Manuscript submitted to AIMS Journal

doi:10.3934/xx.xxxxxxx

PIECEWISE QUADRATIC LYAPUNOV FUNCTIONS FOR STOCHASTIC DIFFERENTIAL EQUATIONS BY LINEAR PROGRAMMING

Peter Giesl*[1], Sigurdur Hafstein[2] and Sareena Pokkakkillath[3]

[1,3]Department of Mathematics, University of Sussex, Falmer BN1 9QH, United Kingdom

[2]Faculty of Physical Sciences, University of Iceland, Dunhagi 5, IS-107 Reykjavik, Iceland

(Communicated by Handling Editor)

Abstract. We develop an algorithm to parameterize continuous and piecewise quadratic (CPQ) Lyapunov functions for stochastic differential equations (SDEs) using linear programming (LP). The algorithm is a non-trivial extension of algorithms to parameterize continuous and piecewise linear (CPA) Lyapunov functions for ordinary differential equations (ODEs), but since the conditions for a Lyapunov function for a stochastic system involve second order derivatives, CPA Lyapunov functions cannot exist for stochastic systems and hence CPQ Lyapunov functions are needed. We demonstrate our algorithm on two examples from the literature.

1. Introduction.

Continuous piecewise affine (CPA) functions have been widely used as a method to approximate functions of interest, such as in the computation of Lyapunov functions for autonomous dynamical systems, see [12]. Since the conditions for a Lyapunov function for stochastic differential equations (SDEs) involve the infinitesimal generator for the SDE, they include second-order derivatives of the Lyapunov functions, and therefore such a function cannot be piecewise linear. In general, the computation of Lyapunov functions for SDEs is considerably more difficult than for ODEs, see, e.g. [8, 14].

In this paper we develop an algorithm that uses linear programming (LP) to compute continuous piecewise quadratic (CPQ) Lyapunov functions for SDEs. More exactly, from a feasible solution to the LP problem, a CPQ function can be parameterized and this CPQ function can be mollified to a smooth function, which is a non-local Lyapunov function for the SDE in question, see Theorem 3.4. The mollification can be be done such that the level-sets of the parameterized CPQ function and the smooth non-local Lyapunov function are arbitrarily close. Since the level-sets are the interesting part of a non-local Lyapunov function, we will simply refer to the CPQ function with these properties as a non-local Lyapunov function. Note that the mollification of a piecewise linear function, i.e. a CPA function, would deliver a function whose second-order derivative is zero, except at the boundaries of the areas where the original function is linear. Hence, it is not suited to assert stability for an SDE.

2020 *Mathematics Subject Classification.* Primary: 60H35, 60H10; Secondary: 65P40, 37C75.

Key words and phrases. Stochastic differential equation, Lyapunov function, continuous piecewise quadratic function, linear programming, γ-basin of attraction.

*Corresponding author: Peter Giesl.

Note that the stability of equilibria for SDEs is a very difficult problem. In the literature, to the best of the authors' knowledge, this problem has only been addressed analytically for particular systems with a simple structure and often the analysis is reduced to verifying mean-square exponential stability, see e.g. [21, 25, 27, 29, 31]. For general nonlinear systems Lyapunov functions have been computed by solving the corresponding stochastic Zubov PDE numerically with [5, 7] or without [14] subsequent verification. For linear SDEs with constant coefficients the problem of computing a Lyapunov function assuring mean-square exponential stability is not too difficult; in general a linear matrix inequality (LMI) approach can be used, see e.g. [3], and in two dimensions there is even an analytical solution [32]. For computing Lyapunov functions to assert the more general global asymptotic stability for linear SDEs with constant coefficients, one can formulate the conditions for a Lyapunov function as bilinear matrix inequalities (BMIs) [3, 4, 16], which can often be solved using heuristics, see e.g. [23, 24].

In this paper we will discuss and review the necessary theory for our algorithm. In particular, in Section 2 we thoroughly discuss our parameterization of CPQ functions, their relation to CPA functions, and derive formulas for their gradient and Hessian in a form that is suited for our LP problem. Section 3 contains the main results of the paper. We first discuss SDEs as well as local- and non-local Lyapunov functions for them, before we derive conditions for a CPQ function to be a non-local Lyapunov function. In Theorem 3.4 we then prove the claimed properties above about the mollification of the CPQ non-local Lyapunov function. We conclude Section 3 by stating our linear programming problem in LP Problem 3.5 and prove that a feasible solution to it parameterizes a CPQ non-local Lyapunov function for the system in question. In Section 4 we compute CPQ Lyapunov functions for two examples from the literature and then we conclude the paper in Section 5.

2. Continuous Piecewise Quadratic (CPQ) Functions.

For a given triangulation $\mathcal{T}$, the set of CPQ functions defined over $\mathcal{T}$ is an extension of the set of CPA functions defined over $\mathcal{T}$. Therefore it is advantageous to first discuss CPA functions and triangulations, before we move the focus to CPQ functions.

2.1. Triangulation and CPA functions.

In this section, we first define simplices and triangulations. Then we define CPA functions and summarize some of their characteristics such as their constant gradient over each simplex and the relation to the shape matrices of the triangulation. Throughout this section, we use the same notation as [12].

Definition 2.1 (simplex)**.** Let $(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_m)$ be an ordered $(m+1)$-tuple of affinely independent vectors in $\mathbb{R}^n$, meaning that $\sum_{i=1}^m \lambda_i(\mathbf{x}_i - \mathbf{x}_0) = \mathbf{0} \Rightarrow \lambda_i = 0\ \forall i = 1, 2, \dots, m$; in particular, this implies $m \le n$. Denote the set of all convex combinations of these vectors by

$$\mathrm{co}(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_m) := \left\{ \sum_{i=0}^m \lambda_i \mathbf{x}_i : 0 \le \lambda_i \le 1, \sum_{i=0}^m \lambda_i = 1 \right\}.$$

Then the set $\mathcal{G} := \mathrm{co}(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_m)$ is called an m-simplex and $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_m$ are its vertices. A face of the m-simplex $\mathrm{co}\big(\mathbf{x}_0, \dots, \mathbf{x}_m\big)$ is a k-simplex, where $0 \le k < m$, and the vertices of the k-simplex, the face, are a subset of the vertices of the m-simplex.

A collection of n-simplices in $\mathbb{R}^n$ is called a triangulation if it fulfills the conditions of the next definition.

Definition 2.2 (triangulation)**.** Let $\mathcal{T}$ be a collection of n-simplices $\mathcal{G}_\nu$ in $\mathbb{R}^n$. If every pair of distinct simplices $\mathcal{G}_\nu, \mathcal{G}_\mu \in \mathcal{T}$, $\nu \neq \mu$, either intersects in a common face or not all, then we call $\mathcal{T}$ a triangulation.

We define the vertex set of a triangulation $\mathcal{T}$ by

$$\mathcal{V}_\mathcal{T} := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \text{ is a vertex of a simplex in } \mathcal{T}\},$$

and we say that $\mathcal{T}$ is a triangulation of the set $\mathcal{D}_\mathcal{T}$ if

$$\mathcal{D}_\mathcal{T} = \bigcup_{\mathcal{G}_\nu \in \mathcal{T}} \mathcal{G}_\nu.$$

We now recall the definitions of a CPA function, shape matrix and its gradient from [12].

Definition 2.3 (CPA function)**.** Let $\mathcal{T} = (\mathcal{G}_\nu)$ be a triangulation of a set $\mathcal{D}_\mathcal{T} \subset \mathbb{R}^n$. Assume we are given a number $P_\mathbf{x} \in \mathbb{R}$ for each vertex $\mathbf{x} \in \mathcal{V}_\mathcal{T}$. Then we can uniquely define a continuous piecewise affine function $P : \mathcal{D}_\mathcal{T} \to \mathbb{R}$ in the following way:

(i) $P(\mathbf{x}) := P_\mathbf{x}$ for every $\mathbf{x} \in \mathcal{V}_\mathcal{T}$;

(ii) P is affine on every simplex $\mathcal{G}_\nu \in \mathcal{T}$, i.e. there exists a vector $\mathbf{a}_\nu \in \mathbb{R}^n$ and a number $b_\nu \in \mathbb{R}$ such that $P(\mathbf{x}) = \mathbf{a}_\nu^\mathrm{T}\mathbf{x} + b_\nu$ for all $\mathbf{x} \in \mathcal{G}_\nu$.

For each simplex $\mathcal{G}_\nu \in \mathcal{T}$ we define $\boldsymbol{\nabla} P_\nu := \boldsymbol{\nabla} P|_{\mathcal{G}_\nu} = \mathbf{a}_\nu$. We note that $\boldsymbol{\nabla} P_\nu$ is constant on every simplex $\mathcal{G}_\nu$.

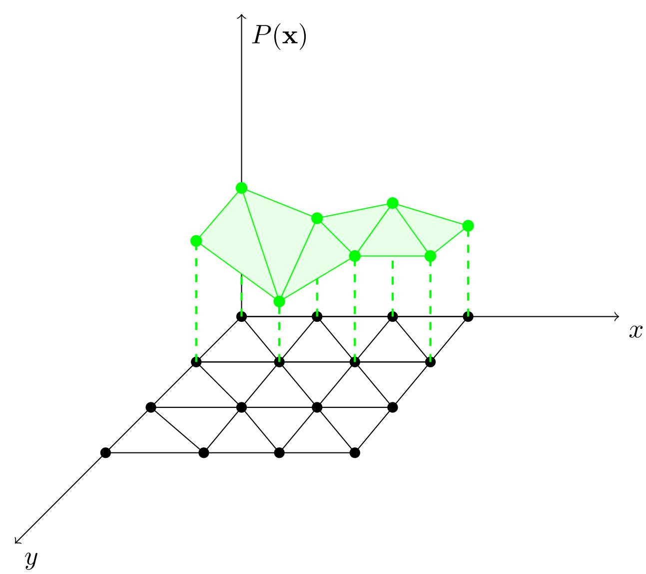

Figure 1. The green plot is a visualisation of part of a CPA function $P : \mathcal{D}_\mathcal{T} \to \mathbb{R}$ where $\mathcal{D}_\mathcal{T} \subset \mathbb{R}^2$. The green nodes represent the number $P_\mathbf{x} \in \mathbb{R}$ that we are given for every $\mathbf{x} \in \mathcal{V}_\mathcal{T}$ (which are represented by the black nodes).

Definition 2.4 (shape matrices)**.** Let $\mathcal{T} = (\mathcal{G}_\nu)$ be a triangulation. For a simplex $\mathcal{G}_\nu = \text{co}(\mathbf{x}_0^\nu, \mathbf{x}_1^\nu, \ldots, \mathbf{x}_n^\nu) \in \mathcal{T}$, we define its shape matrix $\mathbf{X}_\nu \in \mathbb{R}^{n\times n}$ by

$$\mathbf{X}_\nu := \left(\mathbf{x}_1^\nu - \mathbf{x}_0^\nu, \mathbf{x}_2^\nu - \mathbf{x}_0^\nu, \ldots, \mathbf{x}_n^\nu - \mathbf{x}_0^\nu\right)^{\mathrm{T}}. \tag{1}$$

We refer to the set $\{\mathbf{X}_\nu : \mathcal{G}_\nu \in \mathcal{T}\}$ as the shape matrices of the triangulation $\mathcal{T}$.

Remark 2.5. Recall from Definition 2.1 that $\mathbf{x}_0^\nu$, $\mathbf{x}_1^\nu, \ldots, \mathbf{x}_n^\nu$ are affinely independent. This means that the vectors $\mathbf{x}_i^\nu - \mathbf{x}_0^\nu$, $i = 1, \ldots, n$, are linearly independent. Therefore, the rows of $\mathbf{X}_\nu$ are linearly independent so $\mathbf{X}_\nu$ is non-singular and has an inverse.

Lemma 2.6. [11, Remark 9] *Let $\mathcal{T}$ be a triangulation, $P : \mathcal{D}_\mathcal{T} \to \mathbb{R}$ a CPA function, and $\mathcal{G}_\nu = \text{co}(\mathbf{x}_0^\nu, \mathbf{x}_1^\nu, \ldots, \mathbf{x}_n^\nu) \in \mathcal{T}$. Then the gradient of P on the simplex $\mathcal{G}_\nu$ is $\boldsymbol{\nabla} P_\nu = \mathbf{X}_\nu^{-1}\mathbf{p}$ where $\mathbf{p} = (p_1, p_2, \ldots, p_n)^{\mathrm{T}}$ is a column vector with entries $p_i := P_{\mathbf{x}_i^\nu} - P_{\mathbf{x}_0^\nu}$ for $i = 1, 2, \ldots, n$.*

For the CPA interpolation of a function $f\colon \mathcal{G}_\nu \to \mathbb{R}$ we will use the following well-known estimate, cf. e.g. [2, Proposition 4.1].

Lemma 2.7. *Let $f\colon \mathcal{G}_\nu \to \mathbb{R}$ be twice continuously differentiable and denote by $\mathbf{H}(f)\colon \mathcal{G}_\nu \to \mathbb{R}^{n\times n}$ its Hessian, and by $\mathbf{H}\big(f(\mathbf{w})\big)$ the value of the Hessian at $\mathbf{w} \in \mathcal{G}_\nu$. For the CPA interpolation of f, i.e. $\mathbf{x} \mapsto \sum_{k=0}^n \lambda_k f(\mathbf{x}_k^\nu)$ where $\mathbf{x} = \sum_{k=0}^n \lambda_k \mathbf{x}_k^\nu$ with $\lambda_k \in [0, 1]$ and $\sum_{k=0}^n \lambda_k = 1$, the following estimate holds*

$$\left| f\left(\sum_{k=0}^n \lambda_k \mathbf{x}_k^\nu\right) - \sum_{k=0}^n \lambda_k f(\mathbf{x}_k^\nu)\right| \le h_\nu^2 B,$$

where $h_\nu = \max_{\mathbf{x}_i^\nu \in \mathcal{G}_\nu} \|\mathbf{x}_i^\nu - \mathbf{x}_0^\nu\|_2$ and $B := \max_{\mathbf{w}\in\mathcal{G}_\nu} \left\|\mathbf{H}\big(f(\mathbf{w})\big)\right\|_2$.

