying asset price or volatility [6]. These quantities are known as *Greeks* and 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 we call p_0 the initial price of the underlying asset, the *Greek* representing the sensitivity with respect to the initial price is called Δ, and is formally defined as

$$\Delta = \frac{\partial D_{p_0}}{\partial p_0} = \frac{\partial \mathbb{E}[f(P_T^{p_0})]}{\partial p_0}.$$

Through Malliavin Calculus it is possible to provide explicit formulae for the *Greeks* [5], [26]–[30]. We refer to [31] for a computational perspective on these methods.

In our problem setup, we are interested to compute the quantity $D_x\mathbb{P}(\tau^x \geq T)$ used in (3), 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 indicator functional of the exit time. We should otherwise estimate this quantity numerically, with associated unavoidable imprecision. Under Assumption 1, it is possible to show that X_t^x is a.s. differentiable with respect to the starting point x [32]. Following the notations in [5], 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

$$dJ_t = \boldsymbol{\mu}(X_t)J_t dt + \sum_{k=1}^{d} \boldsymbol{\sigma}_k(X_t)J_t dW_t^k, \qquad J_0 = I_d.$$

The main results we leverage is the following.

Theorem 1. *[5, Theorem 2.18] If Assumption 1 holds true, calling τ^1 the time when $\int_0^{\tau^1} dist(X_t, \partial A)^{-2}dt = 1$, then*

$$\frac{\partial}{\partial\varepsilon}\mathbb{P}\left[\tau^{x+\varepsilon\varsigma} \geq T\right]\Big|_{\varepsilon=0} = \mathbb{E}\left[\mathbb{1}_{\{\tau^x \geq T\}} H_{\varsigma,T}\right],$$

$$H_{\varsigma,T} = \sum_{k=1}^{d} \int_0^T \beta_t^k \frac{\mathbb{1}_{\{t<\tau^1\}}}{dist(X_t, \partial A)^2} dW_t^k,$$

[3] This is a slight abuse of notation, indeed $\mathrm{Area}(\partial\mathbb{A}_T^p)$ now represents the Lebesgue measure in $\mathbb{R}^{d-1}$, whist $\mathrm{Area}(A)$ represents the Lebesgue measure in $\mathbb{R}^d$.

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 $\varsigma = e_i, i = 1, \dots, d$, we can obtain the gradient $D_x \mathbb{P}(\tau^x \geq T)$, which lies at the core of our procedure in Equation (3): without this result, this derivative should be estimated alternatively, for instance numerically. 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

$$H_T = \int_0^T \frac{\mathbb{1}_{\{t<\tau^1\}}}{\text{dist}(X_t, \partial A)^2} \beta_t \cdot dW_t, \qquad (4)$$
$$\beta_t = \sigma^{-1}(X_t) \cdot J_t,$$

and σ is the matrix whose columns are the vectors σ_k. Please note that the dimension of β_t, H_t in Equation (4) and Theorem 1 are different.

VI. Properties of the region $\mathbb{A}_T^p$

Whilst Theorem 1 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 (3) $(\mathbb{P}(\tau^x \geq T) - p) D_x \mathbb{P}(\tau^x \geq T) = 0$, regardless of the value of the gradient $D_x \mathbb{P}(\tau^x \geq T)$. Therefore, if we end at a point x where

$$\mathbb{P}(\tau^x \geq T) \neq p \text{ and } (\mathbb{P}(\tau^x \geq T) - p) D_x \mathbb{P}(\tau^x \geq T) = 0,$$

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 (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 countably infinite union of sets, wherein any $\mathbb{A}_i$ is a bounded connected set. Each component $\mathbb{A}_i$ is endowed with interesting properties.

