UNIVERSITY
OF ICELAND

Ph.D. Dissertation
in Mathematics

Lyapunov Functions for Stochastic Systems: Theory and Numerics

Hjörtur Björnsson

May 2023

FACULTY OF NATURAL SCIENCES

Lyapunov Functions for Stochastic Systems: Theory and Numerics

Hjörtur Björnsson

Dissertation submitted in partial fulfillment of a
Philosophiae Doctor degree in Mathematics

Ph.D. Committee
Sigurður Freyr Hafstein (Chair)
Sigurður Örn Stefánsson
Peter Giesl

Gerardo Barrera Vargas
Florian Rupp

Faculty of Natural Sciences
School of Engineering and Natural Sciences
University of Iceland
Reykjavik, May 2023

Lyapunov Functions for Stochastic Systems: Theory and Numerics

Dissertation submitted in partial fulfillment of a *Ph.D.* degree in Mathematics

Faculty of Natural Sciences
School of Engineering and Natural Sciences
University of Iceland
Dunhagi 5
107, Reykjavík Reykjavik
Iceland

Telephone: 525-4000

Bibliographic information:
Hjörtur Björnsson, 2023, *Lyapunov Functions for Stochastic Systems: Theory and Numerics*, Ph.D.Dissertation, Faculty of Natural Sciences, University of Iceland, 112 pp.

Author ORCID: 0009-0004-9455-3577
ISBN: 978-9935-9630-7-9

Printing: Háskólaprent
Reykjavik, Iceland, May 2023

Abstract

In this thesis we explore Lyapunov functions for stochastic differential equations. Lyapunov functions are a useful tool to study the stability of an equilibrium of such stochastic systems and long term behaviour of solutions around an equilibrium point. We discuss a method of estimating, with rigorous lower bounds, what is called the basin of attraction of an equilibrium by using a combination of local and non-local Lyapunov functions. First we talk about a generalisation of Lyapunov functions for almost sure exponential stability, to a larger and more useful class of functions. Second, we describe a method to calculate local Lyapunov functions by linearising the stochastic system. Third, we calculate non-local Lyapunov functions using a numerical method called meshless collocation. This approach allows us to calculate true Lyapunov functions numerically instead of just approximations. We describe this method and talk about the computational challenges that we faced, which led us to study Wendland functions and explore different ways to evaluate them numerically. This resulted in a method to generate Wendland functions in an optimal form and a software library to carry out meshless collocation using Wendland functions to solve partial differential equations numerically.

Ágrip

Í þessari ritgerð munum við skoða Lyapunov föll fyrir slembnar diffurjöfnur. Lyapunov föll eru mjög gagnleg þegar við könnum stöðugleika jafnvægispunkta slembna kerfa, og langtíma hegðun sérlausna í kringum þá. Við fjöllum um aðferð, til að reikna svokallað aðdráttarsvæði og nákvæmar neðri skorður, sem felst í því að nota saman staðbundið Lyapunov fall og óstaðbundið Lyapunov fall. Fyrst fjöllum við um Lyapunov föll, fyrir næstum örugglega veldisaðfellustöðug kerfi, og alhæfum þau fyrir stærra og gagnlegra fallarúm. Í öðru lagi þá fjöllum við um aðferð til að reikna staðbundin Lyapunov föll með því að línugera slembna kerfið. Í þriðja lagi, þá reiknum við óstaðbundið Lyapunov fall með tölulegri aðferð sem kallast möskvalaus samleguaðferð. Með þessari aðferð getum við reiknað út raunverulegt Lyapunov fall tölulega í stað þess reikna út tölulega nálgun. Við fjöllum um reiknierfiðleika sem við tókumst á við, og hvernig í framhaldinu við skoðuðum Wendland föll og mismunandi aðferðir til að reikna þau tölulega. Niðurstaðan var aðferð sem býr til Wendland föll í hagkvæmri mynd og hugbúnaður sem nýtir þau til að leysa hlutafleiðujöfnur með mösvkalausri samleguaðferð.

For Stefania.

Table of Contents

List of Publications

This thesis is comprised of the following papers. A short description of the papers can be found in chapter 5.

Paper I: **H. Bjornsson** and S.F. Hafstein, 2018, Lyapunov functions for almost sure exponential stability. *Dynamical Systems in Theoretical Perspective*, Springer Proceedings in Mathematics and Statistics 248.

Paper II: **H. Bjornsson**, S.F. Hafstein, P. Giesl, E. Scalas, and S. Gudmundsson, 2019, Computation of the stochastic basin of attraction by rigorous construction of a Lyapunov function. *Discrete and Continuous Dynamical Systems*, Series B, 24-8, Issue 2, pp. 4247–4269.

Paper III: **H. Bjornsson**, P. Giesl, S. Gudmundsson, and S.F. Hafstein, 2018, Local Lyapunov Functions for Nonlinear Stochastic Differential Equations by Linearization. *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pp. 579–586.

Paper IV: **H. Bjornsson** and S.F. Hafstein, 2021, Advanced algorithm for interpolation with Wendland functions. *Informatics in Control, Automation and Robotics*, Lecture Notes in Electrical Engineering 720, pp. 99–117.

Paper V: **H. Bjornsson** and S.F. Hafstein, 2019, Algorithm and Software to Generate Code for Wendland Functions in Factorized Form. *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pp. 156–162.

Paper VI: **H. Bjornsson** and S.F. Hafstein, 2018, Verification of a Numerical Solution to a Collocation Problem. *In Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pp. 587–594.

Acknowledgements

First and foremost, I would like to thank my supervisor Sigurður Freyr Hafstein for his guidance, support and patience throughout the years we have worked together, and for offering me the opportunity to pursue a PhD in mathematics. I also thank Peter Giesl who also played a big role during this project as a collaborator and has provided me with much needed help and advice, and Sigurður Örn Stefánsson for allocating his time to help prepare this thesis. Additionally, I would like to thank Skúli Guðmundsson, Enrico Scalas and Carlos Argaez Garcia for their collaborations.

I would also like to thank Watse Sybesma, Bergur Snorrason, Eyleifur Ingþór Bjarkason, and Þórarinn Jónmundsson who I have got to know throughout the years at the office and are some of my dearest friends today.

I am grateful to my parents Bryndís and Björn who have given me all the opportunities and made it possible for me to follow my passions over the years. I would also like to thank my siblings Harpa Lind, Ragnar, and Hanna Auðbjörg.