Note that $\|\mathbf{x}\|_2$ denotes the Euclidean norm of a vector $\mathbf{x} \in \mathbb{R}^n$ and $\|A\|_2 := \max_{\|\mathbf{x}\|_2=1} \|A\mathbf{x}\|_2$ for $A \in \mathbb{R}^{n\times n}$ denotes the induced matrix norm.

2.2. **CPQ functions: definition.** Having reviewed CPA functions, we now introduce continuous piecewise quadratic (CPQ) functions. If we have a triangulation $\mathcal{T} = (\mathcal{G}_\nu)$ of the set $\mathcal{D}_\mathcal{T}$ and we are given a number $r_\mathbf{x} \in \mathbb{R}$ for every vertex $\mathbf{x} \in \mathcal{V}_\mathcal{T}$ such that $f(\mathbf{x}) = r_\mathbf{x}$, then, by Definition 2.3, we can construct a function $P : \mathcal{D}_\mathcal{T} \to \mathbb{R}$ satisfying $P(\mathbf{x}) = r_\mathbf{x}$ for every $\mathbf{x} \in \mathcal{V}_\mathcal{T}$ that is affine on every simplex $\mathcal{G}_\nu$. That is, P is a CPA interpolation of f.

To construct a CPQ function $g : \mathcal{D}_\mathcal{T} \to \mathbb{R}$, that is continuous on its whole domain and a quadratic function on each $\mathcal{G}_\nu$, we first add more points $\mathbf{x} \in \mathbb{R}^n$ where we demand $g(\mathbf{x}) = r_\mathbf{x}$. For any two distinct vertices $\mathbf{x}_i$, $\mathbf{x}_j$ of a simplex $\mathcal{G}_\nu \in \mathcal{T}$, we introduce a new point $\mathbf{x}_{ij} := (\mathbf{x}_i + \mathbf{x}_j)/2$ to be the midpoint between $\mathbf{x}_i$ and $\mathbf{x}_j$; note that $\mathbf{x}_{ij} = \mathbf{x}_{ji}$. Let us denote by $\mathcal{V}_\mathcal{T}^{\text{CPQ}}$ the set containing the vertices $\mathcal{V}_\mathcal{T}$ of $\mathcal{T}$ and these midpoints.

We will prove in the next theorem that we can uniquely construct a function $g : \mathcal{D}_\mathcal{T} \to \mathbb{R}$ by demanding that g is continuous and quadratic and $g(\mathbf{x}) = r_\mathbf{x}$ for every $\mathbf{x} \in \mathcal{V}_\mathcal{T}^{\text{CPQ}}$, where the $r_\mathbf{x} \in \mathbb{R}$ are given numbers, see Figure 2 (left) for a schematic representation in one dimension and Figure 2 (right) for the vertices $\mathcal{V}_\mathcal{T}^{\text{CPQ}}$ of one simplex in two dimensions.

Theorem 2.8. *Let $\mathcal{T} = (\mathcal{G}_\nu)$ be a triangulation of a set $\mathcal{D}_\mathcal{T} \subset \mathbb{R}^n$. For each pair of distinct vertices $\mathbf{x}_k, \mathbf{x}_l$ of a simplex $\mathcal{G}_\nu = \text{co}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{T}$, $k, l \in \{0, 1, \ldots, n\}$,*

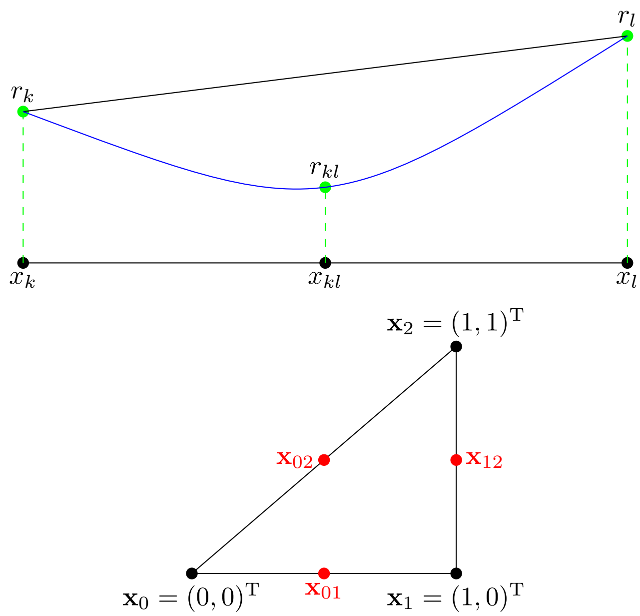

FIGURE 2. (left) Visual representation of the addition of an extra 'vertex' x_{kl} for a one-dimensional simplex. The blue curve represents a CPQ function, while the black line is a CPA function. Note that $r_k := r_{x_k}$ and $r_{kl} := r_{x_{kl}}$. (right) Visual representation of the addition of extra 'vertices' for a two-dimensional simplex.

define $\mathbf{x}_{kl} := (\mathbf{x}_k + \mathbf{x}_l)/2$ *to be the midpoint of the line segment connecting* $\mathbf{x}_k$ *and* $\mathbf{x}_l$*. Denote by* $\mathcal{V}_{\mathcal{T}}^{CPQ}$ *the set of all vertices* $\mathcal{V}_{\mathcal{T}}$ *and all such midpoints, i.e.*

$$\mathcal{V}_{\mathcal{T}}^{CPQ} := \left\{ \frac{\mathbf{x}+\mathbf{y}}{2} \ : \mathbf{x} \text{ and } \mathbf{y} \text{ are vertices of some } \mathcal{G}_\nu, \text{ where } \mathcal{G}_\nu \in \mathcal{T} \right\}; \tag{2}$$

note $\mathbf{x}$ *and* $\mathbf{y}$ *are not necessarily different.*

Assume we are given a value $r_{\mathbf{x}} \in \mathbb{R}$ *for each* $\mathbf{x} \in \mathcal{V}_{\mathcal{T}}^{CPQ}$*. Define the function* $g : \mathcal{D}_{\mathcal{T}} \to \mathbb{R}$ *as follows: For an* $\mathbf{x} \in \mathcal{D}_{\mathcal{T}}$ *there exists a simplex* $\mathcal{G}_\nu$ *such that* $\mathbf{x} \in \mathcal{G}_\nu$ *and* $\mathbf{x}$ *can be written uniquely as a convex combination of its vertices* $\mathbf{x}_0, \ldots, \mathbf{x}_n$

$$\mathbf{x} = \sum_{k=0}^{n} \lambda_k \mathbf{x}_k, \quad \sum_{k=0}^{n} \lambda_k = 1, \quad \text{and} \quad \lambda_k \geq 0 \quad \text{for } k = 0, \ldots, n.$$

We define

$$g(\mathbf{x}) = \sum_{k=0}^{n} \lambda_k r_k + 2 \sum_{k=0}^{n} \sum_{l=k+1}^{n} \lambda_k \lambda_l (2r_{kl} - r_k - r_l), \tag{3}$$

where $r_k := r_{\mathbf{x}_k}$ *and* $r_{kl} := r_{\mathbf{x}_{kl}}$ *for* $k, l = 0, \ldots, n$, $k \neq l$.

Then $g : \mathcal{D}_{\mathcal{T}} \to \mathbb{R}$ *is a well-defined continuous function that is quadratic on each* $\mathcal{G}_\nu$ *and fulfills* $g(\mathbf{x}) = r_{\mathbf{x}}$ *for every* $\mathbf{x} \in \mathcal{V}_{\mathcal{T}}^{CPQ}$.

Proof. We first verify that $g(\mathbf{x}_k) = r_k$ for all $\mathbf{x}_k \in \mathcal{V}_{\mathcal{T}}$ and $g(\mathbf{x}_{kl}) = r_{kl}$ for every $\mathbf{x}_{kl} \in \mathcal{V}_{\mathcal{T}}^{\mathrm{CPQ}} \setminus \mathcal{V}_{\mathcal{T}}$.

Verifying that $g(\mathbf{x}_k) = r_k$: Let $\mathbf{x}_k$ be any vertex of an arbitrary simplex $\mathcal{T} \ni \mathcal{G} = \text{co}\big(\mathbf{x}_0, \dots, \mathbf{x}_n\big)$. By Definition 2.1, we know that $\mathbf{x}_k$ can be written uniquely as a convex combination of the vertices of $\mathcal{G}$, meaning that $\mathbf{x}_k = \sum_{i=0}^n \lambda_i \mathbf{x}_i$ and we must have $\lambda_k = 1$ and all other $\lambda_i = 0$. Then, it is clear from (3) that $g(\mathbf{x}_k) = r_k$.
Verifying that $g(\mathbf{x}_{kl}) = r_{kl}$: Let $\mathbf{x}_{kl}$ be the midpoint of any two distinct vertices $\mathbf{x}_k$, $\mathbf{x}_l$ of an arbitrary simplex $\mathcal{G} = \text{co}\big(\mathbf{x}_0, \dots, \mathbf{x}_n\big) \in \mathcal{T}$, where we assume $k < l$ without loss of generality. Again using the property that since $\mathbf{x}_{kl} \in \mathcal{G}$ we can write $\mathbf{x}_{kl}$ uniquely as a convex combination of the vertices of $\mathcal{G}$ and we must have $\lambda_k = \lambda_l = \frac{1}{2}$ and all other $\lambda_i = 0$. Then, it is clear from (3) that $g(\mathbf{x}_{kl}) = \frac{1}{2}r_k + \frac{1}{2}r_l + 2[\frac{1}{4}(2r_{kl} - r_k - r_l)] = r_{kl}$.

To show that g is a quadratic function on each simplex, we refer to Remark 2.12, which shows that the Hessian of g is constant on each simplex $\mathcal{G}$ and thus g is quadratic.

Finally, we show that g is continuous. By Definition 2.2, we know that simplices either intersect in a common face or not at all. Assume that $\mathcal{G}_\nu$ and $\mathcal{G}_\mu$ are two distinct simplices that intersect in a common face. This means that on this face they share the same vertices. Therefore, the CPQ function defined on this face will be the same for both $\mathcal{G}_\nu$ and $\mathcal{G}_\mu$. Hence, g is continuous over the whole domain $\mathcal{D}_\mathcal{T}$ since it is continuous on each simplex and on the shared common faces between neighbouring simplices. □

Remark 2.9. Consider a function $f : \mathcal{D}_\mathcal{T} \to \mathbb{R}$ and fix $r_\mathbf{x} := f(\mathbf{x})$ for every $\mathbf{x} \in \mathcal{V}_\mathcal{T}^{\text{CPQ}}$. We note that a CPA function P that interpolates f at the vertices $\mathbf{x}_k \in \mathcal{V}_\mathcal{T}$, i.e. $P(\mathbf{x}) = f(\mathbf{x})$ for every $\mathbf{x} \in \mathcal{V}_\mathcal{T}$ is given by $P(\mathbf{x}) = \sum_{k=0}^n \lambda_k r_k$ with the notation from Theorem 2.8. Comparing this with the CPQ function g defined in (3), we observe that g contains the CPA function P plus some additional quadratic terms, i.e.

$$g(\mathbf{x}) = P(\mathbf{x}) + 2\sum_{k=0}^{n}\sum_{l=k+1}^{n} \lambda_k\lambda_l(2r_{kl} - r_k - r_l) \text{ on } \mathcal{G}_\nu.$$

2.3. **CPQ functions: useful formulas.** In Subsection 2.1, we defined for every simplex $\mathcal{G}_\nu$ of a triangulation $\mathcal{T}$ the gradient $\boldsymbol{\nabla} P_\nu$ of a CPA function P, and provided an explicit formula to calculate it. We will now derive formulas to compute the gradient and Hessian of a CPQ function g for every simplex $\mathcal{G}_\nu$. Similarly to the formula given in Lemma 2.6 for $\boldsymbol{\nabla} P_\nu$, we will observe that both of our formulas for the gradient and Hessian on $\mathcal{G}_\nu$ of a CPQ function will involve the simplex's shape matrix $\mathbf{X}_\nu$. Note that the components of the gradient of g are CPA functions and the Hessian of g is constant on each $\mathcal{G}_\nu$.

We proceed in the following way to derive the gradient $\boldsymbol{\nabla} g_\nu$, where the notation g_ν denotes the restriction $g_\nu := g\big|_{\mathcal{G}_\nu}$ of g to $\mathcal{G}_\nu$. We compute the directional derivative of $g_\nu\big(\mathbf{x}\big)$ in direction $(\mathbf{x}_i - \mathbf{x}_0)$ in two ways: we first use the representation of $\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)$ and take the derivative of $g_\nu\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big)$ with respect to h, and then set $h = 0$. On the one hand, we use the chain rule to compute the same derivative. For the Hessian $\mathbf{H}_\nu$, we proceed similarly and compute the derivative of $\frac{\partial}{\partial x_j} g_\nu\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big)$ with respect to h for some fixed $i \in \{0, \dots, n\}$ and $j \in \{1, \dots, n\}$.

We give a formula for the gradient in the following proposition.