Definition 1. *We say that a surface (see [33] for a formal definition) is closed if it partitions the space, e.g.* $\mathbb{R}^d$*, into one bounded connected region and one unbounded region. We denote this bounded region as the* interior *of the surface.*

Theorem 2 (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

$$\begin{aligned}\mathbb{P}(\tau^x \geq T) =& \mathbb{P}\left(\tau^{X_\theta^x} \geq T - \theta^x \middle| \theta^x \leq T\right) \mathbb{P}(\theta^x \leq T) + \\ &+ \mathbb{P}\left(\tau^x \geq T \middle| \theta^x \geq T\right) \mathbb{P}(\theta^x \geq T) \\ =& \mathbb{P}\left(\tau^{X_\theta^x} \geq T - \theta^x \middle| \theta^x \leq T\right) \mathbb{P}(\theta^x \leq T) + \\ &+ \mathbb{P}(\theta^x \geq T),\end{aligned}$$

where thanks to the definition of θ^x, τ^x we have that $\mathbb{P}(\tau^x \geq T \,|\, \theta^x \geq T) = 1$, 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

$$\begin{aligned}\mathbb{P}(\tau^x \geq T) \geq& \mathbb{P}\left(\tau^{X_\theta^x} \geq T \middle| \theta^x \leq T\right) \mathbb{P}(\theta^x \leq T) \\ &+ \mathbb{P}(\theta^x \geq T) \\ \geq& p\mathbb{P}(\theta^x \leq T) + p\mathbb{P}(\theta^x \geq T) \\ =& p,\end{aligned}$$

because $X_\theta^x \in \partial \mathbb{A}_T^p$. □

From an algorithmic point of view, Theorem 2 is remarkable: once the algorithm has obtained a *closed* surface for $\{x : \mathbb{P}(\tau^x \geq t) = p\}$ 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 bounded 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 [34]–[37], 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 (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 1. *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 2 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 1 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 1. *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 1 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, if x is outside $\mathbb{A}_i$, but $S \cap \mathbb{A}_j \neq \emptyset$ for some $j \neq i$. □

VII. 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 in detail the case of models with dimension $d = 2, 3$; if the model dimension is greater than 3, we show that the procedure can be iterated.

Dimension 2

Algorithm 1 explains how to proceed if $d = 2$. Let us discuss the main steps of Algorithm 1:

(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 2 we know that we can stop when we have found a 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

$$\begin{aligned}\mathbb{P}\left(\tau^{x_0} \geq t\right) \approx &\mathbb{P}\left(\tau^x \geq t\right) + D_x\mathbb{P}\left(\tau^x \geq t\right)(\gamma \times dir) + \\ &+ \|\gamma \times dir\|^2 \mathrm{ERR} \\ \approx &p + \|\gamma \times dir\|^2 \mathrm{ERR},\end{aligned}$$

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 VIII.

Algorithm 1 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 $\mathbb{P}\left(\tau^{x_\star} \geq T\right) \approx p$
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 $D_x\mathbb{P}\left(\tau^x \geq t\right)$
8: $x \leftarrow x + \gamma \times \frac{dir}{\|dir\|}$
9: step $\leftarrow$ step $+1$
10: **end while**

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 1 constrained on any given plane $x_1 = \pm i\delta$ that we are considering, see Figure 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 $\min_{x\in\phi} \mathbb{P}\left(\tau^x \geq T\right) < p$, 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).

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 that are relative to some sections, i.e. to generalize directly from the case $d=2$. Let us suppose that $x_\star$ is a point on the border; then we can compute $d-1$ orthonormal vectors $\{g_1^\star,\dots,g_{d-1}^\star\}$ to $D_x\mathbb{P}(\tau^x \geq T)|_{x=x_\star}$, thus running the procedure recursively from any new point $x_\star \pm \gamma g_j^\star, j=1,\dots,d-1$, until we obtain a closed surface. However, attention is needed with the selection of the orthonormal points $\{g_1^\star,\dots,g_{d-1}^\star\}$: 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 - see the discussion relative to Line 5 of Algorithm 1 above.

VIII. Experiments

In this section, we present a case study: the code can be found at `https://github.com/FraCose/Grid-free_prob_safety`.
For the experiments, we use a simulation-based approach, i.e. we use Monte Carlo (MC) techniques, and to reduce the variance we use antithetic Brownian paths [38], [39].

Remark 2. *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 work. 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 [40]–[42] 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 model for the case study, it is important to draw some general considerations on the discussed technique.