Last but certainly not the least, I would like to thank my partner Stefania Crotti who has always been with me, through thick and thin, since our paths intertwined.

This research was supported by the Icelandic Research Fund (Rannis), grant number 152429-051, Lyapunov Methods and Stochastic Stability.

1 Introduction

The concept of stability of a classical dynamical system captures the idea that sample paths with two close starting values will not diverge too far away from each other as time passes. Specifically, an equilibrium is stable if paths that start close to the equilibrium stay close at all times.

Aleksandr Lyapunov defined functions to analyse the stability of an equilibrium for deterministic dynamical systems, and a generalisation of these functions play a central role in our thesis project. Lyapunov functions are comparable to potential energy to some extent. The idea is, to show that the ‘energy’, or function value, is decreasing along all solution paths. If that is the case, then all solution paths must tend to a minimum ‘energy level’ at the equilibrium, so all solution paths tend to the equilibrium. This is a simple but powerful idea that is the cornerstone for the study of stability of dynamical systems.

The thesis topic, however, is the analysis of the stability of *stochastic* differential systems as opposed to *deterministic* dynamical systems. There are, however, many similarities between the two different types of systems, and there is a theme of generalising concepts that work well for the study of deterministic dynamical systems to stochastic ones. One such generalisation is the previously mentioned Lyapunov function. For stochastic systems we can not expect to find a function that is decreasing for all sample paths of a given initial value, as sample paths of stochastic systems can vary wildly. The correct generalisation is then to look for *supermartingales*, that is to say functions whose expectation decreases with time. A complication that arises is that in general we can not expect the Lyapunov functions to be differentiable at the origin, therefore in paper [Bjornsson and Hafstein 2018a(Paper I)] we generalise a result on a type of Lyapunov functions to the larger class of functions not differentiable at the origin.

Another generalisation of a deterministic concept is what we call *γ-basin of attraction* for our stochastic system. For deterministic systems given by an autonomous differential equation, the basin of attraction of an equilibrium is the set of all initial values such that the solutions tend to the equilibrium as time increases. For the stochastic concept, a γ-basin of attraction is the set of initial values such that the sample paths tend to the equilibrium as time increases, with probability at least γ. That is to say, for all initial values inside the γ-basin of attraction, the paths leave the basin with probability less than $1-\gamma$.

Following the method used in the paper [Gudmundsson and Hafstein 2018] we use a combination of a local and a non-local Lyapunov function to estimate the γ-basin of attraction. The local Lyapunov function is calculated by finding a Lyapunov function for the linearised version of the system, as is done in paper [Bjornsson, Giesl, et al. 2018(Paper III)], which will be a Lyapunov function for the original stochastic system.

The non-local Lyapunov function is calculated numerically using so called *meshless collocation* using *radial basis functions* (RBF). The specific radial basis functions we use are called

Wendland functions, which are piece-wise polynomials with compact support. The method aims to solve the Lyapunov PDE boundary value problem, which is a partial differential equation of the form $LV(\mathbf{x}) = h < 0$ for all $\mathbf{x} \in \mathscr{U}$, for a domain $\mathscr{U}$, and some fixed boundary values on $\partial\mathscr{U}$, where L is the second-order differential operator defined in equation (16). This is done by constructing a function V which satisfies the PDE $LV(\mathbf{x}_i) = h < 0$ for a set of points $\mathbf{x}_i \in \mathscr{U}$, using the radial basis functions. The detailed method is laid out in the paper [Bjornsson, Hafstein, et al. 2019(Paper II)], which furthermore provides rigorous a-posteriori estimates on $LV(\mathbf{x})$. This provides us with a method to verify that the approximation is a valid Lyapunov function, even when the theory of elliptic PDEs does not guarantee the existence of a solution to our Lyapunov PDE, as is the case if the operator L is not strictly elliptic.

The verification of the non-local Lyapunov function involves evaluating $LV(\mathbf{x})$ in a test grid, which can have a very high number of points and each point requires us to evaluate the Wendland RBF and its derivatives up to 4th order. It is therefore essential that these evaluations of the Wendland functions can be carried out in an efficient and accurate way. This brings us to the last part of the project which was to investigate different methods of evaluating Wendland functions, as was done in paper [Bjornsson and Hafstein 2018b(Paper IV)]. The result was that a specific factorized form of Wendland functions was most efficient. In papers [Bjornsson and Hafstein 2019b(Paper V)] and [Bjornsson and Hafstein 2019a(Paper IV)] we provide an algorithm and software to generate the Wendland functions in this factorized format. We furthermore provide a software library and examples of how to carry out the meshless collocation using Wendland functions in the specified format. The software is available at `https://gitlab.com/hjortur/wendland-function-generator`.

The outline of the thesis is as follows:

In chapter 2 we give a general description of stochastic differential equations, their stability and a general introduction to Lyapunov functions. We furthermore talk about issues that arise with differentiability at the origin.

In chapter 3 we talk about using two different Lyapunov functions, local and non-local, and give a discussion on their use to estimate the γ-basin of attraction.

In chapter 4 we describe meshless collocation, a numerical method to solve partial differential equations, and specifically how we used this method to calculate non-local Lyapunov functions, using Wendland radial basis functions. We discuss verification of the Lyapunov function properties and computational challenges.

In chapter 5 we list the six papers the thesis project is comprised of, and we provide a short description of each paper.

Finally in chapter 6 we will conclude the thesis with a short discussion on the results obtained.

2 Stochastic differential equations

2.1 Solutions to SDEs

A stochastic differential equation is an equation in the following form

$$dX(t) = f(X(t),t)dt + g(X(t),t)dB(t), \tag{1}$$

where $f : \mathbf{R}^d \times [t_0, \infty[\to \mathbf{R}^d$ and $g : \mathbf{R}^d \times [t_0, \infty[\to \mathbf{R}^d \times \mathbf{R}^Q$. This is simply a short hand for the integral equation

$$X(t) = x_0 + \int_{t_0}^{t} f(X(s),s)ds + \int_{t_0}^{t} g(X(s),s)dB(s), \quad t \geq t_0, \tag{2}$$

where the second integral is a multidimensional Itô integral, see [Øksendal 2010, Def. 3.3.1].