Proposition 2.10. *Let $\mathcal{T} = (\mathcal{G}_\nu)$ be a triangulation of a set $\mathcal{D}_\mathcal{T} \subset \mathbb{R}^n$ and $g : \mathcal{D}_\mathcal{T} \to \mathbb{R}$ be a CPQ function over this set. Then, for each simplex $\mathcal{G}_\nu = \text{co}(\mathbf{x}_0, \dots, \mathbf{x}_n)$, the* *gradient of g restricted to $\mathcal{G}_\nu$ is given by the formula*

$$\nabla g_\nu(\mathbf{x}) = \mathbf{X}_\nu^{-1}\mathbf{b}(\mathbf{x}) \tag{4}$$

for all $\mathbf{x} \in \mathcal{G}_\nu$. Here, $\mathbf{X}_\nu$ is the shape matrix of the simplex $\mathcal{G}_\nu$, see (1), and $\mathbf{b}(\mathbf{x}) = \big(b_1(\mathbf{x}), \dots, b_n(\mathbf{x})\big)^{\mathrm{T}}$ is a column vector with entries given by (5)

$$b_i(\mathbf{x}) = b_i\left(\sum_{j=0}^n \lambda_j \mathbf{x}_j\right) = \sum_{j=0}^n \lambda_j b_i(\mathbf{x}_j) \text{ with } \quad b_i(\mathbf{x}_j) := r_0 - r_i + 4s^i(\mathbf{x}_j) - 4t(\mathbf{x}_j), \tag{5}$$

where $\mathbf{x} = \sum_{j=0}^n \lambda_j \mathbf{x}_j$ with $\sum_{j=0}^n \lambda_j = 1$ and $\lambda_j \in [0,1]$, is the unique representation of $\mathbf{x}$ as a convex combination of the vertices of $\mathcal{G}_\nu$,

$$s^i(\mathbf{x}_j) = \begin{cases} r_{ji}, & j \neq i \\ r_i, & j = i \end{cases} \qquad \text{and} \qquad t(\mathbf{x}_j) = \begin{cases} r_{0j}, & j \neq 0 \\ r_0, & j = 0, \end{cases}$$

and the r_k, r_{kl} are defined by (3). Note that the $b_i(\mathbf{x}_j)$ are linear in the r_k and r_{kl}, where $k, l = 0, 1, \dots, n$ and $k \neq l$.

Proof. Consider an arbitrary simplex $\mathcal{G}_\nu \in \mathcal{T}$. For any $\mathbf{x} \in \mathcal{G}_\nu = \mathrm{co}(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n)$, we write $\mathbf{x} = \sum_{j=0}^n \lambda_j \mathbf{x}_j$ with $\lambda_j \geq 0$ satisfying $\sum_{j=0}^n \lambda_j = 1$. Therefore, for any fixed $i \in \{0, \dots, n\}$ and $\mathbf{x} \in \mathcal{G}_\nu$, we have that

$$\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0) = (\lambda_0 - h)\mathbf{x}_0 + \sum_{j=1}^{i-1} \lambda_j \mathbf{x}_j + (\lambda_i + h)\mathbf{x}_i + \sum_{j=i+1}^{n} \lambda_j \mathbf{x}_j. \tag{6}$$

By (3) we have

$$\begin{aligned} g_\nu(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)) = (\lambda_0 - h)r_0 &+ \sum_{\substack{l=1 \\ l\neq i}}^n \lambda_l r_l + (\lambda_i + h) r_i \\ &+ 2\sum_{\substack{l=1 \\ l\neq i}}^n (\lambda_0 - h)\lambda_l(2r_{0l} - r_0 - r_l) \\ &+ 2\sum_{\substack{l=1 \\ l\neq i}}^n (\lambda_i + h)\lambda_l(2r_{il} - r_i - r_l) \\ &+ 2(\lambda_0 - h)(\lambda_i + h)(2r_{0i} - r_0 - r_i). \end{aligned}$$

Now we take the derivative of $g_\nu\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big)$ with respect to h. Using that $r_{ki} = r_{ik}$ for any $i, k \in \{0, \dots, n\}$, $i \neq k$, we obtain

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}h} g_\nu\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big) \\ &= \frac{\mathrm{d}}{\mathrm{d}h}\left[(\lambda_0 - h)r_0 + \sum_{\substack{l=1 \\ l\neq i}}^n \lambda_l r_l + (\lambda_i + h) r_i\right] + \frac{\mathrm{d}}{\mathrm{d}h}\Big[2\sum_{\substack{l=1 \\ l\neq i}}^n (\lambda_0 - h)\lambda_l(2r_{0l} - r_0 - r_l) \\ &\qquad + 2\sum_{\substack{l=1 \\ l\neq i}}^n (\lambda_i + h)\lambda_l(2r_{il} - r_i - r_l) + 2(\lambda_0 - h)(\lambda_i + h)(2r_{0i} - r_0 - r_i)\Big] \end{aligned}$$

$$
\begin{aligned}
&= r_i - r_0 - 2\sum_{\substack{l=1\\ l\neq i}}^{n} \lambda_l(2r_{0l} - r_0 - r_l) + 2\sum_{\substack{l=1\\ l\neq i}}^{n} \lambda_l(2r_{li} - r_i - r_l)\\
&\quad + 2(\lambda_0 - \lambda_i - 2h)(2r_{0i} - r_0 - r_i)\\
&= r_i - r_0 + 2(\lambda_0 - \lambda_i - 2h)(2r_{0i} - r_0 - r_i) + 2\sum_{\substack{l=1\\ l\neq i}}^{n} \Big[\lambda_l(2r_{li} - r_i - 2r_{0l} + r_0)\Big].
\end{aligned}
$$

Letting $h = 0$, we have, using $\sum_{l=0}^{n} \lambda_l = 1$, which implies that $\sum_{\substack{l=1\\ l\neq i}}^{n} \lambda_l = 1 - \lambda_0 - \lambda_i$

$$
\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}h} g_\nu\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big)\Big|_{h=0}\\
&= r_i - r_0 + 2(\lambda_0 - \lambda_i)(2r_{0i} - r_0 - r_i) + 2\sum_{\substack{l=1\\ l\neq i}}^{n} \Big[\lambda_l(2r_{li} - r_i - 2r_{0l} + r_0)\Big]\\
&= r_i - r_0 + 2(\lambda_0 - \lambda_i)2r_{0i} + 2(\lambda_0 - \lambda_i)(-r_0 - r_i)\\
&\quad + 2\sum_{\substack{l=1\\ l\neq i}}^{n} \lambda_l(r_0 - r_i) + 4\sum_{\substack{l=1\\ l\neq i}}^{n} \lambda_l(r_{li} - r_{0l})\\
&= (r_i - r_0)\Bigg[1 - 2\sum_{\substack{l=1\\ l\neq i}}^{n} \lambda_l\Bigg] + 4\lambda_0 r_{0i} - 4\lambda_i r_{0i}\\
&\quad - 2(\lambda_0 - \lambda_i)(r_0 + r_i) + 4\sum_{\substack{l=1\\ l\neq i}}^{n} \lambda_l(r_{li} - r_{0l})\\
&= (r_i - r_0)\big[1 - 2(1 - \lambda_0 - \lambda_i)\big] + 4\lambda_0 r_{0i} - 4\lambda_i r_{0i}\\
&\quad - 2(\lambda_0 - \lambda_i)(r_0 + r_i) + 4\sum_{\substack{l=1\\ l\neq i}}^{n} \lambda_l(r_{li} - r_{0l})\\
&= (r_i - r_0)(-1 + 2\lambda_0 + 2\lambda_i) + 4\lambda_0 r_{0i} - 4\lambda_i r_{0i}\\
&\quad - 2(\lambda_0 - \lambda_i)(r_0 + r_i) + 4\sum_{\substack{l=1\\ l\neq i}}^{n} \lambda_l(r_{li} - r_{0l})\\
&= r_0 - r_i + 2(\lambda_0 r_i - \lambda_0 r_0 + \lambda_i r_i - \lambda_i r_0) - 2(\lambda_0 r_0 + \lambda_0 r_i - \lambda_i r_0 - \lambda_i r_i)\\
&\quad + 4\lambda_0 r_{0i} - 4\lambda_i r_{0i} + 4\sum_{\substack{l=1\\ l\neq i}}^{n} \lambda_l(r_{li} - r_{0l})\\
&= r_0 - r_i - 4\lambda_0 r_0 + 4\lambda_i r_i + 4\lambda_0 r_{0i} - 4\lambda_i r_{0i} + 4\sum_{\substack{l=1\\ l\neq i}}^{n} \lambda_l(r_{li} - r_{0l})\\
&= r_0 - r_i + 4\Bigg[\sum_{\substack{l=1\\ l\neq i}}^{n} \lambda_l r_{li} + \lambda_0 r_{0i} + \lambda_i r_i\Bigg] - 4\Bigg[\sum_{\substack{l=1\\ l\neq i}}^{n} \lambda_l r_{0l} + \lambda_i r_{0i} + \lambda_0 r_0\Bigg]
\end{aligned}
$$

$$= r_0 - r_i + 4\left[\sum_{\substack{l=0\\l\neq i}}^{n} \lambda_l r_{li} + \lambda_i r_i\right] - 4\left[\sum_{l=1}^{n} \lambda_l r_{0l} + \lambda_0 r_0\right]$$
$$= b_i(\mathbf{x})$$

by (5). We also have by the chain rule that

$$b_i(\mathbf{x}) = \frac{\mathrm{d}}{\mathrm{d}h} g_\nu\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big)\bigg|_{h=0}$$
$$= \sum_{j=1}^{n} \frac{\partial}{\partial x_j} g_\nu\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big)(\mathbf{x}_i - \mathbf{x}_0)_j\bigg|_{h=0} = \big(\boldsymbol{\nabla} g_\nu^{\mathrm{T}}(\mathbf{x})\mathbf{X}_\nu^{\mathrm{T}}\big)_i, \tag{7}$$

where $\big(\boldsymbol{\nabla} g_\nu^{\mathrm{T}}(\mathbf{x})\mathbf{X}_\nu^{\mathrm{T}}\big)_i$ denotes the i-th column of the vector in $\mathbb{R}^{1\times n}$ and $\mathbf{X}_\nu$ is the shape matrix of the simplex (see Definition 2.4). Therefore, we obtain

$$\mathbf{b}(\mathbf{x})^{\mathrm{T}} = \boldsymbol{\nabla} g_\nu^{\mathrm{T}}(\mathbf{x})\mathbf{X}_\nu^{\mathrm{T}} \Rightarrow \boldsymbol{\nabla} g_\nu^{\mathrm{T}}(\mathbf{x}) = \mathbf{b}(\mathbf{x})^{\mathrm{T}}\mathbf{X}_\nu^{-\mathrm{T}} \Rightarrow \boldsymbol{\nabla} g_\nu(\mathbf{x}) = \mathbf{X}_\nu^{-1}\mathbf{b}(\mathbf{x}),$$

which shows (4). □

We now give a formula for the Hessian in the following proposition.

Proposition 2.11. *Let $\mathcal{T} = (\mathcal{G}_\nu)$ be a triangulation of a set $\mathcal{D}_\mathcal{T} \subset \mathbb{R}^n$ and $g : \mathcal{D}_\mathcal{T} \to \mathbb{R}$ be a CPQ function over this set. Then, for each simplex $\mathcal{G}_\nu$, the Hessian of $g_\nu := g\big|_{\mathcal{G}_\nu}$ is the constant matrix*

$$\mathbf{H}_\nu = \mathbf{X}_\nu^{-1}\tilde{\mathbf{H}}, \tag{8}$$

where $\mathbf{X}_\nu$ is the shape matrix of the simplex $\mathcal{G}_\nu$, see (1), and $\tilde{\mathbf{H}} \in \mathbb{R}^{n\times n}$ is a matrix with entries given by (9)

$$\tilde{h}_{ij} := \sum_{\substack{k=1\\k\neq i}}^{n} (\mathbf{X}_\nu^{-1})_{jk}\Big[4(r_0 + r_{ik} - r_{0i} - r_{0k})\Big] + (\mathbf{X}_\nu^{-1})_{ji}\Big[4(r_0 + r_i - 2r_{0i})\Big] \tag{9}$$

and the r_k, r_{kl} are defined by (3).

Remark 2.12. We stress that the matrices $\mathbf{H}_\nu$ and $\tilde{\mathbf{H}}$ are both independent of $\mathbf{x}$ and therefore constant, showing that the function g_ν is quadratic.

Proof. From (4), we have that

$$\frac{\partial}{\partial x_j} g_\nu(\mathbf{x}) = \sum_{k=1}^{n} (\mathbf{X}_\nu^{-1})_{jk} b_k(\mathbf{x}), \tag{10}$$

where $(\mathbf{X}_\nu^{-1})_{jk}$ is the entry in the j-th row and k-th column of $\mathbf{X}_\nu^{-1}$, and $b_k(\mathbf{x})$ is the k-th entry of $\mathbf{b}$.

From (5), we have that

$$b_k(\mathbf{x}) = r_0 - r_k + 4\sum_{l=0}^{n} \lambda_l s^k(\mathbf{x}_l) - 4\sum_{l=0}^{n} \lambda_l t(\mathbf{x}_l).$$

Now replacing $\mathbf{x}$ by $\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)$ and using (6) so that λ_0 becomes $\lambda_0 - h$ and λ_i becomes $\lambda_i + h$, we can find a formula for $b_k\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big)$ in the following way.