Complexity

Let us recall the definition of H_t and β_t:

$$H_T = \int_0^T \frac{\mathbb{1}_{\{t<\tau^1\}}}{\text{dist}(X_t,\partial A)^2}\beta_t \cdot dW_t,$$
$$\beta_t = \sigma^{-1}(t,X_t)\cdot J_t.$$

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 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 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 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 work. 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 [43] 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 (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 [43]

$$\left|\mathbb{P}(\tau^x \geq T) - \widehat{\mathbb{P}(\tau_n^x \geq T)}\right| \leq O(n^{-1/2}) + O\left(\frac{1}{\sqrt{N}}\right)Z,$$

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 $\left|D_x\mathbb{P}(\tau^x \geq T) - \widehat{D_x\mathbb{P}(\tau_n^x \geq T)}\right|$ [43]–[45], though an adaptation is needed due to the presence of τ_1 in the definition of H in Theorem 1. 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 $D_x\mathbb{P}(\tau^x \geq T)$ 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 (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 [25]). Instead, if we are considering the minimization procedure in Algorithm 1, since we should be already close to the border we should select a small λ.

(iv) γ, the "border exploration" parameter in Algorithm 1 - 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 exploration of the border $\partial\mathbb{A}_T^p$ will be quite slow.

Acceleration of the exploration

Algorithm 1 is a good starting point to explore 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.

Firstly, we would like to explore 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 already points 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 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 [46].

Secondly, going towards the direction perpendicular to the gradient, see Section VII, 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 Figure 1 and 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 exploration, since it reduces the approximation error relative to the gradient.

Finally, it is better to constraint the space where the algorithm searches for the next point of the border. In Figure 1 it is shown

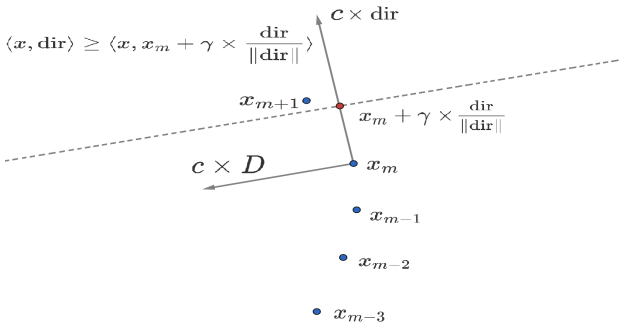

Fig. 1: Representation of how the algorithm explores the border: given the points $\{x_m, x_{m-1}, \ldots\}$ already found by the algorithm, it proposes the red point $x_m + \gamma \times \mathrm{dir}/\|\mathrm{dir}\|$ as the new point of the region, and from there it runs the GD to find the new point 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 2.

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[4] (dir). Given dir and γ, it is possible to search the new point only in the part of plane where there are not "recent points" considering the line perpendicular to the direction passing through the point guess $x_m + \gamma \times \mathrm{dir}/\|\mathrm{dir}\|$. 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 2 we present a pseudo-code of the procedure.

Algorithm 2 Adaptive constraint for the GD procedure

```
Given x_m
Run the constrained GD from x_m + γ dir/‖dir‖ up to a point x_⋆
step ← 1
while P(τ^{x_⋆} ≥ T) ≉ p do
    dir ← 2x_⋆ − x_m
    x̄ ← x_m + γ/(2·step) · dir/‖dir‖
    Update the plane using the new dir and x̄
    Run the constrained GD from x̄ up to a point x_⋆
    step ← step + 1
end while
```

If the direction guides towards points already explored

[4]Possibly as a weighted average of the gradient and some local approximation of the curvature as explained before.

recently, because for instance the discretization error is too high or due to the constrained updates in Algorithm 2, given that on a plane the perpendicular vectors to a vector are two, it is enough to invert the direction.