This integral is with respect to a Q-dimensional Wiener process, or *Brownian motion* $\{B(t)\}_{t\geq 0}$, which is also sometimes denoted by $\{W(t)\}_{t\geq 0}$.

We say that the process $\{B(t)\} = (B_1(t), B_2(t), \ldots, B_Q(t))\}_{t\geq 0}$ is a *standard* Q-dimensional Brownian motion if all $\{B_j(t)\}_{t\geq 0}$, $j = 1, \ldots, Q$ are 1-dimensional Brownian motions and all $\{B_1(t)\}_{t\geq 0}$, $\{B_2(t)\}_{t\geq 0}$, $\cdots$, $\{B_Q(t)\}_{t\geq 0}$ are independent. Now for a fixed t, $B(t)$ is a random variable, but by abuse of notation we often call $B(t)$ a Brownian motion. In this case we are referring to the whole Brownian motion stochastic process instead of a fixed time t.

The result of the thesis project is mostly concerned with *autonomous* differential equations, that is where neither f nor g depend explicitly on the time variable t. In that case we can write the SDE in the form:

$$dX(t) = f(X(t))dt + g(X(t))dB(t). \tag{3}$$

However, we present some of the theory here for the more general case of time dependent SDEs.

We work in a complete probability space $(\Omega, \mathscr{F}, \mathbf{P})$ with a right continuous filtration $\{\mathscr{F}_t\}_{t\geq 0}$ such that $\mathscr{F}_0$ contains all $\mathbf{P}$ null sets. We furthermore assume that the process $\{B(t)\}_{t\geq 0}$ is a Brownian motion with respect to the filtration $\{\mathscr{F}_t\}_{t\geq 0}$. We denote by $\mathbf{E}$ the expectation with respect to the probability measure $\mathbf{P}$. A solution (or *strong* solution) of the SDE in equation (1) is a process $X(t)$ that satisfies the following condition:

Definition 2.1. *An $\mathbf{R}^d$ valued stochastic process $X(t)$ for $t_0 \leq t$ is called a solution of the SDE in* (1) *if it has the following properties:*

(i) $X(t)$ is continuous and $\mathscr{F}_t$-adapted.

(ii) The processes $f(X(t),t)$ and $g(X(t),t)$ are in $\mathscr{L}^1$ and $\mathscr{L}^2$ respectively.

(iii) Equation (2) *holds for all t in* $[t_0,\infty[$ *with probability* 1.

We say that a solution $X(t)$ of (1) *is* unique *if for any other solution $Y(t)$ we have*

$$\mathbf{P}\big[X(t)=Y(t) \ \textit{for all } t \geq t_0\big]=1,$$

that is, $X(t)$ and $Y(t)$ are indistinguishable.

In this thesis we assume that the functions f and g in equation (1) satisfy the conditions below which are sufficient for the existence of unique *strong solutions* to the SDE. These are relatively mild conditions on the functions f and g, and are summed up in the following theorem, see [Mao 2008; Khasminskii 2012]:

Theorem 2.2. *Assume that for every $T \geq t_0$, there exist two positive constants K_T and $\overline{K}_T$ such that for all $x,y \in \mathbf{R}^d$ and all $t \in [t_0,T]$:*

(i)

$$\|f(x,t)-f(y,t)\|^2 \bigvee \|g(x,t)-g(y,t)\|^2 \leq K_T\|x-y\|^2; \tag{4}$$

(ii)

$$\|f(x,t)\|^2 \bigvee \|g(x,t)\|^2 \leq \overline{K}_T(1+\|x\|^2). \tag{5}$$

Then there exists a unique solution to the SDE in (1) *and furthermore that solution belongs to* $\mathscr{M}^2([t_0,\infty[,\mathbf{R}^d)$.

Here the symbol $\vee$ denotes the *maximum*, i.e. $a \vee b = \max(a,b)$, and $\mathscr{M}^2([t_0,\infty[,\mathbf{R}^d)$ denotes the set of $\{\mathscr{F}_t\}$-adapted $\mathbf{R}^d$ valued processes $h(t)$ such that

$$\mathbf{E}\int_{t_0}^{T} \|h(t)\|^2 dt < \infty \quad \text{for every } T \geq t_0. \tag{6}$$

The conditions given in the previous theorem are known as the *Lipschitz conditions* and the *Linear growth conditions*. These conditions, as well as the assumptions on the probability space $(\Omega,\mathscr{F},\mathbf{P})$ and filtration $\{\mathscr{F}_t\}_{t\geq t_0}$ are called *the usual conditions*.

The usual conditions guarantee that for any given *deterministic* initial value $X(t_0)=\mathbf{x}_0 \in \mathbf{R}^d$ there exists a unique strong solution to equation (1), with continuous sample paths, denoted by $X^{\mathbf{x}_0,t_0}$. In integral form

$$X^{\mathbf{x}_0,t_0}(t)=\mathbf{x}_0+\int_{t_0}^{t} f(X^{\mathbf{x}_0,t_0}(s),s)ds+\int_{t_0}^{t} g(X^{\mathbf{x}_0,t_0}(s),s)dB(s), \quad t_0 \leq t. \tag{7}$$

We often even suppress the initial time t_0 from the notation, and simply write $X^{\mathbf{x}_0}$ as the initial time is just 0 or understood from context.

Finally, we assume that

$$f(0,t)=0 \quad \text{and} \quad g(0,t)=0 \quad \text{for all } t \geq t_0, \tag{8}$$

which ensures that the origin is an equilibrium of the SDE and we have the *trivial* solution $X(t)=0$. Note that this is a necessary condition for the trivial solution to even exist. These zeros of the functions f and g along with the *usual conditions* are assumed to hold throughout this thesis.

2.2 Stability of SDEs

There are many different versions of stability for SDEs that are in use. Here we review some of them. For further information we direct the reader to [Khasminskii 2012; Mao 2008; Kloeden et al. 1992].

Definition 2.3. *(Stability in Probability (SiP)) The trivial solution* $X(t) = 0$ *of the standard SDE in* (1) *is said to be* stochastically stable *or* stable in probability *(SiP) if for every* $r > 0$ *and* $0 < \varepsilon < 1$ *there exists a* $\delta > 0$ *such that:*

$$\|\mathbf{x}\| < \delta \quad \textit{implies} \quad \mathbf{P}\left\{ \sup_{t>0} \|X^{\mathbf{x}}(t)\| < r \right\} \geq 1 - \varepsilon. \tag{9}$$

Recall that $X^{\mathbf{x}}(t)$ *denotes the solution of equation* (1) *which satisfies the initial condition* $X^{\mathbf{x}}(0) = \mathbf{x}$*, i.e. the solution started at* $\mathbf{x}$.