Case 1: $k = i$

$$b_i\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big) = r_0 - r_i + 4\Big[(\lambda_0 - h)r_{0i} + \sum_{\substack{l=1\\l\neq i}}^{n} \lambda_l r_{li} + (\lambda_i + h)r_i\Big] - 4\Big[(\lambda_0 - h)r_0 + (\lambda_i + h)r_{0i} + \sum_{\substack{l=1\\l\neq i}}^{n} \lambda_l r_{0l}\Big].$$

Therefore, when $k = i$,

$$\begin{aligned}\frac{d}{dh}\Big[b_i\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big)\Big] &= -4r_{0i} + 4r_i + 4r_0 - 4r_{0i}\\ &= 4r_0 + 4r_i - 8r_{0i}.\end{aligned}$$

Case 2: $k \neq i$

$$b_k\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big) = r_0 - r_k + 4\Big[(\lambda_0 - h)r_{0k} + (\lambda_i + h)r_{ik} + \sum_{\substack{l=1\\l\neq i\\l\neq k}}^{n} \lambda_l r_{lk} + \lambda_k r_k\Big] - 4\Big[(\lambda_0 - h)r_0 + (\lambda_i + h)r_{0i} + \sum_{\substack{l=1\\l\neq i}}^{n} \lambda_l r_{0l}\Big].$$

Therefore, when $k \neq i$,

$$\frac{d}{dh}\Big[b_k\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big)\Big] = -4r_{0k} + 4r_{ik} + 4r_0 - 4r_{0i} = 4(-r_{0k} + r_{ik} + r_0 - r_{0i}).$$

Now we take the directional derivative with respect to $(\mathbf{x}_i - \mathbf{x}_0)$, using (10):

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}h}\Big[\frac{\partial}{\partial x_j} g_\nu\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big)\Big]\\ &= \sum_{k=1}^{n} \frac{\mathrm{d}}{\mathrm{d}h}\Big[(\mathbf{X}_\nu^{-1})_{jk} b_k\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big)\Big]\\ &= \sum_{k=1}^{n} (\mathbf{X}_\nu^{-1})_{jk} \frac{\mathrm{d}}{\mathrm{d}h}\Big[b_k\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big)\Big]\\ &= \sum_{\substack{k=1\\k\neq i}}^{n} (\mathbf{X}_\nu^{-1})_{jk}\Big[4(-r_{0k} + r_{ik} + r_0 - r_{0i})\Big] + (\mathbf{X}_\nu^{-1})_{ji}(4r_0 + 4r_i - 8r_{0i}).\\ &= \tilde{h}_{ij},\end{aligned} \tag{11}$$

see (9), which is independent of h.

By the chain rule, we also have that

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}h}\Big[\frac{\partial}{\partial x_j} g_\nu\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big)\Big] &= \sum_{q=1}^{n} \frac{\partial^2}{\partial x_q \partial x_j} g_\nu\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big)(\mathbf{x}_i - \mathbf{x}_0)_q\\ &= \sum_{q=1}^{n} \Big[\mathbf{H}\big(g_\nu\big(\mathbf{x} + h(\mathbf{x}_i - \mathbf{x}_0)\big)\big)\Big]_{qj} (\mathbf{x}_i - \mathbf{x}_0)_q.\end{aligned} \tag{12}$$

Denote by h_{qj} the entry in the q-th row and j-th column of the Hessian $\mathbf{H}_\nu$. Then, from (11) and (12) and with $h \to 0$, we have that

$$\begin{aligned}\tilde{h}_{ij} = \sum_{q=1}^{n} h_{qj}(\mathbf{x}_i - \mathbf{x}_0)_q &\Rightarrow \tilde{\mathbf{H}}^{\mathrm{T}} = \mathbf{H}_\nu^{\mathrm{T}}\mathbf{X}_\nu^{\mathrm{T}} \\ &\Rightarrow \mathbf{H}_\nu^{\mathrm{T}} = \tilde{\mathbf{H}}^{\mathrm{T}}(\mathbf{X}_\nu^{-\mathrm{T}}) \\ &\Rightarrow \mathbf{H}_\nu = \mathbf{X}_\nu^{-1}\tilde{\mathbf{H}}\end{aligned}$$

which shows (8). □

3. Computation of Lyapunov functions for SDEs using CPQ interpolation.

We are interested in using LP to parameterize CPQ functions that are non-local Lyapunov functions for stochastic differential equations (SDEs). Lyapunov functions are often used to analyze the stability of a dynamical system's equilibrium as the existence of such a function cannot only imply the asymptotic stability of an equilibrium, but also provide some information regarding its basin of attraction, cf. e.g. [9, 19, 20, 21, 26, 30, 33]. The idea of non-local Lyapunov functions for SDEs is to use linearization to obtain a local Lyapunov function close to an equilibrium and then obtain stronger stability guarantees by combining it with non-local Lyapunov functions. We will not discuss the theory further here, but refer the interested reader to [15].

We first summarize some basic theory regarding SDE of Itô type from [7] and [13]. We will see that a Lyapunov function V for an SDE satisfies an inequality involving its gradient $\boldsymbol{\nabla} V(\mathbf{x})$ and Hessian $\mathbf{H}\big(V(\mathbf{x})\big)$. Since the Hessian of a CPA function is zero, CPA functions cannot be used as Lyapunov functions for an SDEs. However, we will prove that a CPQ function V can be used, provided that V satisfies certain conditions at every vertex of the triangulation. The method we use to prove this is similar to that used in the proof of Theorem 2.6 of [10], and all definitions in Subsection 3.1 are from [7]. The important part is that these conditions can be formulated as linear constraints in the variables of a linear programming (LP) problem. Hence, we can use LP to compute non-local Lyapunov functions for SDEs. The LP program is given in LP Problem 3.5 and in Theorem 3.4 we prove that a feasible solution to the LP problem delivers a non-local Lyapunov function for the SDE in question. The LP problem is a feasibility problem and the objective of the LP problem is not needed. In Example 4.2 we show how the objective can be used to force some conditions on the computed Lyapunov function.

3.1. Stochastic Differential Equation.

We first define the SDE that we are considering; note that these are differential equations whose solutions are random processes. The interested reader can find further information regarding SDEs in [22, 27, 28].

Definition 3.1 (SDE of Itô type)**.** A stochastic differential equation of Itô type is of the form

$$d\mathbf{X}(t) = \mathbf{f}\big(\mathbf{X}(t)\big)dt + \mathbf{g}\big(\mathbf{X}(t)\big)d\mathbf{W}(t), \tag{13}$$

where $\mathbf{W}(t)$ is a Q-dimensional Wiener process, and $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^{n\times Q}$ are Lipschitz continuous.

We will assume that $\mathbf{f}(\mathbf{0}) = \mathbf{g}(\mathbf{0}) = \mathbf{0}$ such that $\mathbf{X}(t)\equiv\mathbf{0}$ is an equilibrium. We want to learn more about the stability of the equilibrium $\mathbf{X}(t)\equiv\mathbf{0}$ of such systems, in particular its γ-basin of attraction which we define below.

Definition 3.2 (γ-basin of attraction)**.** Consider the stochastic system given in (13) and let $0 < \gamma \le 1$. The γ-basin of attraction is the set of all initial conditions $\mathbf{x}$ such that their trajectories will converge to the equilibrium as time tends to infinity with a probability of at least γ. It can be represented by

$$\gamma\text{-BOA} = \Big\{\mathbf{x} \in \mathbb{R}^n : \mathbb{P}\Big(\lim_{t\to\infty} \|\mathbf{X}^{\mathbf{x}}(t)\| = 0\Big) \ge \gamma\Big\},$$

where $\mathbf{X}^{\mathbf{x}}(t)$ denotes the trajectory of the SDE with initial condition $\mathbf{x}$.

Definition 3.3 (Non-local Lyapunov function for SDE)**.** A non-local Lyapunov function for a stochastic differential equation (SDE) is a function $V : U \setminus \overline{M} \subset \mathbb{R}^n \to \mathbb{R}$, $V \in C^2(U \setminus \overline{M})$, where U and M are neighbourhoods of the origin, $U \supset \overline{M}$, and $\overline{M}$ denotes the closure of M, satisfying

$$LV(\mathbf{x}) := \boldsymbol{\nabla} V(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}) + \frac{1}{2}\sum_{i,j=1}^{n} \Big[\mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^{\mathrm{T}}\Big]_{ij} \Big[\mathbf{H}\big(V(\mathbf{x})\big)\Big]_{ij} < 0,$$

$\forall \mathbf{x} \in U \setminus \overline{M}$, where $\mathbf{H}\big(V(\mathbf{x})\big)$ is the Hessian of V at $\mathbf{x}$.

Typically U is a large neighbourhood and M a small one. The stability properties of solution trajectories in M are taken care of by a local Lyapunov function that can be computed by linearizing $\mathbf{f}$ and $\mathbf{g}$ around the origin [6]. The stability properties of solution trajectories in U are then taken care of by the non-local Lyapunov function [15]. We will only discuss the non-local Lyapunov function here. If a local Lyapunov function is given and $LV(\mathbf{x}) < 0$ for all $\mathbf{x} \in U \setminus \overline{M}$, then it follows by Theorem 2.5 of [7] that we can find a subset of the γ-basin of attraction of the origin. Therefore, we are interested in determining what conditions we must impose on a CPQ function V such that $LV(\mathbf{x}) \le -C$, where C is some positive constant. Then, by Definition 3.3, V is a non-local Lyapunov function for the SDE.

To enhance the readability we first derive several estimates in the next section, before we state our LP problem in Section 3.4 and prove that a feasible solution to it delivers a non-local Lyapunov function in Theorem 3.4.

3.2. **Some useful estimates for the LP problem.** Consider the stochastic differential equation given in (13). We want to determine under what conditions a CPQ function $V : \overline{U \setminus \overline{M}} \subset \mathbb{R}^n \to \mathbb{R}$ is a non-local Lyapunov function for the SDE. Assume $\overline{U \setminus \overline{M}}$ is triangulated by $\mathcal{T}$ and let us consider an arbitrary, but fixed simplex $\mathcal{G}_\nu \in \mathcal{T}$. To derive error estimates we first consider a quadratic function $V : \mathcal{G}_\nu \to \mathbb{R}$. Denote $F_1(\mathbf{x}) := \boldsymbol{\nabla} V(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x})$ and $F_2(\mathbf{x}) := \sum_{i,j=1}^{n} \big[\mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^{\mathrm{T}}\big]_{ij} \big[\mathbf{H}\big(V(\mathbf{x})\big)\big]_{ij}$. Then $LV(\mathbf{x}) = F_1(\mathbf{x}) + \frac{1}{2}F_2(\mathbf{x})$. We know we can write any $\mathbf{x} \in \mathcal{G}_\nu$ as $\mathbf{x} = \sum_{k=0}^{n} \lambda_k \mathbf{x}_k$ where the $\mathbf{x}_k$ are the vertices of $\mathcal{G}_\nu$, $\lambda_k \ge 0$ and $\sum_{k=0}^{n} \lambda_k = 1$. Therefore,

$$\begin{aligned}
F_1(\mathbf{x}) + \frac{1}{2}F_2(\mathbf{x}) &= F_1\left(\sum_{k=0}^{n} \lambda_k \mathbf{x}_k\right) + \frac{1}{2}F_2\left(\sum_{k=0}^{n} \lambda_k \mathbf{x}_k\right) \\
&= \sum_{k=0}^{n} \lambda_k F_1(\mathbf{x}_k) + F_1\left(\sum_{k=0}^{n} \lambda_k \mathbf{x}_k\right) - \sum_{k=0}^{n} \lambda_k F_1(\mathbf{x}_k) \\
&\quad + \frac{1}{2}\sum_{k=0}^{n} \lambda_k F_2(\mathbf{x}_k) + \frac{1}{2}F_2\left(\sum_{k=0}^{n} \lambda_k \mathbf{x}_k\right) - \frac{1}{2}\sum_{k=0}^{n} \lambda_k F_2(\mathbf{x}_k)
\end{aligned}$$

$$\leq \sum_{k=0}^{n} \lambda_k F_1(\mathbf{x}_k) + \left| F_1\left(\sum_{k=0}^{n} \lambda_k \mathbf{x}_k\right) - \sum_{k=0}^{n} \lambda_k F_1(\mathbf{x}_k) \right|$$
$$+ \frac{1}{2}\sum_{k=0}^{n} \lambda_k F_2(\mathbf{x}_k) + \frac{1}{2}\left| F_2\left(\sum_{k=0}^{n} \lambda_k \mathbf{x}_k\right) - \sum_{k=0}^{n} \lambda_k F_2(\mathbf{x}_k) \right|.$$

Note that we have added and subtracted the CPA approximations of $F_1(\mathbf{x})$ and $F_2(\mathbf{x})$ in order to bound $LV(\mathbf{x})$ above by the CPA error estimates that we calculated in Section 2.1.

Recall that by Lemma 2.7 we have $\left| F_j\big(\sum_{k=0}^{n} \lambda_k \mathbf{x}_k\big) - \sum_{k=0}^{n} \lambda_k F_j(\mathbf{x}_k) \right| \leq h_\nu^2 B_j$, where $h_\nu = \max_{\mathbf{x}_i \in \mathcal{G}_\nu} \|\mathbf{x}_i - \mathbf{x}_0\|_2$, and $B_j = \max_{\mathbf{w} \in \mathcal{G}_\nu} \left\| \mathbf{H}\big(F_j(\mathbf{w})\big) \right\|_2$ for $j = 1, 2$, where $\mathbf{H}\big(F_j(\mathbf{w})\big)$ is the Hessian of F_j at $\mathbf{w}$. To simplify calculations, we bound the matrix norm $\|A\|_2 = \max_{\|\mathbf{x}\|_2 = 1} \|A\mathbf{x}\|_2$ above by the max matrix norm $\|A\|_{\max} := \max_{i,j=1,\dots,n} |a_{ij}|$. Then,

$$\max_{\mathbf{w} \in \mathcal{G}_\nu} \left\| \mathbf{H}\big(F_j(\mathbf{w})\big) \right\|_2 \leq n \max_{\mathbf{w} \in \mathcal{G}_\nu} \left\| \mathbf{H}\big(F_j(\mathbf{w})\big) \right\|_{\max}$$

and we obtain,

$$\begin{aligned} F_1(\mathbf{x}) + \frac{1}{2}F_2(\mathbf{x}) &\leq \sum_{k=0}^{n} \lambda_k \Big[F_1(\mathbf{x}_k) + \frac{1}{2}F_2(\mathbf{x}_k) \Big] + h_\nu^2 \Big(B_1 + \frac{B_2}{2} \Big) \\ &\leq \sum_{k=0}^{n} \lambda_k \Big[F_1(\mathbf{x}_k) + \frac{1}{2}F_2(\mathbf{x}_k) \Big] + h_\nu^2 n \max_{\mathbf{w} \in \mathcal{G}} \left\| \mathbf{H}\big(F_1(\mathbf{w})\big) \right\|_{\max} \\ &\quad + \frac{1}{2} h_\nu^2 n \max_{\mathbf{w} \in \mathcal{G}} \left\| \mathbf{H}\big(F_2(\mathbf{w})\big) \right\|_{\max}. \end{aligned} \tag{14}$$

We now calculate upper bounds on $\max_{\mathbf{x} \in \mathcal{G}_\nu} \left\| \mathbf{H}\big(F_j(\mathbf{x})\big) \right\|_{\max}$ for $j = 1, 2$.