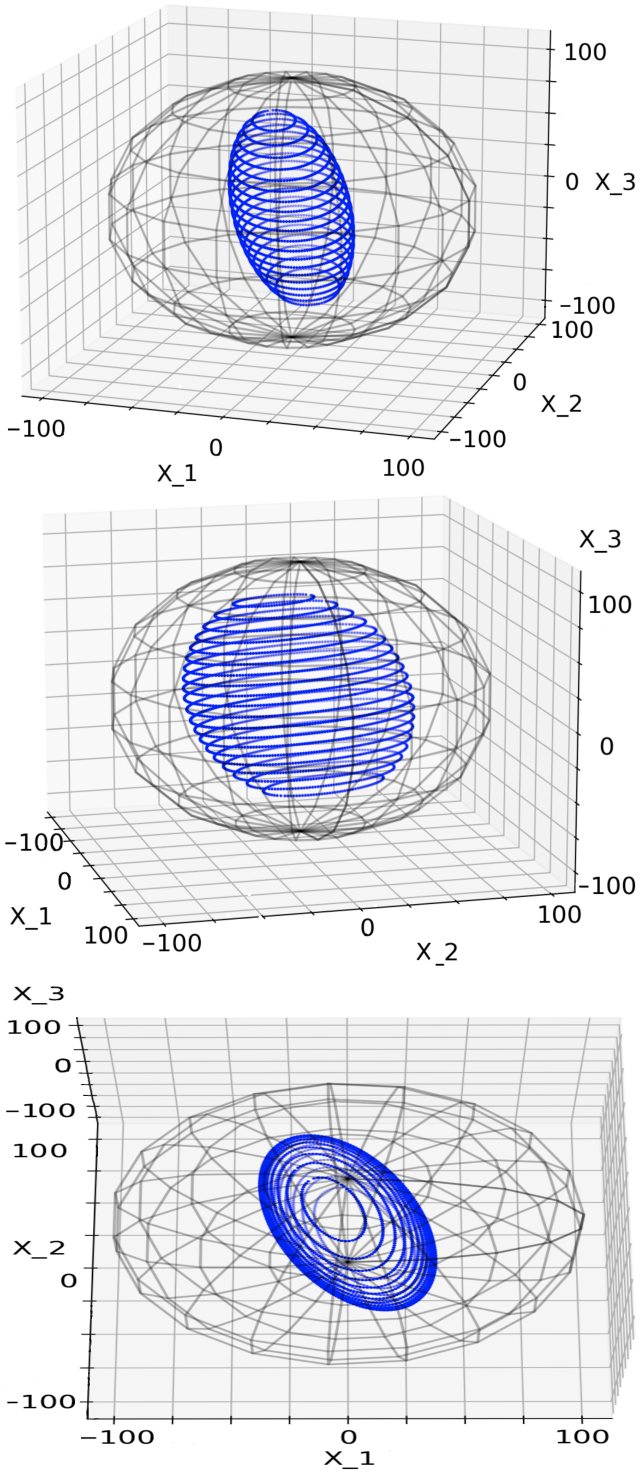

Fig. 2: The plots show the surface $\partial\mathbb{A}_1^{0.5}$ (in blue) found when the region A is a (black) sphere. This has been computed sectioning the region A across 2D planes (we have discussed at the end of Section VII the use of alternative grid-free approaches for exploration in 3D (and higher-dimensional) cases).

Case study

The model considered for the experiment is:

$$\begin{pmatrix} dX_t^{(1)} \\ dX_t^{(2)} \\ dX_t^{(3)} \end{pmatrix} = \begin{pmatrix} X_t^{(1)} \\ \frac{1}{2}X_t^{(1)} + \frac{1}{2}X_t^{(2)} \\ \frac{1}{3}X_t^{(1)} + \frac{1}{3}X_t^{(2)} + \frac{1}{3}X_t^{(3)} \end{pmatrix} dt + \\ + \frac{1}{3}\begin{pmatrix} \omega_1, & \omega_2, & \omega_2 \\ \omega_2, & \omega_1, & \omega_2 \\ \omega_2, & \omega_2, & \omega_1 \end{pmatrix} dW_t, \tag{5}$$

$$\omega_1 := 2\sqrt{1-\rho} + \sqrt{1+2\rho},$$
$$\omega_2 := -\sqrt{1-\rho} + \sqrt{1+2\rho}.$$

If we define $d\tilde{W}_t = \sigma dW_t$, where σ is the diffusion matrix in Equation (5), then we have that $\mathrm{Corr}(d\tilde{W}_t^{(i)}, d\tilde{W}_t^{(j)}) = \rho$, $i \neq j$ and $i, j \in \{1, 2, 3\}$. In the experiment we have used $\rho = 0.5$. For the region A, we have considered two caseual 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 work are not satisfied. Nevertheless, the procedure is still able to explore 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. $|\mathbb{P}(\tau^{x_\star} \geq 1) - 0.5| < \mathrm{err}$ – 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 1, i.e. we fix the plane $x^{(3)} = x_\star^{(3)}$, see Figures 3, 4 for the results of the experiments. See Figure 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 [25], a version of GD with momentum and adaptive learning rate that has shown impressive performance in Machine Learning and it is considered the state-of-the-art optimisation tool. 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 as a function of x.