Definition 2.4. *(Asymptotic Stability in Probability (ASiP)) The trivial solution* $X(t) = 0$ *to the standard SDE in* (1) *is said to be* stochastically asymptotically stable *or* asymptotically stable in probability *(ASiP) if it is stochastically stable and in addition for every* $0 < \varepsilon < 1$ *there exists a* $\delta > 0$ *such that:*

$$\|\mathbf{x}\| < \delta \quad \textit{implies} \quad \mathbf{P}\left\{ \lim_{t\to\infty} \|X^{\mathbf{x}}(t)\| = 0 \right\} > 1 - \varepsilon. \tag{10}$$

A quick explanation of these different conditions is the following: The first condition (SiP) tells us that if we start close enough to the origin, then the solution stays close to the origin with as high a probability as we want. The second condition (ASiP) says that starting close enough the solution not only stays close to the origin but also tends to the zero solution.

In his book [Khasminskii 2012] writes these definitions as the limits

$$\lim_{\|\mathbf{x}\|\to 0} \mathbf{P}\left\{ \sup_{t>0} \|X^{\mathbf{x}}(t)\| \leq r \right\} = 1 \quad \forall r > 0 \quad \text{SiP} \tag{11}$$

$$\lim_{\|\mathbf{x}\|\to 0} \mathbf{P}\left\{ \limsup_{t\to\infty} \|X^{\mathbf{x}}(t)\| = 0 \right\} = 1 \qquad \text{ASiP}. \tag{12}$$

The reason we do chose a different representation is that we want to look at the stochastic analog of what is called *basin of attraction* (BOA) in deterministic theory. Its motivation is simple and it aims to answer the following question: How far from the origin can sample paths start and still approach the equilibrium as $t \to \infty$ with probability greater than a given confidence γ.

Definition 2.5. *(γ Basin of attraction) Consider a system with domain* $\mathscr{U} \subset \mathbf{R}^d$ *satisfying the usual conditions. For any* $0 < \gamma \leq 1$ *the set* $\mathscr{A}_\gamma \subset \mathscr{U}$*, given by:*

$$\mathscr{A}_\gamma = \left\{ \mathbf{x} \in \mathscr{U} \;\middle|\; \mathbf{P}\left\{ \lim_{t\to\infty} \|X^{\mathbf{x}}(t)\| = 0 \right\} \geq \gamma \right\}, \tag{13}$$

is called the γ basin of attraction *or the* γ*-BOA, see (Paper II and Paper III). Here* $X^{\mathbf{x}}(t)$ *denotes the unique strong solution of the SDE with initial condition* $\mathbf{x}$.

Now if $\mathscr{A}_\gamma$ is the γ-BOA for an SDE satisfying the usual conditions, then any sample path started in $\mathscr{A}_\gamma$ will tend towards the origin with probability at least γ.

There is also another type of stability we deal with in this thesis, called *almost sure exponential stability*, see e.g. [Mao 2008], which is the following condition:

Definition 2.6. *(Almost sure exponential stability) The trivial solution $X(t) = 0$ of equation* (1) *is said to be almost surely exponentially stable if for every initial value $\mathbf{x}$, the solution $X^{\mathbf{x}}(t)$ to* (1) *satisfies:*

$$\limsup_{t\to\infty} \frac{1}{t} \log \|X^{\mathbf{x}}(t)\| < 0 \quad \textit{almost surely}. \tag{14}$$

In contrast to the previous types of stability, this last one says that almost all sample paths of the solution will tend to the origin exponentially fast.

2.3 Lyapunov functions

For the stochastic differential equation in (1), written here again for clarity,

$$dX(t) = f(X(t),t)dt + g(X(t),t)dB(t), \tag{15}$$

there is an associated *generator*:

$$\begin{aligned} Lv(\mathbf{x},t) &= \partial_t v(\mathbf{x},t) + f(\mathbf{x},t)\cdot\nabla_{\mathbf{x}} v(\mathbf{x},t) + \frac{1}{2}\operatorname{tr}\left(g(\mathbf{x},t)^\top \mathscr{H}v(\mathbf{x},t)g(\mathbf{x},t)\right) \\ &= \frac{\partial v}{\partial t}(\mathbf{x},t) + \sum_{i=1}^d f_i(\mathbf{x},t)\frac{\partial v}{\partial x_i}(\mathbf{x},t) + \frac{1}{2}\sum_{q=1}^Q \sum_{i,j=1}^d g_i^q(\mathbf{x},t)\frac{\partial^2 v}{\partial x_i \partial x_j}(\mathbf{x},t) g_j^q(\mathbf{x},t) \\ &= \frac{\partial v}{\partial t}(\mathbf{x},t) + \sum_{i=1}^d f_i(\mathbf{x},t)\frac{\partial v}{\partial x_i}(\mathbf{x},t) + \frac{1}{2}\sum_{i,j}^d \left[g(\mathbf{x},t)g(\mathbf{x},t)^\top\right]_{ij} \frac{\partial^2 v}{\partial x_i \partial x_j}(\mathbf{x},t). \end{aligned} \tag{16}$$

where $v : \mathbf{R}^d \times \mathbf{R}_+$ is an appropriately differentiable function, and $\mathscr{H}v(\mathbf{x},t)$ is the *Hessian* of the function $\mathbf{x} \mapsto v(\mathbf{x},t)$. In this thesis we follow the convention that $\mathbf{R}_+ = \{t \in \mathbf{R} \mid t \geq 0\}$. Note that this is just the drift term of the stochastic differential for the Itô process $t \mapsto v(X(t),t)$ where $X(t)$ is a process with differential given by (15). We write here the famous Itô's formula for completeness [Øksendal 2010, Thm. 4.2.1]:

Theorem 2.7. *(Itô's formula) Let $v(\mathbf{x},t)$ be a function defined on $\mathbf{R}^d \times \mathbf{R}_+$ that has continuous partial derivatives up to second order in $\mathbf{x}$ and up to first order in t. If $X(t)$ is a d-dimensional Itô process with differentials given by* (1) *then the process $Y(t) = v(X(t),t)$ is also an Itô process with differential given by*

$$\begin{aligned} dY(t) = dv(X(t),t) &= Lv(X(t),t)dt + \sum_i^d \frac{\partial v}{\partial x_i}(X(t),t)g_i(X(t),t)dB(t) \\ &= Lv(X(t),t)dt + \nabla_{\mathbf{x}} v(X(t),t)g(X(t),t)dB(t). \end{aligned} \tag{17}$$

Here Lv is the generator given by (16) *and $\nabla_{\mathbf{x}} v(\mathbf{x},t)$ is the gradient with respect to the spatial coordinate $\mathbf{x}$, written as a row vector.*

We usually just write $dv(X(t),t)$ instead of $dY(t)$ in the theorem above. We remind the reader that the differential formulation given by Thm. 2.7 is a short hand for writing out the process

$$v(X(t),t) = v(X(t_0),t_0) + \int_{t_0}^{t} Lv(X(s),s)ds + \int_{t_0}^{t} \nabla_{\mathbf{x}} v(X(t),t) \cdot g(X(t),t)dB(t), \tag{18}$$

and by taking the expectation of both sides we get

$$\mathbf{E}\left[v(X(t),t\right] = v(X(t_0),t_0) + \mathbf{E}\left[\int_{t_0}^{t} Lv(X(s),s)ds\right]. \tag{19}$$

This is because the stochastic integral part in equation (18) is a martingale with expectation zero. Let U be some closed bounded domain in $\mathbf{R}^d$ and assume that $v(\mathbf{x},t) \geq 0$ and $Lv(\mathbf{x},t) \leq 0$ for all $(\mathbf{x},t) \in U \times [t_0,\infty[$. Then the process $v(X(t),t)$ is a *supermartingale* with

$$\mathbf{E}\left[v(X(\tau\wedge t),\tau\wedge t) \mid \mathscr{F}_s\right] \leq v(X(s),s) \quad \text{a.s.} \tag{20}$$

where $\tau = \inf\{t \mid X(t) \in U^c\}$ is the first exit time from U, and $\tau \wedge t = \min(\tau,t)$. Note that the above equation furthermore implies

$$\mathbf{E}\left[v(X(t),t\right] \leq v(X(t_0),t_0). \tag{21}$$

This is the key property that has been used to study stability of SDEs, see e.g. [Mao 2008; Khasminskii 2012], and these type of functions v as described above, with additional restrictions, are called *Lyapunov* functions. There are many different definitions of Lyapunov functions used throughout the literature, see e.g. [Menshikov, Popov, and Wade 2016] for Lyapunov functions for Random Walks, and [Shaikhet 2013] for Lyapunov functions for stochastic difference equations. All of these are different variations on the theme described before.

2.4 Differentiability at the origin

It turns out that the condition that $v(\mathbf{x},t)$ being twice continuously differentiable in the $\mathbf{x}$ coordinate is too strict. Rafail Khasminskii showed in his book "Stochastic Stability of Differential Equations" [Khasminskii 2012, Chpt. 5.3] that even for the simplest case, 1 dimensional autonomous SDE which is *stable* and with constant coefficients, there may not exist such (Lyapunov) v which are smooth at the origin. Indeed the example Khasminskii uses is

$$dX = bXdt + \sigma XdB, \tag{22}$$

where b and σ are constants such that $0 \leq b < \sigma^2/2$. This system is *stable* (see Definition 2.3) but since $b \geq 0$ its *deterministic* part is unstable, and exhibits the interesting phenomenon that a deterministic system can be stabilised with noise.

This gives rise to a class of functions denoted $C_0^{2,1}(U \times \mathbf{R}_+)$, where we say that a function $v(\mathbf{x},t)$ is of class $C_0^{2,1}(U \times \mathbf{R}_+)$ if it is twice continuously differentiable with respect to $\mathbf{x}$ and continuously differentiable with respect to t for all $(\mathbf{x},t) \in U \times \mathbf{R}_+$, except possibly for the set $\mathbf{x} = 0$. This is the correct class of Lyapunov functions to consider and the definition here is the one used by [Khasminskii 2012].

To make the definitions easier to state we use so called $\mathcal{K}_\infty$-functions. We say that μ is a $\mathcal{K}_\infty$-function, or $\mu \in \mathcal{K}_\infty$ if $\mu : [0,\infty[\to [0,\infty[$, is a continuous and strictly increasing function such that $\mu(0) = 0$, and $\lim_{x\to\infty} \mu(x) = \infty$. We then get the following theorem, see [Khasminskii 2012]:

Theorem 2.8. *(Stochastic Lyapunov function) If there exists a function $V(\mathbf{x},t) \in C_0^{2,1}(U \times \mathbf{R}_+)$, with $V(0,t) = 0$, and a $\mu \in \mathcal{K}_\infty$ such that:*

(i) $\mu(\|\mathbf{x}\|) \leq V(\mathbf{x},t)$ for all $(\mathbf{x},t) \in U \times \{t > 0\}$,

(ii) $LV(\mathbf{x},t) \leq 0$ for all $\mathbf{x} \neq 0$.

Then the trivial solution of (1) *is* SiP *(see Definition 2.3).*

If we replace condition (ii) in the theorem above with $LV(\mathbf{x},t) \leq \mu_2(\|\mathbf{x}\|)$ for some $\mathcal{K}_\infty$ function μ_2, then the trivial solution of (1) is not just stable in probability but *asymptotically stable in probability* (see Definition 2.4).

Note that in the statement of Itô's formula in Theorem 2.7 we require V to be differentiable at $\mathbf{x} = 0$. This is not a problem and the function V in Theorem 2.8 is still a supermartingale and equation (20) still holds. The reason for this is that the set $\mathbf{x} \neq 0$ is *inaccessible*.

Definition 2.9. *(Inaccessible set) Let $X(t)$ be a solution of the SDE in* (1)*. Let $\tau^\Gamma = \inf\{t \mid X(t) \in \Gamma\}$ be the first hitting time of the set Γ. We say that the set Γ is* inaccessible *to the process $X(t)$ if*

$$\mathbf{P}\{\tau^\Gamma < \infty\} = 0. \tag{23}$$

It is known that for SDEs that satisfy the *usual conditions* and have an equilibrium at the origin, that the origin point $\mathbf{x} = 0$ is inaccessible to any sample path $X^{\mathbf{x}_0}$ for $\mathbf{x}_0 \neq 0$, see e.g. [Khasminskii 2012, Lemma 5.3]. Itô's formula is then applicable to functions V as in Theorem 2.8 since they fail to be differentiable at an inaccessible point $\mathbf{x} = 0$. For further information see [Khasminskii 2012, Chpt. 5.2].