<u>Calculating a bound on $\max_{\mathbf{x} \in \mathcal{G}_\nu} \left\| \mathbf{H}\big(F_1(\mathbf{x})\big) \right\|_{\max}$:</u>

The (i,j)-th entry of $\mathbf{H}\big(F_1(\mathbf{x})\big) \in \mathbb{R}^{n \times n}$ is given by $\frac{\partial^2 F_1(\mathbf{x})}{\partial x_i \partial x_j}$. For fixed $i, j \in \{1, \dots, n\}$, we have for any $\mathbf{x} \in \mathcal{G}_\nu$

$$\begin{aligned} \frac{\partial^2 F_1(\mathbf{x})}{\partial x_i \partial x_j} &= \frac{\partial^2}{\partial x_i \partial x_j} \Big(\boldsymbol{\nabla} V(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}) \Big) \\ &= \frac{\partial^2}{\partial x_i \partial x_j} \left(\sum_{k=1}^{n} \frac{\partial V(\mathbf{x})}{\partial x_k} \mathbf{f}_k(\mathbf{x}) \right) \\ &= \frac{\partial}{\partial x_i} \left[\sum_{k=1}^{n} \frac{\partial V(\mathbf{x})}{\partial x_k} \frac{\partial \mathbf{f}_k(\mathbf{x})}{\partial x_j} + \frac{\partial^2 V(\mathbf{x})}{\partial x_j \partial x_k} \mathbf{f}_k(\mathbf{x}) \right] \\ &= \sum_{k=1}^{n} \left(\frac{\partial V(\mathbf{x})}{\partial x_k} \frac{\partial^2 \mathbf{f}_k(\mathbf{x})}{\partial x_i \partial x_j} + \frac{\partial^2 V(\mathbf{x})}{\partial x_i \partial x_k} \frac{\partial \mathbf{f}_k(\mathbf{x})}{\partial x_j} + \frac{\partial^2 V(\mathbf{x})}{\partial x_j \partial x_k} \frac{\partial \mathbf{f}_k(\mathbf{x})}{\partial x_i} \right), \end{aligned}$$

where we have used that V is a quadratic function on $\mathcal{G}_\nu$.

Then, with $h_\nu^\infty := \max_{\mathbf{x} \in \mathcal{G}_\nu} \|\mathbf{x} - \mathbf{x}_0\|_\infty$, we get

$$\max_{\mathbf{x} \in \mathcal{G}_\nu} \left\| \mathbf{H}\big(F_1(\mathbf{x})\big) \right\|_{\max}$$

$$
\begin{aligned}
&= \max_{\mathbf{x}\in\mathcal{G}_\nu}\left[\max_{i,j=1\dots,n}\left|\sum_{k=1}^{n}\frac{\partial V(\mathbf{x})}{\partial x_k}\frac{\partial^2\mathbf{f}_k(\mathbf{x})}{\partial x_i\partial x_j}+\frac{\partial^2 V(\mathbf{x})}{\partial x_i\partial x_k}\frac{\partial\mathbf{f}_k(\mathbf{x})}{\partial x_j}+\frac{\partial^2 V(\mathbf{x})}{\partial x_j\partial x_k}\frac{\partial\mathbf{f}_k(\mathbf{x})}{\partial x_i}\right|\right]\\
&\le \max_{\mathbf{x}\in\mathcal{G}_\nu}\left[\sum_{k=1}^{n}\max_{i,j=1,\dots,n}\left|\frac{\partial V(\mathbf{x})}{\partial x_k}\frac{\partial^2\mathbf{f}_k(\mathbf{x})}{\partial x_i\partial x_j}+\frac{\partial^2 V(\mathbf{x})}{\partial x_i\partial x_k}\frac{\partial\mathbf{f}_k(\mathbf{x})}{\partial x_j}+\frac{\partial^2 V(\mathbf{x})}{\partial x_j\partial x_k}\frac{\partial\mathbf{f}_k(\mathbf{x})}{\partial x_i}\right|\right]\\
&\le \max_{\mathbf{x}\in\mathcal{G}_\nu}\sum_{k=1}^{n}\left[\left(\max_{p=1,\dots,n}\left|\frac{\partial V(\mathbf{x})}{\partial x_p}\right|\right)\left(\max_{i,j,p=1,\dots,n}\left|\left[\mathbf{H}\big(\mathbf{f}_p(\mathbf{x})\big)\right]_{ij}\right|\right)\right.\\
&\qquad +\left(\max_{i,p=1\dots,n}\left|\left[\mathbf{H}\big(V(\mathbf{x})\big)\right]_{ip}\right|\right)\left(\max_{j,p=1,\dots,n}\left|\left[D\mathbf{f}_p(\mathbf{x})\right]_{j}\right|\right)\\
&\qquad \left.+\left(\max_{j,p=1\dots,n}\left|\left[\mathbf{H}\big(V(\mathbf{x})\big)\right]_{jp}\right|\right)\left(\max_{i,p=1,\dots,n}\left|\left[D\mathbf{f}_p(\mathbf{x})\right]_{i}\right|\right)\right]\\
&\le \max_{\mathbf{x}\in\mathcal{G}_\nu}\sum_{k=1}^{n}\left[\left\|\boldsymbol{\nabla}V(\mathbf{x})\right\|_\infty\cdot\max_{p=1,\dots,n}\left\|\mathbf{H}\big(\mathbf{f}_p(\mathbf{x})\big)\right\|_{\max}\right.\\
&\qquad \left.+2\left\|\mathbf{H}\big(V(\mathbf{x})\big)\right\|_{\max}\cdot\left\|D\mathbf{f}(\mathbf{x})\right\|_{\max}\right]\\
&\le n\left[\max_{\mathbf{x}\in\mathcal{G}_\nu}\left\|\boldsymbol{\nabla}V(\mathbf{x})\right\|_\infty\cdot\max_{\mathbf{x}\in\mathcal{G}_\nu}\left(\max_{p=1,\dots,n}\left\|\mathbf{H}\big(\mathbf{f}_p(\mathbf{x})\big)\right\|_{\max}\right)\right.\\
&\qquad \left.+2\max_{\mathbf{x}\in\mathcal{G}_\nu}\left\|\mathbf{H}\big(V(\mathbf{x})\big)\right\|_{\max}\cdot\max_{\mathbf{x}\in\mathcal{G}_\nu}\left(\left\|D\mathbf{f}(\mathbf{x})\right\|_{\max}\right)\right]\\
&\le n\left[\left(\left\|\boldsymbol{\nabla}V(\mathbf{x}_0)\right\|_\infty+nh_\nu^\infty\max_{\mathbf{x}\in\mathcal{G}_\nu}\left\|\mathbf{H}\big(V(\mathbf{x})\big)\right\|_{\max}\right)\cdot\max_{\mathbf{x}\in\mathcal{G}_\nu}\left(\max_{p=1,\dots,n}\left\|\mathbf{H}\big(\mathbf{f}_p(\mathbf{x})\big)\right\|_{\max}\right)\right.\\
&\qquad \left.+2\max_{\mathbf{x}\in\mathcal{G}_\nu}\left\|\mathbf{H}\big(V(\mathbf{x})\big)\right\|_{\max}\cdot\max_{\mathbf{x}\in\mathcal{G}_\nu}\left\|D\mathbf{f}(\mathbf{x})\right\|_{\max}\right]\\
&=: E_1^\nu, \qquad (15)
\end{aligned}
$$

where $\mathbf{f}_p(\mathbf{x})$ is the p-th entry of the vector-valued function $\mathbf{f}$, $\mathbf{H}\big(\mathbf{f}_p(\mathbf{x})\big)$ is the Hessian of the p-th entry of $\mathbf{f}$, and $D\mathbf{f}_p(\mathbf{x})$ is the vector whose j-th entry is $\left[D\mathbf{f}_p(\mathbf{x})\right]_j = \frac{\partial\mathbf{f}_p(\mathbf{x})}{\partial x_j}$. We also used that

$$
\begin{aligned}
\left\|\boldsymbol{\nabla}V(\mathbf{x})\right\|_\infty-\left\|\boldsymbol{\nabla}V(\mathbf{x}_0)\right\|_\infty &\le \left\|\boldsymbol{\nabla}V(\mathbf{x})-\boldsymbol{\nabla}V(\mathbf{x}_0)\right\|_\infty\\
&=\left\|\int_0^1\mathbf{H}\big(V(\mathbf{x}_0+t[\mathbf{x}-\mathbf{x}_0])\big)\cdot[\mathbf{x}-\mathbf{x}_0]dt\right\|_\infty\\
&\le h_\nu^\infty\max_{\mathbf{w}\in\mathcal{G}_\nu}\left\|\mathbf{H}\big(V(\mathbf{w})\big)\right\|_\infty\\
&\le nh_\nu^\infty\max_{\mathbf{w}\in\mathcal{G}_\nu}\left\|\mathbf{H}\big(V(\mathbf{w})\big)\right\|_{\max}.
\end{aligned}
$$

Calculating a bound on $\max_{\mathbf{x}\in\mathcal{G}_\nu}\left\|\mathbf{H}\big(F_2(\mathbf{x})\big)\right\|_{\max}$:

We note that $\mathbf{H}\big(F_2(\mathbf{x})\big)\in\mathbb{R}^{n\times n}$ where its (l,m)-th entry is $\frac{\partial^2 F_2(\mathbf{x})}{\partial x_l\partial x_m}$. Recall from

Remark 2.12 that $\mathbf{H}\big(V(\mathbf{x})\big)$ is constant and therefore its derivatives are zero. Then, for fixed $l, m \in \{1, \dots, n\}$, we have for any $\mathbf{x} \in \mathcal{G}_\nu$ that

$$\begin{aligned}
\frac{\partial^2 F_2(\mathbf{x})}{\partial x_l \partial x_m} &= \frac{\partial^2}{\partial x_l \partial x_m}\left(\sum_{i,j=1}^{n}\Big[\mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^{\mathrm{T}}\Big]_{ij}\Big[\mathbf{H}\big(V(\mathbf{x})\big)\Big]_{ij}\right)\\
&= \frac{\partial^2}{\partial x_l \partial x_m}\left(\sum_{i,j=1}^{n}\sum_{k=1}^{Q}\mathbf{g}_{ik}(\mathbf{x})\mathbf{g}_{jk}(\mathbf{x})\frac{\partial^2 V(\mathbf{x})}{\partial x_i \partial x_j}\right)\\
&= \frac{\partial}{\partial x_l}\sum_{i,j=1}^{n}\sum_{k=1}^{Q}\left(\frac{\partial \mathbf{g}_{ik}(\mathbf{x})}{\partial x_m}\mathbf{g}_{jk}(\mathbf{x}) + \mathbf{g}_{ik}(\mathbf{x})\frac{\partial \mathbf{g}_{jk}(\mathbf{x})}{\partial x_m}\right)\frac{\partial^2 V(\mathbf{x})}{\partial x_i \partial x_j}\\
&= \sum_{i,j=1}^{n}\sum_{k=1}^{Q}\left(\frac{\partial^2 \mathbf{g}_{ik}(\mathbf{x})}{\partial x_l \partial x_m}\mathbf{g}_{jk}(\mathbf{x}) + \frac{\partial \mathbf{g}_{ik}(\mathbf{x})}{\partial x_m}\frac{\partial \mathbf{g}_{jk}(\mathbf{x})}{\partial x_l}\right.\\
&\qquad\qquad \left. + \frac{\partial \mathbf{g}_{ik}(\mathbf{x})}{\partial x_l}\frac{\partial \mathbf{g}_{jk}(\mathbf{x})}{\partial x_m} + \mathbf{g}_{ik}(\mathbf{x})\frac{\partial^2 \mathbf{g}_{jk}(\mathbf{x})}{\partial x_l \partial x_m}\right)\frac{\partial^2 V(\mathbf{x})}{\partial x_i \partial x_j}.
\end{aligned}$$