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 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 the reader to the code at `https://github.com/FraCose/Grid-free_prob_safety`.

IX. Conclusions

We have presented a new approach to find and compute probabilistic safety regions for stochastic differential equations (SDE) without resorting to the discretisation of their state space, which is by and large the standard approach in literature, which comes with limits related to precision and computational scalability. This is done by formulating an optimisation problem: to solve this, we have borrowed techniques and ideas from Malliavin Calculus and Mathematical Finance.

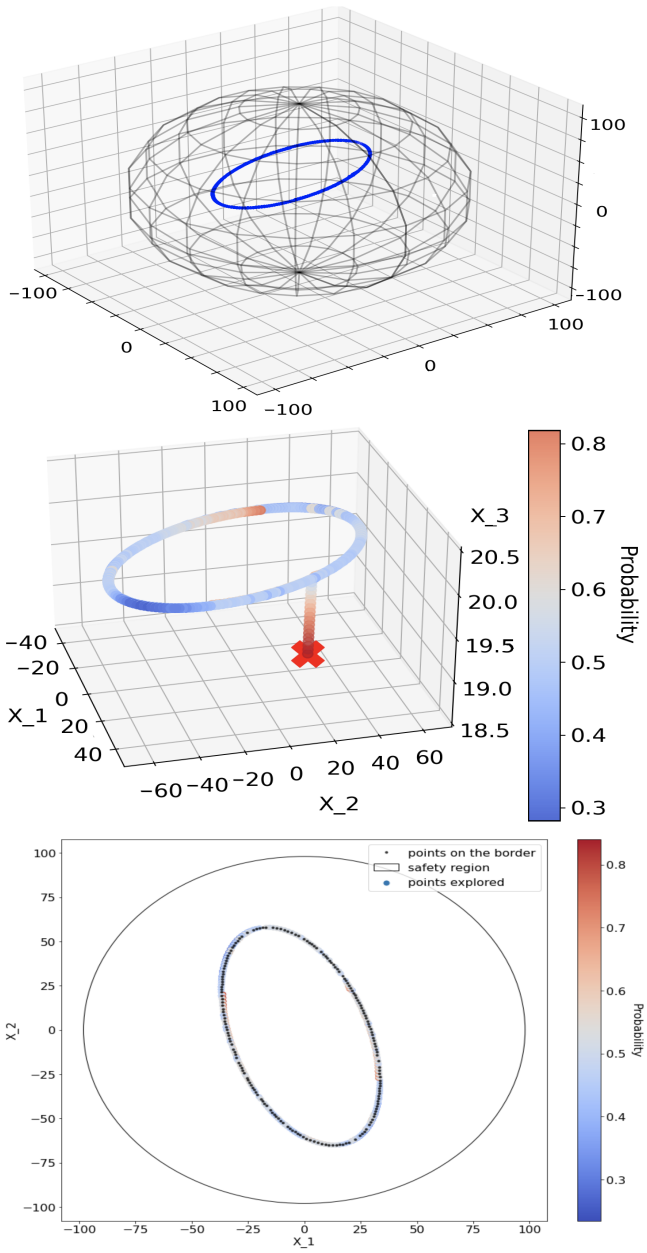

Fig. 3: The plots show the points found while and exploring the border of the 3D region $\mathbb{A}_1^{0.5}$, with respect to one plane (or section), when A is a sphere. Top: full-view of the problem. Middle: same plot but zoomed, with colour scale showing the safety probabilities of the points explored during exploration. It can be seen how starting from the red-cross point, with a probability approximately 0.8, 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 the GD has explored adjacent points. Bottom: planar section of the space considered, fixing direction x_3. The black points represent those we consider being part of the border (up to an approximation error), whilst the circle points are those explored via the GD method. It can be seen that we explore points associated with probabilities between 0.2 and 0.8.