Seeing the different approaches for these Lyapunov functions in the literature and especially in light of the result that Khasminskii obtained, we looked at results that Xuerong Mao got in his book [Mao 2008, Thm. 3.3]. Specifically, Mao writes the following theorem:

Theorem 2.10. *Assume that there exists a function $V \in C^{2,1}(\mathbf{R}^d \times [t_0,\infty[)$ and constants $p > 0$, $c_1 > 0$, $c_2 \in \mathbf{R}$, $c_3 \geq 0$, such that for all $\mathbf{x} \neq 0$ and $t \geq t_0$,*

(i) $c_1\|\mathbf{x}\|^p \leq V(\mathbf{x},t)$,

(ii) $LV(\mathbf{x},t) \leq c_2 V(\mathbf{x},t)$,

(iii) $\|(\nabla_\mathbf{x} V(\mathbf{x},t))\, g(\mathbf{x},t)\|^2 \geq c_3 V^2(\mathbf{x},t)$.

Then

$$\limsup_{t\to\infty} \frac{1}{t} \log(X^{\mathbf{x}_0}(t)) \leq -\frac{c_3 - 2c_2}{2p} \quad a.s. \tag{24}$$

for all $\mathbf{x}_0 \in \mathbf{R}^d$. In particular, if $c_3 > 2c_2$, the trivial solution of (1) *is* almost surely exponentially stable*, see Definition 2.6.*

In our paper [Bjornsson and Hafstein 2018a] we generalise this theorem, and another related one, to functions V like above that are of class $C_0^{2,1}$, i.e. not differentiable at $\mathbf{x} = 0$. We get that the above theorem still holds without further modifications if we drop the requirement that V be differentiable at the origin.

3 Local and non-local Lyapunov functions

There is a technical problem that arises when trying to solve the *partial differential equation* (PDE) $LV \leq 0$ numerically on some domain $\mathscr{A}$ that contains the origin. The operator L is elliptic on a domain $\mathscr{A}$ if and only if the matrix $g(\mathbf{x},t)g(\mathbf{x},t)^\top$ from equation (16) is positive definite, i.e. all of its eigenvalues are strictly positive, see [Gilbarg and Trudinger 2001, Chpt. 3]. Furthermore, L is strictly elliptic if the eigenvalues of the matrix $g(\mathbf{x},t)g(\mathbf{x},t)^\top$ are larger than λ for some constant $\lambda > 0$ on the entire domain $\mathscr{A}$.

Since $g(\mathbf{0},t) = \mathbf{0}$ we see that the operator L can not be elliptic on a domain that contains the origin, and therefore the standard theory of elliptic PDEs does not apply. By cutting out a small closed set B that contains the origin we can frequently guarantee the strict ellipticity of the operator L on the domain $\mathscr{A} \setminus B$, and therefore the existence and uniqueness of the solution to the PDE $LV(\mathbf{x},t) = h < 0$ on the domain $\mathscr{A} \setminus B$. For this purpose we introduce the notion of a non-local Lyapunuv function. By solving this PDE instead of $LV(\mathbf{x},t) \leq 0$ we can often guarantee that our solution V is a true Lyapunov function instead of just an approximation to a solution. This is possible even in the case that L is not elliptic, e.g. if $g(\mathbf{x},t)$ does not have full rank.

Since we have removed the equilibrium point from the domain of the non-local Lyapunov function, we need to *patch-up* the hole around the origin to check if the null solution is actually stable. For this purpose we calculate what we call a local Lyapunov function for the system in a domain $\mathscr{B}$ containing B, using different methods. We then use these two Lyapunov functions to estimate the probabilistic basin of attraction (*BOA*, see Definition 2.5).

In this thesis project we are mostly concerned with the autonomous SDEs of the form

$$dX(t) = f(X(t))dt + g(X(t))dB(t). \tag{25}$$

Up until now we have presented some of the general theory of Lyapunov functions, but the results we obtained for local and non-local Lyapunov functions are for autonomous systems. We therefore assume from now on that we are working with the system in equation (25) satisfying the *usual conditions* and that we have an equilibrium at $\mathbf{x} = 0$ (see chapter 2.1).

3.1 Local Lyapunov function

Following the conventions used in the papers [Gudmundsson and Hafstein 2018] and [Bjornsson, Hafstein, et al. 2019] we say that a function W is a local Lyapunov function if it satisfies the following conditions.

Definition 3.1 (Local Lyapunov function)**.** *Consider the system* (25)*. A function* $W \in C(\mathscr{N}) \cap C^2(\mathscr{N} \setminus \{0\})$*, where* $0 \in \mathscr{N} \subset \mathbf{R}^d$ *is a domain, is called a (local) Lyapunov function for the system* (25)*, if there are functions* $\mu_1, \mu_2, \mu_3 \in \mathscr{K}_\infty$*, such that W fulfils the properties* :

(i) $\mu_1(\|\mathbf{x}\|) \leq W(\mathbf{x}) \leq \mu_2(\|\mathbf{x}\|)$ *for all* $\mathbf{x} \in \mathscr{N}$*,*

(ii) $LW(\mathbf{x}) \leq -\mu_3(\|\mathbf{x}\|)$ *for all* $\mathbf{x} \in \mathscr{N} \setminus \{0\}$.