Then,

$$\begin{aligned}
&\max_{\mathbf{x}\in\mathcal{G}_\nu}\Big\|\mathbf{H}\big(F_2(\mathbf{x})\big)\Big\|_{\max}\\
&= \max_{\mathbf{x}\in\mathcal{G}_\nu}\left(\max_{l,m=1,\dots,n}\left|\sum_{i,j=1}^{n}\sum_{k=1}^{Q}\left(\frac{\partial^2 \mathbf{g}_{ik}(\mathbf{x})}{\partial x_l \partial x_m}\mathbf{g}_{jk}(\mathbf{x}) + \frac{\partial \mathbf{g}_{ik}(\mathbf{x})}{\partial x_m}\frac{\partial \mathbf{g}_{jk}(\mathbf{x})}{\partial x_l}\right.\right.\right.\\
&\qquad\left.\left.\left. + \frac{\partial \mathbf{g}_{ik}(\mathbf{x})}{\partial x_l}\frac{\partial \mathbf{g}_{jk}(\mathbf{x})}{\partial x_m} + \mathbf{g}_{ik}(\mathbf{x})\frac{\partial^2 \mathbf{g}_{jk}(\mathbf{x})}{\partial x_l \partial x_m}\right)\frac{\partial^2 V(\mathbf{x})}{\partial x_i \partial x_j}\right|\right)\\
&\le \max_{\mathbf{x}\in\mathcal{G}_\nu}\left[\sum_{i,j=1}^{n}\sum_{k=1}^{Q}\left|\frac{\partial^2 V(\mathbf{x})}{\partial x_i \partial x_j}\right| \cdot \max_{l,m=1,\dots,n}\left|\frac{\partial^2 \mathbf{g}_{ik}(\mathbf{x})}{\partial x_l \partial x_m}\mathbf{g}_{jk}(\mathbf{x}) + \frac{\partial \mathbf{g}_{ik}(\mathbf{x})}{\partial x_m}\frac{\partial \mathbf{g}_{jk}(\mathbf{x})}{\partial x_l}\right.\right.\\
&\qquad\left.\left. + \frac{\partial \mathbf{g}_{ik}(\mathbf{x})}{\partial x_l}\frac{\partial \mathbf{g}_{jk}(\mathbf{x})}{\partial x_m} + \mathbf{g}_{ik}(\mathbf{x})\frac{\partial^2 \mathbf{g}_{jk}(\mathbf{x})}{\partial x_l \partial x_m}\right|\right]\\
&\le \max_{\mathbf{x}\in\mathcal{G}_\nu}\sum_{i,j=1}^{n}\sum_{k=1}^{Q}\max_{p,r=1,\dots,n}\left|\Big[\mathbf{H}\big(V(\mathbf{x})\big)\Big]_{pr}\right| \cdot\\
&\quad\left[\left(\max_{\substack{p,l,m=1,\dots,n\\ q=1,\dots,Q}}\left|\Big[\mathbf{H}\big(\mathbf{g}_{pq}(\mathbf{x})\big)\Big]_{lm}\right|\right)\left(\max_{\substack{r=1,\dots,n\\ q=1,\dots,Q}}\left|\mathbf{g}_{rq}(\mathbf{x})\right|\right)\right.\\
&\quad + \left(\max_{\substack{p,m=1,\dots,n\\ q=1,\dots,Q}}\left|\Big[\boldsymbol{\nabla}\mathbf{g}_{pq}(\mathbf{x})\Big]_{m}\right|\right)\left(\max_{\substack{r,l=1,\dots,n\\ q=1,\dots,Q}}\left|\Big[\boldsymbol{\nabla}\mathbf{g}_{rq}(\mathbf{x})\Big]_{l}\right|\right)\\
&\quad + \left(\max_{\substack{p,l=1,\dots,n\\ q=1,\dots,Q}}\left|\Big[\boldsymbol{\nabla}\mathbf{g}_{pq}(\mathbf{x})\Big]_{l}\right|\right)\left(\max_{\substack{r,m=1,\dots,n\\ q=1,\dots,Q}}\left|\Big[\boldsymbol{\nabla}\mathbf{g}_{rq}(\mathbf{x})\Big]_{m}\right|\right)\\
&\quad \left. + \left(\max_{\substack{p=1,\dots,n\\ q=1,\dots,Q}}\left|\mathbf{g}_{pq}(\mathbf{x})\right|\right)\left(\max_{\substack{r,l,m=1,\dots,n\\ q=1,\dots,Q}}\left|\Big[\mathbf{H}\big(\mathbf{g}_{rq}(\mathbf{x})\big)\Big]_{lm}\right|\right)\right]
\end{aligned}$$

$$= \max_{\mathbf{x}\in\mathcal{G}_\nu}\left[\sum_{i,j=1}^{n}\sum_{k=1}^{Q}\left\|\mathbf{H}\big(V(\mathbf{x})\big)\right\|_{\max}\left(2\max_{\substack{p=1,\ldots,n\\ q=1,\ldots,Q}}\left\|\mathbf{H}\big(\mathbf{g}_{pq}(\mathbf{x})\big)\right\|_{\max}\left\|\mathbf{g}(\mathbf{x})\right\|_{\max}\right.\right.$$
$$\left.\left.+\,2\max_{\substack{p=1,\ldots,n\\ q=1,\ldots,Q}}\left\|\boldsymbol{\nabla}\mathbf{g}_{pq}(\mathbf{x})\right\|_\infty^2\right)\right]$$
$$\leq 2n^2Q\max_{\mathbf{x}\in\mathcal{G}_\nu}\left\|\mathbf{H}\big(V(\mathbf{x})\big)\right\|_{\max}\left[\max_{\mathbf{x}\in\mathcal{G}_\nu}\left(\max_{\substack{p=1,\ldots,n\\ q=1,\ldots,Q}}\left\|\mathbf{H}\big(\mathbf{g}_{pq}(\mathbf{x})\big)\right\|_{\max}\right)\max_{\mathbf{x}\in\mathcal{G}_\nu}\left\|\mathbf{g}(\mathbf{x})\right\|_{\max}\right.$$
$$\left.+\max_{\mathbf{x}\in\mathcal{G}_\nu}\left(\max_{\substack{p=1,\ldots,n\\ q=1,\ldots,Q}}\left\|\boldsymbol{\nabla}\mathbf{g}_{pq}(\mathbf{x})\right\|_\infty^2\right)\right] =: E_2^\nu, \tag{16}$$

where $\mathbf{g}_{pq}(\mathbf{x})$ is the (p,q)-th entry of the matrix-valued function $\mathbf{g}$, $\mathbf{H}\big(\mathbf{g}_{pq}(\mathbf{x})\big)$ is the Hessian of the (p,q)-th entry of $\mathbf{g}$, and $\boldsymbol{\nabla}\mathbf{g}_{pq}(\mathbf{x})$ is the vector whose m-th entry is $\big[\boldsymbol{\nabla}\mathbf{g}_{pq}(\mathbf{x})\big]_m = \frac{\partial \mathbf{g}_{pq}(\mathbf{x})}{\partial x_m}$.

Substituting (15) and (16) into (14), we get

$$F_1(\mathbf{x}) + \frac{1}{2}F_2(\mathbf{x}) \leq \sum_{k=0}^{n}\lambda_k\Big[F_1(\mathbf{x}_k)+\frac{1}{2}F_2(\mathbf{x}_k)\Big] + nh_\nu^2\left(E_1^\nu + \frac{1}{2}E_2^\nu\right).$$

Denote $E^\nu := nh_\nu^2\big(E_1^\nu + E_2^\nu/2\big)$. Assume that $LV(\mathbf{x}_k) = F_1(\mathbf{x}_k) + \frac{1}{2}F_2(\mathbf{x}_k) + E^\nu \leq -C$ for every vertex $\mathbf{x}_k$ of $\mathcal{G}_\nu$, where C is some positive constant. Then

$$F_1(\mathbf{x}) + \frac{1}{2}F_2(\mathbf{x}) \leq \sum_{k=0}^{n}\lambda_k\Big[F_1(\mathbf{x}_k)+\frac{1}{2}F_2(\mathbf{x}_k)\Big] + E^\nu \leq \sum_{k=0}^{n}(-\lambda_k)C = -C < 0$$

for every $\mathbf{x}\in\mathcal{G}_\nu$.

We now consider how to implement the error term E^ν as linear constraints in the values of V at the vertices $\mathbf{x}_k$. Let us first rewrite

$$E^\nu = \left\|\boldsymbol{\nabla}V(\mathbf{x}_0)\right\|_\infty \cdot n^2h_\nu^2\max_{\mathbf{x}\in\mathcal{G}_\nu}\left(\max_{p=1,\ldots,n}\left\|\mathbf{H}\big(\mathbf{f}_p(\mathbf{x})\big)\right\|_{\max}\right)$$
$$+\max_{\mathbf{x}\in\mathcal{G}_\nu}\left\|\mathbf{H}\big(V(\mathbf{x})\big)\right\|_{\max}\cdot n^2h_\nu^2\left\{nQ\left[\max_{\mathbf{x}\in\mathcal{G}_\nu}\left(\max_{\substack{p=1,\ldots,n\\ q=1,\ldots,Q}}\left\|\mathbf{H}\big(\mathbf{g}_{pq}(\mathbf{x})\big)\right\|_{\max}\right)\max_{\mathbf{x}\in\mathcal{G}_\nu}\left\|\mathbf{g}(\mathbf{x})\right\|_{\max}\right.\right.$$
$$\left.+\max_{\mathbf{x}\in\mathcal{G}_\nu}\left(\max_{\substack{p=1,\ldots,n\\ q=1,\ldots,Q}}\left\|\boldsymbol{\nabla}\mathbf{g}_{pq}(\mathbf{x})\right\|_\infty^2\right)\right] + nh_\nu^\infty\max_{\mathbf{x}\in\mathcal{G}_\nu}\left(\max_{p=1,\ldots,n}\left\|\mathbf{H}\big(\mathbf{f}_p(\mathbf{x})\big)\right\|_{\max}\right)$$
$$\left.\left.+\,2\max_{\mathbf{x}\in\mathcal{G}_\nu}\left\|D\mathbf{f}(\mathbf{x})\right\|_{\max}\right]\right\}. \tag{17}$$

Now note, that $\boldsymbol{\nabla}V(\mathbf{x}_0)$ is a vector, whose components are linear in the values of V at $\mathbf{x}_k$ and $\mathbf{x}_{lk}$ by Proposition 2.10 and $\mathbf{H}^\nu := \mathbf{H}\big(V(\mathbf{x})\big)$ is a symmetric $n\times n$ matrix independent of $\mathbf{x}\in\mathcal{G}_\nu$, whose entries are also linear in the values of V at the vertices $\mathbf{x}_k$ and $\mathbf{x}_{lk}$ by Proposition 2.11. All the other terms in (17) are constants that can be computed or bounded by the problem data, i.e. the functions $\mathbf{f}$ and $\mathbf{g}$ and the simplex $\mathcal{G}_\nu$.

The maximum of absolute values is easily modelled by linear constrains, i.e.

$$\max\{|a_1|,|a_2|,\ldots,|a_k|\}\leq A \quad\Leftrightarrow\quad -A\leq a_i\leq A \ \text{ for } i=1,2,\ldots,k.$$

Hence, if N_ν and P_ν are variables such that

$$\big\|\boldsymbol{\nabla} V(\mathbf{x}_0)\big\|_\infty \le N_\nu \quad \text{and} \quad \big\|\mathbf{H}^\nu\big\|_{\max} \le P_\nu \tag{18}$$

and C_ν^1 and C_ν^2 are constants such that

$$C_1^\nu \ge n^2 h_\nu^2 \max_{\mathbf{x}\in\mathcal{G}_\nu}\left(\max_{p=1,\dots,n}\Big\|\mathbf{H}\big(\mathbf{f}_p(\mathbf{x})\big)\Big\|_{\max}\right) \tag{19}$$

and

$$\begin{aligned} C_2^\nu \ge n^2 h_\nu^2 \Bigg\{ nQ\Bigg[&\max_{\mathbf{x}\in\mathcal{G}_\nu}\left(\max_{\substack{p=1,\dots,n\\ q=1,\dots,Q}}\Big\|\mathbf{H}\big(\mathbf{g}_{pq}(\mathbf{x})\big)\Big\|_{\max}\right)\max_{\mathbf{x}\in\mathcal{G}_\nu}\Big\|\mathbf{g}(\mathbf{x})\Big\|_{\max} \\ &+ \max_{\mathbf{x}\in\mathcal{G}_\nu}\left(\max_{\substack{p=1,\dots,n\\ q=1,\dots,Q}}\Big\|\boldsymbol{\nabla}\mathbf{g}_{pq}(\mathbf{x})\Big\|_\infty^2\right)\Bigg] + nh_\nu^\infty \max_{\mathbf{x}\in\mathcal{G}_\nu}\left(\max_{p=1,\dots,n}\Big\|\mathbf{H}\big(\mathbf{f}_p(\mathbf{x})\big)\Big\|_{\max}\right) \\ &+ 2\max_{\mathbf{x}\in\mathcal{G}_\nu}\Big\|D\mathbf{f}(\mathbf{x})\Big\|_{\max}\Bigg]\Bigg\}, \end{aligned} \tag{20}$$

then

$$\boldsymbol{\nabla} V(\mathbf{x}_k)\cdot\mathbf{f}(\mathbf{x}_k) + \frac{1}{2}\sum_{i,j=1}^n \big[\mathbf{g}(\mathbf{x}_k)\mathbf{g}(\mathbf{x}_k)^{\mathrm{T}}\big]_{ij}\mathbf{H}_{ij}^\nu + C_1^\nu N_\nu + C_2^\nu P_\nu \le -C \tag{21}$$

for every vertex $\mathbf{x}_k$ of $\mathcal{G}_\nu$ implies that

$$\boldsymbol{\nabla} V(\mathbf{x})\cdot\mathbf{f}(\mathbf{x}) + \frac{1}{2}\sum_{i,j=1}^n \big[\mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^{\mathrm{T}}\big]_{ij}\big[\mathbf{H}\big(V(\mathbf{x})\big)\big]_{ij} \le -C \tag{22}$$

for all $\mathbf{x}\in\mathcal{G}_\nu$. In the next section we assume we have a CPQ function V defined on $\mathcal{D}_\mathcal{T} = \overline{U\setminus\overline{M}}$, such that its restriction $V\big|_{\mathcal{G}_\nu}$ to each $\mathcal{G}_\nu\in\mathcal{T}$ fulfills (21), and we will show that if ∇V is continuous, this implies for every compact $K\subset(\mathcal{D}_\mathcal{T})^\circ$ and every $\delta>0$ the existence of a non-local Lyapunov function $V_\varepsilon : K\to\mathbb{R}$, such that $|V_\varepsilon(\mathbf{x}) - V(\mathbf{x})| < \delta$ for all $\mathbf{x}\in K$.