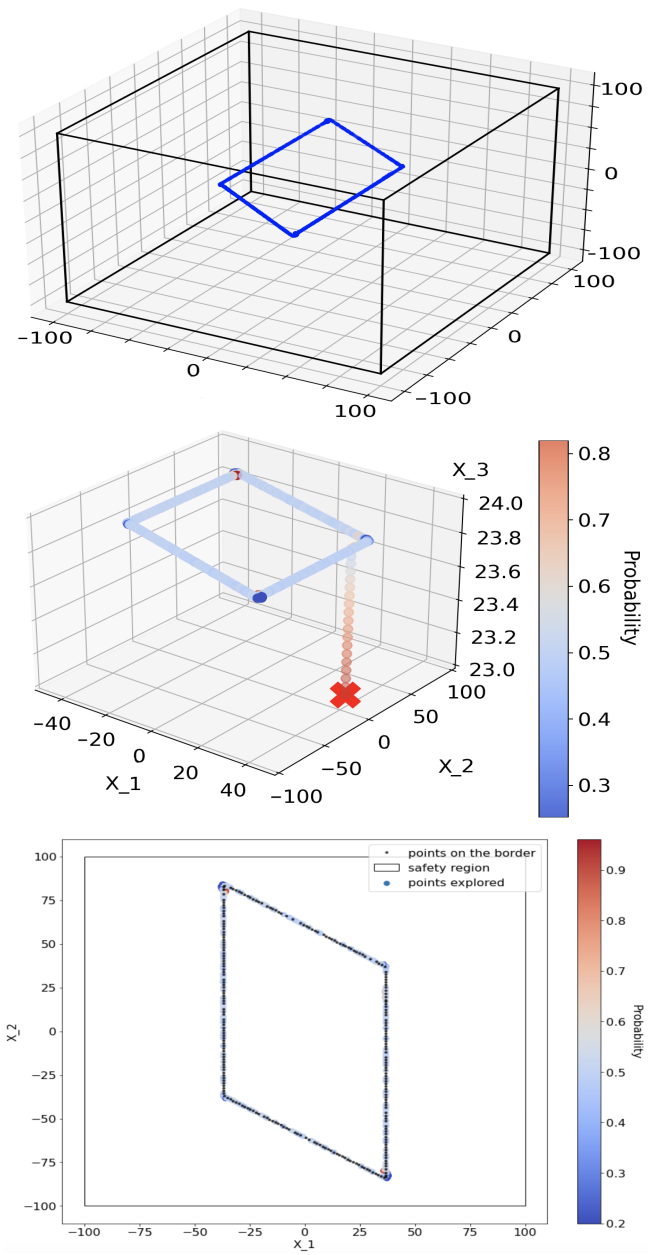

Fig. 4: Similarly to Figure 3, the plots show the points found seeking and exploring the border of the 3D region $\mathbb{A}_1^{0.5}$, with respect to one plane (or section), when A is a 3D cube.

We have discussed two formal results that allow one to explore relevant parts of the regions of interest, thus focusing computational load related to probabilistic safety computation for continuous-space models, such as SDEs. We have discussed possible algorithmic issues related the procedure, and offered strategies to cope with them. We conclude suggesting that more work on the generalisation of the approach on high-dimensional models in a completely automatic fashion is a goal worth pursuing.

Acknowledgements and Disclosure of Funding

The authors want to thank The Alan Turing Institute and the University of Oxford for the financial support given. FC is supported by the University of Oxford and The Alan

Turing Institute, TU/C/000021, under the EPSRC Grant No. EP/N510129/1. HO is supported by the EPSRC grant "Datasig" [EP/S026347/1], The Alan Turing Institute, the Oxford-Man Institute and the University of Oxford.

REFERENCES

[1] 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*. Springer Berlin Heidelberg, 2007, pp. 4–17.

[2] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems," *Automatica*, vol. 44, no. 11, pp. 2724–2734, nov 2008.

[3] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*, 1st ed. Birkhäuser Basel, 2007.

[4] 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*. IEEE, 2017, pp. 2390–2395.

[5] P. Malliavin and A. Thalmaier, *Stochastic calculus of variations in mathematical finance*. Springer-Verlag, Berlin, 2006.

[6] J. C. Hull, *Options, Futures, and Other Derivatives, Global Edition*. Pearson, 2017.

[7] J. Hu, J. Lygeros, and S. Sastry, "Towars a theory of stochastic hybrid systems," in *Hybrid Systems: Computation and Control, Third International Workshop, HSCC 2000, Pittsburgh, PA, USA, March 23-25, 2000, Proceedings*, ser. Lecture Notes in Computer Science, N. A. Lynch and B. H. Krogh, Eds., vol. 1790. Springer, 2000, pp. 160–173.