It is difficult to solve the system above for general functions f and g, but paper [Bjornsson, Giesl, et al. 2018] introduces a method to find such a local Lyapunov function. The idea is to consider the linearisation of the system in equation (25), i.e.

$$d\tilde{X}(t) = F\tilde{X}(t)dt + \sum_{q=1}^{Q} G^q \tilde{X}(t)dB_q(t), \tag{26}$$

with $F = Df(0)$ and G^q is the Jacobian of the q-th column of g at $\mathbf{0}$, and $dB_q(t)$ is the q-th component of $dB(t)$. We can look for a local Lyapunov function W for the system in equation (26) of the form $W(x) = \|\mathbf{x}\|_Q^{\frac{p}{2}} = \left(\mathbf{x}^\top Q\mathbf{x}\right)^{\frac{p}{2}}$ where Q is a symmetric positive definite matrix and $\|\cdot\|_Q$ denotes the Q-norm. Now this function W will also be a local Lyapunov function for the original system, in equation (25), in some domain around the origin. The paper provides an explicit lower bound on the domain and shows that W is a local Lyapunov function for the original system in a domain

$$\mathscr{D} = \{\mathbf{x} \in \mathbf{R}^d \ : \ \|\mathbf{x}\|_Q \leq \rho\}, \tag{27}$$

where $\rho > 0$ is obtained by explicit bounds on the second derivatives and Taylor remainders of the functions f and g. For the full details see [Bjornsson, Giesl, et al. 2018, Thm. 3.4].

There have been other developments for calculating local Lyapunov functions, in the thesis [Bjarkason 2022] a method to formulate the local Lyapunov function conditions as a *bilinear matrix inequality* (BMI) is presented. Additionally they present a program that generates this BMI, and attempts to solve it, automatically. For a different viewpoint on studying the dynamics of the stochastic differential equation around the equilibrium see [Arnold 2003].

3.2 Non-local Lyapunov function

Next we introduce a non-local Lyapunov function in the set $\mathscr{U}$ as in [Gudmundsson and Hafstein 2018]. A non-local Lyapunov function satisfies $LV < 0$ in a large set $\mathscr{U}$, not including a small neighborhood $\mathscr{B}$ of the equilibrium.

Definition 3.2 (Non-local Lyapunov function)**.** *Let* $\mathscr{A}, \mathscr{B} \subset \mathbf{R}^d$*,* $\mathscr{B} \subset \mathscr{A}^\circ$*, be simply connected compact neighbourhoods of the origin with* C^2 *boundaries and set* $\mathscr{U} := \mathscr{A} \setminus \mathscr{B}^\circ$*. A function* $V \in C^2(\mathscr{U})$ *for the system* (25) *such that*

(i) $b \leq V(\mathbf{x}) \leq a$ *for all* $\mathbf{x} \in \mathscr{U}$*,* $V^{-1}(b) = \partial\mathscr{B}$*,* $V^{-1}(a) = \partial\mathscr{A}$ *with* $b < a$*, and*

(ii) $LV(\mathbf{x}) < 0$ *for all* $\mathbf{x} \in \mathscr{U}$*,*

is called a non-local Lyapunov function *for the system* (25)*. We refer to* $\partial\mathscr{A}$ *as the outer boundary of* $\mathscr{U}$ *and* $\partial\mathscr{B}$ *as the inner boundary of* $\mathscr{U}$*.*

This is a slight modification of the original definition where we have replaced the original constants 0 and 1, by b and a respectively.

To find a non-local Lyapunov function V using a computer we simply look for numerical solutions of the PDE

$$\begin{cases} LV(\mathbf{x}) = r(x) & \text{for } \mathbf{x} \in \mathscr{U}, \\ V(\mathbf{x}) = a & \text{for } \mathbf{x} \in \partial\mathscr{A}, \\ V(\mathbf{x}) = b & \text{for } \mathbf{x} \in \partial\mathscr{B}, \end{cases} \tag{28}$$

where we usually set $b = 0$, $a = 1$, and $r(x) = -h$ for a sufficiently small positive constant h. The method chosen to obtain these numerical solutions for this thesis project was to use *meshless collocation* using *radial basis functions* (RBF). This method is described in the next section.

A Lyapunov function as in equation (28) lets us estimate the probability that a solution started in a point $\mathbf{x}_0 \in \mathscr{U}$ exits throught the inner boundary $\partial\mathscr{B}$ before the outer boundary $\partial\mathscr{A}$. Set τ as the first exit time from $\mathscr{U}$, then by equation (20) we have

$$V(\mathbf{x}_0) \geq \mathbf{E}[V(X^{\mathbf{x}_0}(\tau))] = a\mathbf{P}\{X^{\mathbf{x}_0}(\tau) \in \partial\mathscr{A}\} + b\mathbf{P}\{X^{\mathbf{x}_0}(\tau) \in \partial\mathscr{B}\}, \tag{29}$$

since $\mathbf{P}\{\tau < \infty\} = 1$, see [Gudmundsson and Hafstein 2018, p. Lm. 10]. For the case $a = 1$ and $b = 0$, this reduces to $V(\mathbf{x}_0) \geq \mathbf{P}\{X^{\mathbf{x}_0}(\tau) \in \partial\mathscr{A}\}$.

3.3 Estimating the basin of attraction

The next theorem, from paper [Gudmundsson and Hafstein 2018] shows how the local and non-local Lyapunov functions taken together give us information about the γ-BOA of the autonomous system in equation (25). We estimate the probability that a solution that starts in the set $\mathscr{U}$ leaves the set through the *inner*-boundary using the non-local Lyapunov function, and then using the local Lyapunov function we estimate the probability that they converge to the origin. The combined probability can then be bounded from below by the γ given in the next theorem.

Theorem 3.3. *Consider the system in* (25) *and assume that there exists a local Lyapunov function* $W : \mathscr{N} \to \mathbf{R}_+$ *as in definition 3.1 and a non-local Lyapunov function* $V : \mathscr{U} \to \mathbf{R}_+$ *as in Definition 3.2 for the system. Let* $0 < \beta < 1$ *and* $b < \lambda < \alpha < a$ *and the set* $\mathscr{B}$ *from Definition 3.2 such that*

$$W^{-1}(W_{max}) \subset V^{-1}([b,\lambda]) \quad \textit{and} \quad \partial\mathscr{B} = V^{-1}(b) \subset W^{-1}([0,\beta W_{max}]). \tag{30}$$

Then the set $V^{-1}([b,\alpha]) \cup \mathscr{B}$ *is a subset of the* γ*-BOA of the origin, where*

$$\gamma = \frac{(a-\alpha)(1-\beta)}{a-b-\beta(a-\lambda)}. \tag{31}$$

4 Meshless collocation

4.1 Overview

Recall that to calculate a non-local Lyapunov function we essentially, see equation (28), have to solve a PDE of the form

$$\begin{cases} LV(\mathbf{x}) = r(\mathbf{x}) & \text{for } \mathbf{x} \in \mathcal{U}, \\ V(\mathbf{x}) = c(\mathbf{x}) & \text{for } \mathbf{x} \in \partial\mathcal{U}, \end{cases} \tag{32}$$

where r and c are some suitable functions, $\mathcal{U} \subset \mathbf{R}^d$ is a bounded domain with smooth boundary $\partial\mathcal{U}$ and L is the differential operator given in (16). To solve this PDE numerically we use so called *Meshless collocation*. The method is an approach to finding norm minimal solutions to an interpolation problem in a *Reproducing Kernel Hilbert Space* (RKHS), see [Wendland 2005, Def. 10.1] which in our case is a Sobolev space [Giesl 2007a, p. Lm. 3.13].