3.3. **Non-local Lyapunov function from a CPQ function.** Assume V is a CPQ function defined on $\mathcal{D}_\mathcal{T} = \overline{U\setminus\overline{M}}$ and for every $\mathcal{G}_\nu\in\mathcal{T}$ denote by $V_\nu\colon\mathcal{G}_\nu\to\mathbb{R}$ its restriction $V\big|_{\mathcal{G}_\nu}$ to $\mathcal{G}_\nu$. Then, by (5) of Proposition 2.10, the components of $\nabla V_\nu(\mathbf{x})$ are affine for every $\mathcal{G}_\nu\in\mathcal{T}$. It follows that if

$$\nabla V_\nu(\mathbf{x}_i^\nu) = \nabla V_\mu(\mathbf{x}_j^\mu), \tag{23}$$

whenever $\mathbf{x}_i^\nu = \mathbf{x}_j^\mu$ is a vertex of both $\mathcal{G}_\nu$ and $\mathcal{G}_\mu$ in $\mathcal{T}$, then $\nabla V\colon\mathcal{D}_\mathcal{T}\to\mathbb{R}^n$, $\nabla V(\mathbf{x}) := \nabla V_\nu(\mathbf{x})$ if $\mathbf{x}\in\mathcal{G}_\nu$, is a well-defined and continuous function. Indeed, each of its components $[\nabla V]_i$, $i=1,\dots,n$, is a CPA function on the triangulation $\mathcal{T}$. Hence, we can force continuity of ∇V by using the constraints (23) in an LP program. We come to the main theorem of this section:

Theorem 3.4. *Assume V is a CPQ function defined on $\mathcal{D}_\mathcal{T} = \overline{U\setminus\overline{M}}$ that fulfills (23) and such that its restriction $V\big|_{\mathcal{G}_\nu}$ to each $\mathcal{G}_\nu\in\mathcal{T}$ fulfills (21). Let $\delta>0$ and $K\subset(\mathcal{D}_\mathcal{T})^\circ$ be a compact set. Then, for all sufficiently small $\varepsilon>0$, there exists a non-local Lyapunov function $V_\varepsilon\colon\mathcal{D}_\mathcal{T}^{-\varepsilon}\to\mathbb{R}$ such that $|V_\varepsilon(\mathbf{x}) - V(\mathbf{x})| < \delta$ for every $\mathbf{x}\in\mathcal{D}_\mathcal{T}^{-\varepsilon}$, where*

$$\mathcal{D}_\mathcal{T}^{-\varepsilon} := \{\mathbf{x}\in\mathcal{D}_\mathcal{T}\colon B_\varepsilon(\mathbf{x})\subset\mathcal{D}_\mathcal{T}\}\supset K. \tag{24}$$

Proof. Define $\phi : \mathbb{R}^n \to \mathbb{R}_0^+$, $\phi(\mathbf{x}) := C^* \exp(-1/(1-\|\mathbf{x}\|_2))$ for $\|\mathbf{x}\|_2 < 1$ and $\phi(\mathbf{x}) := 0$ otherwise and choose the constant C^* such that $\int_{\mathbb{R}^n} \phi(\mathbf{y})\, \mathrm{d}\mathbf{y} = 1$. For $\varepsilon > 0$ define

$$\widetilde{\phi}_\varepsilon(\mathbf{x}) := \frac{\phi(\mathbf{x}/\varepsilon)}{\varepsilon^n}.$$

Define $V_\varepsilon := V * \widetilde{\phi}_\varepsilon$, i.e., $V_\varepsilon(\mathbf{x}) = \int_{\mathcal{D}_\mathcal{T}} V(\mathbf{y})\widetilde{\phi}_\varepsilon(\mathbf{x}-\mathbf{y})\, \mathrm{d}\mathbf{y}$. It is well-known that $V_\varepsilon, \widetilde{\phi}_\varepsilon \in C^\infty(\mathbb{R}^n)$ and that V_ε and ∇V_ε approximate V and ∇V uniformly on $\mathcal{D}_\mathcal{T}^{-\varepsilon}$, i.e.

$$\sup_{\mathbf{x}\in\mathcal{D}_\mathcal{T}^{-\varepsilon}} \max\{|V_\varepsilon(\mathbf{x}) - V(\mathbf{x})|, \|\nabla V_\varepsilon(\mathbf{x}) - \nabla V(\mathbf{x})\|_2\} \to 0 \quad \text{as} \quad \varepsilon \to 0^+.$$

By [18, Lemma 4.13] we have for $\varepsilon > 0$ and $\mathbf{x} \in \mathcal{D}_\mathcal{T}^{-\varepsilon}$ the formula

$$\mathbf{H}\big(V_\varepsilon(\mathbf{x})\big) = \sum_\nu \alpha_\nu^{\mathbf{x},\varepsilon}\mathbf{H}^\nu, \quad \text{where} \quad \alpha_\nu^{\mathbf{x},\varepsilon} := \int_{\mathcal{G}_\nu \cap B_\varepsilon(\mathbf{x})} \widetilde{\phi}_\varepsilon(\mathbf{x}-\mathbf{y})\, \mathrm{d}\mathbf{y}. \tag{25}$$

Note that the nonnegative numbers $\alpha_\nu^{\mathbf{x},\varepsilon}$ only depend on $\mathbf{x}$ and $\varepsilon > 0$ and not on V, and they sum to one for every $\mathbf{x} \in \mathcal{D}_\mathcal{T}^{-\varepsilon}$; the sum $\sum_\nu$ denotes that we sum over all ν such that $\mathcal{G}_\nu \in \mathcal{T}$.

Now, for a given $\delta > 0$ and compact $K \subset (\mathcal{D}_\mathcal{T})^\circ$, and the constant $C > 0$ in (21), choose $\varepsilon > 0$ so small that $K \subset \mathcal{D}_\mathcal{T}^{-\varepsilon}$,

$$|V_\varepsilon(\mathbf{x}) - V(\mathbf{x})| < \delta \quad \text{and} \quad \|\boldsymbol{\nabla} V_\varepsilon(\mathbf{x}) - \boldsymbol{\nabla} V(\mathbf{x})\|_2 \cdot \|\mathbf{f}(\mathbf{x})\|_2 \le \frac{C}{2}$$

for every $\mathbf{x} \in \mathcal{D}_\mathcal{T}^{-\varepsilon}$. Then, for every $\mathbf{x} \in \mathcal{D}_\mathcal{T}^{-\varepsilon}$ we have

$$\begin{aligned}
&\boldsymbol{\nabla} V_\varepsilon(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}) + \frac{1}{2}\sum_{i,j=1}^n \big[\mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^{\mathrm{T}}\big]_{ij} \big[\mathbf{H}\big(V_\varepsilon(\mathbf{x})\big)\big]_{ij} \\
&\quad = [\boldsymbol{\nabla} V_\varepsilon(\mathbf{x}) - \boldsymbol{\nabla} V(\mathbf{x})] \cdot \mathbf{f}(\mathbf{x}) \\
&\qquad + \sum_\nu \alpha_\nu^{\mathbf{x},\varepsilon}\Big[\boldsymbol{\nabla} V(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}) + \frac{1}{2}\sum_{i,j=1}^n \big[\mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^{\mathrm{T}}\big]_{ij}\big[\mathbf{H}^\nu\big]_{ij}\Big] \\
&\quad \le \frac{C}{2} - C = -\frac{C}{2},
\end{aligned}$$

which concludes the proof. □

For V_ε to strictly fulfill the conditions in Definition 3.3 one can choose open neighbourhoods $U' \subset U$ and $M' \supset M$ of the origin with $\overline{M'} \subset U'$, such that $\mathcal{D}_\mathcal{T}^{-\varepsilon} \supset U' \setminus \overline{M'} \supset K$.

3.4. **The LP problem.**

Linear Programming Problem 3.5. Consider the SDE (13) and assume that two neighbourhoods $U, M \subset \mathbb{R}^n$ of the origin are given, $\overline{M} \subset U$, together with a triangulation $\mathcal{T}$ of $\overline{U \setminus \overline{M}}$. The variables of the LP problem are $V_\mathbf{x} \in \mathbb{R}$ for every $\mathbf{x} = (\mathbf{x}_i^\nu + \mathbf{x}_j^\nu)/2$, where $\mathbf{x}_i^\nu$ and $\mathbf{x}_j^\nu$ are vertices of a simplex $\mathcal{G}_\nu \in \mathcal{T}$. Note that with $i = j$ the formula $\mathbf{x} = (\mathbf{x}_i^\nu + \mathbf{x}_j^\nu)/2$ includes the vertices $\mathbf{x} \in \mathbb{R}^n$ of the simplex $\mathcal{G}_\nu \in \mathcal{T}$. These values correspond to the values r_i and r_{ij} needed to define a CPQ function on $\mathcal{T}$. Further variables are $N_\nu \in \mathbb{R}$ and $P_\nu \in \mathbb{R}$ for every simplex $\mathcal{G}_\nu \in \mathcal{T}$ and $B \in \mathbb{R}$ to separate values of V on ∂U from those on ∂M. Recall that the components of $\nabla V_\nu(\mathbf{x}_i^\nu)$ and $\mathbf{H}_{ij}^\nu$ are linear in the variables $V_\mathbf{x}$ by Propositions 2.10 and 2.11, respectively.

The constants of the LP problem are $C > 0$, $\delta_\partial > 0$, and C_1^ν and C_2^ν as defined in (19) and (20) for every ν such that $\mathcal{G}_\nu \in \mathcal{T}$. The constant $C > 0$ is used to force LV to be negative and the constant $\delta_\partial > 0$ is used to separate values of V on ∂U from those on ∂M. Both are typically small, e.g. 10^{-4}.

The constraints of the LP problem are:

- For every ν such that $\mathcal{G}_\nu \in \mathcal{T}$ we demand

$$\left\|\boldsymbol{\nabla} V(\mathbf{x}_0)\right\|_\infty \leq N_\nu \quad \text{and} \quad \left\|\mathbf{H}^\nu\right\|_{\max} \leq P_\nu. \tag{26}$$

- For every $\mathbf{x} = \mathbf{x}_i^\nu = \mathbf{x}_j^\mu$, where $\mathbf{x}_i^\nu$ is a vertex of $\mathcal{G}_\nu \in \mathcal{T}$ and $\mathbf{x}_j^\mu$ is a vertex of $\mathcal{G}_\mu \in \mathcal{T}$, we demand

$$\nabla V_\nu(\mathbf{x}_i^\nu) = \nabla V_\mu(\mathbf{x}_j^\mu). \tag{27}$$

- For every $\mathcal{G}_\nu \in \mathcal{T}$ and every vertex $\mathbf{x}_k^\nu$ of $\mathcal{G}_\nu$ we demand

$$\boldsymbol{\nabla} V_\nu(\mathbf{x}_k^\nu) \cdot \mathbf{f}(\mathbf{x}_k^\nu) + \frac{1}{2}\sum_{i,j=1}^{n} \left[\mathbf{g}(\mathbf{x}_k^\nu)\mathbf{g}(\mathbf{x}_k^\nu)^{\mathrm{T}}\right]_{ij} \mathbf{H}_{ij}^\nu + C_1^\nu N_\nu + C_2^\nu P_\nu \leq -C. \tag{28}$$

The next two constraints are used to let V take lower values at the inner boundary than at the outer boundary of $\overline{U \setminus \overline{M}}$:

- For every $\mathbf{x} = \mathbf{x}_i^\nu \in \partial M$, where $\mathbf{x}_i^\nu$ is a vertex of $\mathcal{G}_\nu \in \mathcal{T}$, we demand

$$V_\mathbf{x} \leq B - \delta_\partial, \tag{29}$$

  and for every $\mathbf{x} = \mathbf{x}_i^\nu \in \partial U$, where $\mathbf{x}_i^\nu$ is a vertex of $\mathcal{G}_\nu \in \mathcal{T}$, we demand

$$V_\mathbf{x} \geq B + \delta_\partial. \tag{30}$$

Let us discuss the constraints of LP Problem 3.5 and their significance.

- The constraints (26), see (18), are used to obtain upper bounds N_ν and P_ν on $\left\|\boldsymbol{\nabla} V(\mathbf{x}_0)\right\|_\infty$ and $\left\|\mathbf{H}^\nu\right\|_{\max}$, respectively, that are used to bound the interpolation error in constraints (28).
- The constraints (27), see (23), are used to force ∇V to be continuous.
- The constraints (28), see (21) and (22), ensure that $LV_\nu \leq -C$ on each simplex.
- The constraints (29) and (30) are used to let V take lower values at the inner boundary than at the outer boundary of $\overline{U \setminus \overline{M}}$. This is useful because the level sets of a non-local Lyapunov function are needed to make stability guarantees. Note, however, that $V_\mathbf{x} \leq B - \delta_\partial$ for all vertices at the inner boundary ∂M of $\overline{U \setminus \overline{M}}$ does not necessarily imply that $V(\mathbf{x}) \leq B - \delta_\partial$ for all $\mathbf{x}$ at the inner boundary ∂M of $\overline{U \setminus \overline{M}}$ as V is a piecewise quadratic function, nor analogously for the outer boundary. Hence, the level sets of the computed V need be checked a posteriori.

Altogether, these constraints imply with Theorem 3.4 the existence of a non-local Lyapunov function.

4. **Examples.** We demonstrate our method for two systems from the literature. We implemented the method in C++ and used the linear solver Gurobi to solve the resulting LP problems. Both examples were solved in less than 20 seconds on a normal PC running on Linux Mint 20.1.

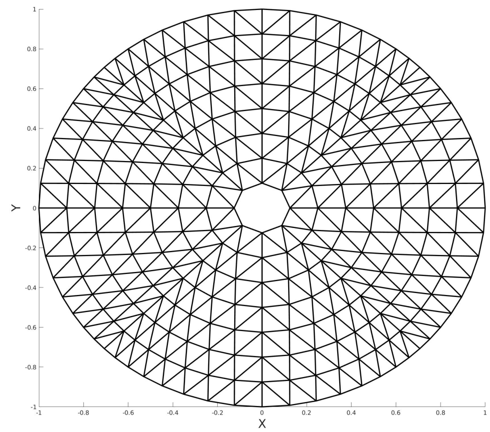

FIGURE 3. The triangulation used to compute a non-local Lyapunov function for system (31).