[8] M. Bujorianu and J. Lygeros, "Toward a general theory of stochastic hybrid systems," in *Stochastic Hybrid Systems*, ser. Lecture Notes in Control and Information Science, H. Blom and J. Lygeros, Eds., vol. 337. Springer, 2006.

[9] M. H. A. Davis, *Markov models and optimization*, ser. Monographs on Statistics and Applied Probability. London: Chapman & Hall, 1993, vol. 49.

[10] M. L. Bujorianu, "Extended stochastic hybrid systems and their reachability problem," in *Hybrid Systems: Computation and Control, 7th International Workshop, HSCC 2004, Philadelphia, PA, USA, March 25-27, 2004, Proceedings*, ser. Lecture Notes in Computer Science, R. Alur and G. J. Pappas, Eds., vol. 2993. Springer, 2004, pp. 234–249.

[11] 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 2006, Santa Barbara, CA, USA, March 29-31, 2006, Proceedings*, ser. Lecture Notes in Computer Science, J. P. Hespanha and A. Tiwari, Eds., vol. 3927. Springer, 2006, pp. 377–391.

[12] H. M. Soner and N. Touzi, "Stochastic target problems, dynamic programming, and viscosity solutions," *SIAM J. Control. Optim.*, vol. 41, p. 404—424, 2002.

[13] 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 2009, San Francisco, CA, USA, April 13-15, 2009. Proceedings*, ser. Lecture Notes in Computer Science, R. Majumdar and P. Tabuada, Eds., vol. 5469. Springer, 2009, pp. 460–464.

[14] P. Mohajerin Esfahani, D. Chatterjee, and J. Lygeros, "The stochastic reach-avoid problem and set characterization for diffusions," *Automatica*, vol. 70, pp. 43–56, 2016.

[15] R. Wisniewski, M. Svenstrup, A. Pedersen, and C. Steiniche, "Certificate for safe emergency shutdown of wind turbines," in *Proceedings of American Control Conference*, 2013.

[16] R. Wisniewski, M. L. Bujorianu, and C. Sloth, "p-safe analysis of stochastic hybrid processes," *IEEE Trans. Autom. Control.*, vol. 65, no. 12, pp. 5220–5235, 2020.

[17] R. Wisniewski and C. Sloth, "Safety analysis of stochastic dynamical systems," in *IFAC Conference on Analysis and Design of Hybrid Systems, ADHS*. IFAC, 2015.

[18] 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*. IEEE, 2019, pp. 4433–4438.

[19] 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 2003, Cambridge, UK, September 1-4, 2003*. IEEE, 2003, pp. 2225–2230.

[20] M. Prandini, J. Hu, J. Lygeros, and S. Sastry, "A probabilistic approach to aircraft conflict detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 4, pp. 199–220, 2000.

[21] F. Shmarov and P. Zuliani, "Probabilistic hybrid systems verification via smt and monte carlo techniques," in *12th Haifa Verification Conference HVC*, ser. Lecture Notes in Computer Science, vol. 10028. Springer, 2006, pp. 152–168.

[22] M. Zamani, P. M. Esfahani, R. Majumdar, A. Abate, and J. Lygeros, "Symbolic control of stochastic systems via approximately bisimilar finite abstractions," *IEEE Trans. Autom. Control.*, vol. 59, no. 12, pp. 3135–3150, 2014.

[23] M. Zamani, P. M. Esfahani, A. Abate, and J. Lygeros, "Symbolic models for stochastic control systems without stability assumptions," in *12th European Control Conference, ECC 2013, Zurich, Switzerland, July 17-19, 2013*. IEEE, 2013, pp. 4257–4262.

[24] S. W. N. Ikeda, *Stochastic Differential Equations and Diffusion Processes*. Elsevier S&T, 2014.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980v9*, Dec. 2014.

[26] 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*, vol. 3, no. 4, pp. 391–412, 1999.

[27] 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*, vol. 5, no. 2, pp. 201–236, 2001.

[28] N. Privault and X. Wei, "A malliavin calculus approach to sensitivity analysis in insurance," *Econometrics eJournal*, 2003.

[29] V. Bally, G. Pages, and J. Printems, "A quantization tree method for pricing and hedging multidimensional american options," *Mathematical Finance*, vol. 15, no. 1, pp. 119–168, jan 2005.

[30] E. Gobet and R. Munos, "Sensitivity analysis using it[o-circumflex]–malliavin calculus and martingales, and application to stochastic optimal control," *SIAM J. Control. Optim.*, vol. 43, pp. 1676–1713, 2005.

[31] E. Gobet and A. Kohatsu-Higa, "Computation of Greeks for barrier and look-back options using Malliavin calculus," *Electronic Communications in Probability*, vol. 8, pp. 51–62, 2003.

[32] H. Kunita, "Stochastic differential equations and stochastic flows of diffeomorphisms," in *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 1984, pp. 143–303.

[33] L. C. Kinsey, *Topology of Surfaces*. Springer New York, 1993.

[34] E. Haines, "Point in polygon strategies," in *Graphics Gems*. Elsevier, 1994, pp. 24–46.

[35] K. Hormann and A. Agathos, "The point in polygon problem for arbitrary polygons," *Computational Geometry*, vol. 20, no. 3, pp. 131–144, nov 2001.

[36] G. N. Kumar and M. Bangi, "An extension to winding number and point-in-polygon algorithm," *IFAC-PapersOnLine*, vol. 51, no. 1, pp. 548–553, 2018.

[37] C.-W. Huang and T.-Y. Shih, "On the complexity of point-in-polygon algorithms," *Computers & Geosciences*, vol. 23, no. 1, pp. 109–118, feb 1997.

[38] P. Glasserman, *Monte Carlo Methods in Financial Engineering*. New York, NY: Springer New York, 2003.

[39] H. T. Huynh, V. S. Lai, and I. Soumaré, "Solution of stochastic differential equations," in *Stochastic Simulation and Applications in Finance with MATLAB® Programs*. John Wiley & Sons, Ltd., may 2012, pp. 123–148.

[40] P. Andersson and A. Kohatsu-Higa, "Unbiased simulation of stochastic differential equations using parametrix expansions," *Bernoulli*, vol. 23, pp. 2028–2057, 2017.

[41] 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*, vol. 24, no. 0, 2019.

[42] P. Henry-Labordère, X. Tan, and N. Touzi, "Unbiased simulation of stochastic differential equations," *The Annals of Applied Probability*, vol. 27, no. 6, dec 2017.

[43] E. Gobet, "Weak approximation of killed diffusion using Euler schemes," *Stochastic Processes and their Applications*, vol. 87, no. 2, pp. 167–197, 2000.

[44] E. Gobet and S. Menozzi, "Discrete sampling of functionals of ito processes," in *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 2007, pp. 355–374.

[45] ——, "Stopped diffusion processes: Boundary corrections and overshoot," *Stochastic Processes and their Applications*, vol. 120, no. 2, pp. 130–162, feb 2010.

[46] K. G. Suffern, "Quadtree algorithms for contouring functions of two variables," *The Computer Journal*, vol. 33, no. 5, pp. 402–407, may 1990.

Francesco Cosentino is a Doctoral Candidate in the Mathematical Institute at the University of Oxford and The Alan Turing Institute in London. He received a BSc and an MSc in Mathematics respectively in July 2014 and July 2016 from the University of Turin (IT). For his BSc and MSc, he received a scholarship from Collegio Carlo Alberto (Turin, IT), from which he obtained an MA in Financial Economics in October 2016. After his studies and before joining Oxford, he has been an FX risk analyst at Fiat Chrysler Automobiles.

Harald Oberhauser obtained his PhD in 2010 in the Statslab at the University of Cambridge and held post-doctoral positions at the Technical University of Berlin, and the Oxford-Man Institute. In 2014 he joined the Mathematics Department of University College London, and in 2015 he joined the Mathematical Institute in Oxford as Associate Professor. His research focuses on stochastic analysis and its applications.

Alessandro Abate (S'02-M'08-SM'19) is Professor of Verification and Control in the Department of Computer Science at the University of Oxford, and a fellow of the Alan Turing Institute for Data Sciences in London. He received a Laurea in Electrical Engineering in October 2002 from the University of Padova (IT), an MS in May 2004 and a PhD in December 2007, both in Electrical Engineering and Computer Sciences, at UC Berkeley (USA). He has been an International Fellow in the CS Lab at SRI International in Menlo Park (USA), and a PostDoctoral Researcher at Stanford University (USA), in the Department of Aeronautics and Astronautics. He has also been an Assistant Professor at the Delft Centre for Systems and Control, TU Delft (NL).