4.1. **Two-dimensional System.** Consider a harmonic oscillator

$$\ddot{x}(t) + \gamma \dot{x}(t) + \kappa x(t) = 0,$$

where the damping γ and the intensity of the force κ fluctuate randomly (white noise). This system has been studied in [22, Example 6.6] and [17]. In state-space form this can be modelled with the two-dimensional linear SDE, denoting $\mathbf{X}(t) = \begin{pmatrix} x(t) \\ \dot{x}(t) \end{pmatrix}$,

$$d\mathbf{X} = A\mathbf{X}dt + B_1\mathbf{X}dW_1 + B_2\mathbf{X}dW_2, \tag{31}$$

where W_1 and W_2 are independent one-dimensional Brownian motions,

$$A := \begin{pmatrix} 0 & 1 \\ -\kappa & -\gamma \end{pmatrix}, \quad B_1 := \begin{pmatrix} 0 & 0 \\ 0 & -\sigma_1 \end{pmatrix}, \quad B_2 := \begin{pmatrix} 0 & 0 \\ -\sigma_2 & 0 \end{pmatrix},$$

and the constants σ_1 and σ_2 determine the intensity of the fluctuations γ and κ respectively. For our method the simple form of the SDE, i.e. linear with constant coefficients, is not an advantage. However, this means that the stability of the origin for the system is more tractable with classical methods.

We fix the parameters of the problem as $\kappa = 1$, $\gamma = 0.1$, $\sigma_1 = 0.3$, and $\sigma_2 = 0.5$. One can verify that there does not exist a symmetric and positive definite $P \in \mathbb{R}^{2\times 2}$ such that $A^TP + PA + B_1^TPB_1 + B_2^TPB_2$ is negative definite, which implies that there does not exist a quadratic Lyapunov function $V(\mathbf{x}) = \mathbf{x}^TP\mathbf{x}$ for the system assuring mean-square stability of the origin. This implies by [1, Corollary 11.4.14] that the origin cannot be mean-square stable. However, using

$$C_1^\nu = 0 \quad \text{and} \quad C_2^\nu = n^2h_\nu^2[nQ\max(\sigma_1^2, \sigma_2^2) + 2\max(1, |\kappa|, |\gamma|)] = 12h_\nu^2,$$

see formulas (19) and (20), and the triangulation shown in Figure 3, we were able to compute a non-local Lyapunov function for the system using the LP Problem 3.5. The Lyapunov function is depicted in Figure 4.

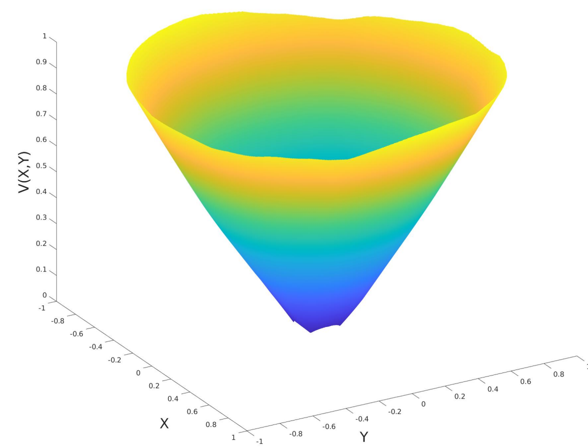

FIGURE 4. Non-local Lyapunov function for system (31).

4.2. **One-dimensional System.** We consider the non-linear system

$$dX = \sin(X)dt + \frac{3X}{1+X^2}dW \tag{32}$$

from [7], where W is a one-dimensional Wiener-process. Note that the deterministic part of the system, i.e. $\dot{x} = \sin(x)$, has an unstable equilibrium at the origin. Hence, it is an example of a system with an unstable equilibrium which is stabilized by noise. Since $f(x) = \sin(x)$ and $g(x) = 3x/(1+x^2)$ are odd functions, it is enough to compute a non-local Lyapunov function V for $x \geq 0$, because it can be extended symmetrically $V(x) = V(-x)$ to $x \leq 0$. The one-dimensional simplices of the triangulation of $[0, 8]$ we used for our computation were $[x_{i-1}, x_i]$ with $x_i = i \cdot 8/2400$ and $i = 4, 5, \ldots, 2400$. We used LP Problem 3.5 to assert $LV(x) \leq -C$ with $C = 10^{-7}$ on $[x_3, x_{2400}] = [0.01, 8]$ and $V(x_{2400}) - V(x_3) \geq 2\delta_\partial = 2 \cdot 10^{-4}$.

Since the LP Problem 3.5 is a feasibility problem, and the objective is not needed to compute a non-local Lyapunov function, we experimented with an objective. For this we added to the original LP problem the auxiliary variable D and the additional constraints: for every simplex $\mathcal{G}_\nu = [x^\nu_{i-1}, x^\nu_i]$ we demand for $k = i-1, i$ that

$$\nabla V_\nu(x^\nu_k) \cdot \mathbf{f}(x^\nu_k) + \frac{1}{2}\sum_{i,j=1}^{n} \left[\mathbf{g}(x^\nu_k)\mathbf{g}(x^\nu_k)^{\mathrm{T}}\right]_{ij} \mathbf{H}^\nu_{ij} - C^\nu_1 N_\nu - C^\nu_2 P_\nu \geq -C - D. \tag{33}$$

The objective of the LP problem was then to minimize D. Hence, with a similar argumentation as (21) to (22), from a feasible solution we get a $V \in C^1([0.01, 8])$ that fulfills

$$-C - D \leq LV(x) \leq -C$$

except (possibly) at the points x_i, $i = 3, 4, \ldots, 2400$, where LV might not even be defined. Hence, the objective and additional constraints force the computed non-local Lyapunov function to have $LV(x)$ very close to zero. In our example, we obtained a value of $D = 8 \cdot 10^{-7}$.

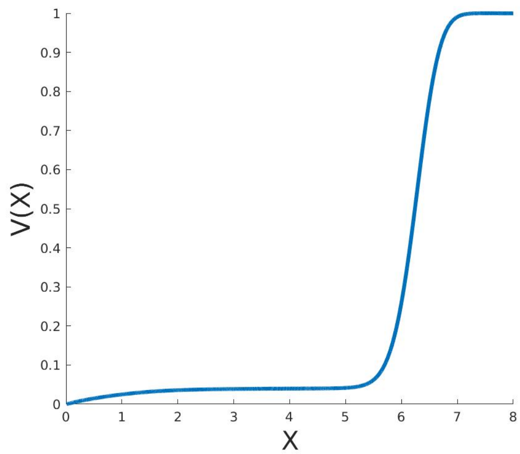

FIGURE 5. Non-local Lyapunov function for system (32).

The computed Lyapunov function is qualitatively the same as the one computed in [7] by numerically solving $LV(x) = -10^{-3}$ using collocation with radial basis functions. However, as we manage to keep $LV(x)$ closer to zero, the level-sets are slightly larger, which results in the asserted γ-basin of attraction to be larger. Further, note that our computed function delivers a true non-local Lyapunov function V_ε automatically by construction, whereas the function computed in [7] had to be rigourously verified by evaluating it at 750 million points. Hence, the computational time is shortened to seconds in our new method from hours needed for the verification in [7].

5. **Conclusion.** We have developed a linear programming (LP) problem to compute continuous piecewise quadratic (CPQ) non-local Lyapunov functions for stochastic differential equations (SDEs) with an equilibrium which is asymptotically stable in probability. As the Lyapunov function of a SDE must satisfy an inequality involving a second-order differential operator, one cannot employ continuous piecewise affine (CPA) functions.

We have provided explicit formulas for the gradient and Hessian of CPQ functions over a simplex, which, similarly to the gradient of a CPA function, involve the simplex's shape matrix. By enforcing continuity of the gradient, the CPQ function can be mollified with arbitrarily close level sets. We have applied our method to two systems from the literature and have used the proposed LP problem to compute non-local Lyapunov functions for these systems.

In the future, we plan on making our software to compute CPQ Lyapunov functions for SDEs more user-friendly and publish it in a public repository.

REFERENCES

[1] L. Arnold. *Stochastic Differential Equations: Theory and Applications*. Wiley, 1974.
[2] R. Baier, L. Grüne, and S. Hafstein. Linear programming based Lyapunov function computation for differential inclusions. *Discrete Contin. Dyn. Syst. Ser. B*, 17(1):33–56, 2012.
[3] G. Barrera, E. Bjarkason, and S. Hafstein. The stability of the multivariate geometric Brownian motion as a bilinear matrix inequality problem. *SIAM J. Appl. Dyn. Syst.*, to appear.

[4] E. Bjarkason. *Lyapunov functions for linear stochastic differential equations by bilinear matrix inequalities–theory and implementation*. MSc thesis: University of Iceland, 2022.
[5] H. Björnsson. *Lyapunov Functions for Stochastic Systems: Theory and Numerics*. PhD thesis: University of Iceland, 2023.
[6] H. Björnsson, P. Giesl, S. Gudmundsson, and S. Hafstein. Local Lyapunov functions for nonlinear stochastic differential equations by linearization. In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pages 579–586, 2018.
[7] H. Björnsson, S. Hafstein, P. Giesl, E. Scalas, and S. Gudmundsson. Computation of the stochastic basin of attraction by rigorous construction of a Lyapunov function. *Discrete Contin. Dyn. Syst Ser. B*, 24(8):4247–4269, 2019.
[8] F. Camilli, L. Grüne, and F. Wirth. Control Lyapunov functions and Zubov's method. *SIAM J. Control Optim.*, 47(1):301–326, 2008.
[9] C. Conley. *Isolated Invariant Sets and the Morse Index*. CBMS Regional Conference Series no. 38. American Mathematical Society, 1978.
[10] P. Giesl and S. Hafstein. Construction of Lyapunov functions for nonlinear planar systems by linear programming. *J. Math. Anal. Appl.*, 388:463–479, 2012.
[11] P. Giesl and S. Hafstein. Revised CPA method to compute Lyapunov functions for nonlinear systems. *J. Math. Anal. Appl.*, 410(1):292–306, 2014.
[12] P. Giesl and S. Hafstein. Computation and verification of Lyapunov functions. *SIAM J. Appl. Dyn. Syst.*, 14(4):1663–1698, 2015.
[13] I. Gihman and A. Skorohod. *Stochastic Differential Equations*. Springer-Verlag Berlin, Heidelberg, New York, 1972.
[14] L. Grüne and F. Camilli. Characterizing attraction probabilities via the stochastic Zubov equation. *Discrete Contin. Dyn. Syst. Ser. B*, 3(3):457–468, 2003.
[15] S. Gudmundsson and S. Hafstein. Probabilistic basin of attraction and its estimation using two Lyapunov functions. *Complexity*, Article ID:2895658, 2018.
[16] S. Hafstein. Lyapunov functions for linear stochastic differential equations: Bmi formulation of the conditions. In *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pages 147–155, 2019.
[17] S. Hafstein, S. Gudmundsson, P. Giesl, and E. Scalas. Lyapunov function computation for autonomous linear stochastic differential equations using sum-of-squares programming. *Discrete Contin. Dyn. Syst Ser. B*, 23(2):939–956, 2018.
[18] S. Hafstein and C. Kawan. Numerical approximation of the data-rate limit for state estimation under communication constraints. *J. Math. Anal. Appl.*, 473(2):1280–1304, 2019.
[19] W. Hahn. *Stability of Motion*. Springer, Berlin, 1967.
[20] M. Hurley. Lyapunov functions and attractors in arbitrary metric spaces. *Proc. Amer. Math. Soc.*, 126:245–256, 1998.
[21] H. Khalil. *Nonlinear Systems*. Pearson, 3. edition, 2002.
[22] R. Khasminskii. *Stochastic stability of differential equations*. Springer, 2nd edition, 2012.
[23] M. Kheirandishfard, F. Zohrizadeh, M. Adil, and R. Madani. Convex relaxation of bilinear matrix inequalities part II: Applications to optimal control synthesis. In *Proceedings of 57rd IEEE Conference on Decision and Control (CDC)*, pages 75–82, 2018.
[24] M. Kheirandishfard, F. Zohrizadeh, and R. Madani. Convex relaxation of bilinear matrix inequalities part I: Theoretical results. In *Proceedings of 57rd IEEE Conference on Decision and Control (CDC)*, pages 67–74, 2018.
[25] A. Lahrouz, L. Omari, D. Kiouach, and A. Balmaâti. Deterministic and stochastic stability of a mathematical model of smoking. *Stat. Probabil. Lett.*, 81:1276–1284, 2011.
[26] A. M. Lyapunov. The general problem of the stability of motion. *Internat. J. Control*, 55(3):521–790, 1992. Translated by A. T. Fuller from Édouard Davaux's French translation (1907) of the 1892 Russian original. With an editorial (historical introduction) by Fuller, a biography of Lyapunov by V. I. Smirnov, and the bibliography of Lyapunov's works collected by J. F. Barrett, Lyapunov centenary issue.
[27] X. Mao. *Stochastic Differential Equations and Applications*. Woodhead Publishing, 2nd edition, 2008.
[28] B. Øksendal. *Stochastic Differential Equations*. Springer, 6th edition, 2010.
[29] R. Sarkar and S. Banerjee. Cancer self remission and tumor stability–a stochastic approach. *Math. Biosci.*, 196(1):65–81, 2005.
[30] S. Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Springer, 1999.

[31] M. Senosiain and A. Tocino. A survey of mean-square destabilization of multidimensional linear stochastic differential systems with non-normal drift. *Numer. Algorithms*, 93(4):1543–1559, 2023.

[32] M. Shumafov and V. Tlyachev. Construction of Lyapunov functions for second-order linear stochastic stationary systems. *J. Math. Sciences*, 250(5):835–846, 2020.

[33] M. Vidyasagar. *Nonlinear System Analysis*. Classics in Applied Mathematics. SIAM, 2. edition, 2002.

Received xxxx 20xx; revised xxxx 20xx; early access xxxx 20xx.