This interpolation problem will guarantee that V satisfies the PDE (32) at the given collocation points, and furthermore this interpolation problem has a solution [Iske 2018, Thm 8.3] and [Giesl 2007a, Prop. 3.20], even when there is no solution to (32).

4.2 Interpolation problem

As mentioned in the introduction our goal is to solve, approximately, the PDE (32). Our solution will be a function in a RKHS, which is a Hilbert space H of functions $\mathcal{U} \to \mathbf{R}$ with inner product $\langle \cdot, \cdot \rangle_H$ and a kernel $\Phi : \mathcal{U} \times \mathcal{U} \to \mathbf{R}$ such that

(i) $\Phi(\cdot, \mathbf{x}) \in H$ for all $\mathbf{x} \in \mathcal{U}$,

(ii) $g(\mathbf{x}) = \langle g, \Phi(\cdot, \mathbf{x}) \rangle_H$ for all $\mathbf{x} \in \mathcal{U}$ and $g \in H$.

In our thesis project we work with radially symmetric kernels $\Phi(\mathbf{x}, \mathbf{y}) = \psi(\|\mathbf{x} - \mathbf{y}\|)$, where $\psi = \psi_{l,k}$ is a *radial basis function* (RBF) given by a Wendland function, see chapter 4.3. We choose points $X_1 = \{\mathbf{x}_1, \dots \mathbf{x}_N\} \subset \mathcal{U}$ and $X_2 = \{\xi_1, \dots, \xi_M\} \subset \partial\mathcal{U}$, and look to solve the interpolation problem

$$\begin{cases} LV(\mathbf{x}_i) = r(\mathbf{x}_i) & \text{for all } i = 1, \dots N, \\ V(\xi_i) = c(\xi_i) & \text{for all } i = 1, \dots, M. \end{cases} \tag{33}$$

The solution V to this interpolation problem is given by

$$V(\mathbf{x}) = \sum_{k=1}^{N} \alpha_k (\delta_{x_k} \circ L)^{\mathbf{y}} \psi(\|\mathbf{x} - \mathbf{y}\|) + \sum_{k=1}^{M} \alpha_{N+k} (\delta_{\xi_k} \circ L^0)^{\mathbf{y}} \psi(\|\mathbf{x} - \mathbf{y}\|), \tag{34}$$

where L^0 is the identity operator, $\delta_{\mathbf{y}}V(\cdot) = V(\mathbf{y})$ and superscript $\mathbf{y}$ denotes that the operator is applied with respect to the variable $\mathbf{y}$.

The constants α_i are determined as a solution to the linear system

$$A\alpha = \gamma, \tag{35}$$

where A, called the *interpolation matrix*, is the symmetric matrix

$$A = \begin{bmatrix} B & C \\ C^T & D \end{bmatrix} \tag{36}$$

and the matrices $B = (b_{jk})_{j,k=1,\dots,N}$, $C = (c_{jk})_{j=1,\dots,N,k=1,\dots,M}$ and $D = (d_{jk})_{j,k=1,\dots,M}$ have elements

$$\begin{aligned} b_{jk} &= (\delta_{\mathbf{x}_j} \circ L)^{\mathbf{x}}(\delta_{\mathbf{x}_k} \circ L)^{\mathbf{y}}\psi(\|\mathbf{x}-\mathbf{y}\|), \\ c_{jk} &= (\delta_{\mathbf{x}_j} \circ L)^{\mathbf{x}}(\delta_{\xi_k} \circ L^0)^{\mathbf{y}}\psi(\|\mathbf{x}-\mathbf{y}\|), \\ d_{jk} &= (\delta_{\xi_j} \circ L^0)^{\mathbf{x}}(\delta_{\xi_k} \circ L^0)^{\mathbf{y}}\psi(\|\mathbf{x}-\mathbf{y}\|). \end{aligned}$$

The vector γ has components given by

$$\begin{aligned} \gamma_j &= r(\mathbf{x}_j), \quad 1 \le j \le N, \\ \gamma_{j+N} &= c(\xi_j), \quad 1 \le j \le M. \end{aligned}$$

Using Wendland functions as our radial basis function ensures that the interpolation matrix A in equations (35, 36) is symmetric and positive definite. Since these Wendland functions have compact support the matrix A is also sparse. There are other choices for radial basis functions for the interpolation problem e.g. Gaussian $\psi(r) = \exp(-r^2)$, but this function does not have a compact support.

4.3 Wendland functions

The Wendland functions, see [Wendland 2017; Wendland 2005; Giesl 2007a], are compactly supported radial basis functions that are polynomials on their support, which makes computations with them simple. They are a family of functions depending on two parameters $l,k \in \mathbf{N}_0$, using the same numbering scheme as used in [Giesl 2007a], defined by the recursive relations:

$$\psi_{l,0}(r) = [(1-r)_+]^l \tag{37}$$

and

$$\psi_{l,k+1}(r) = C_{l,k+1}\int_r^1 t\,\psi_{l,k}(t)\mathrm{d}t, \tag{38}$$

where $(1-r)_+ := \max\{1-r, 0\}$ and $C_{l,k+1} \neq 0$ is a constant. Therefore these functions also satisfy the relation

$$-C_{l,k+1}\psi_{l,k}(r) = \frac{\frac{d}{dr}\psi_{l,k+1}(r)}{r}. \tag{39}$$

For interpolation using a particular Wendland function as the base function, the value of the constant $C_{l,k+1} \neq 0$ is not of importance because the Wendland function appears linearly on both sides of a linear equation. Therefore, one can just fix values that are convenient for the problem at hand and we will do this in the following section. However, when solving collocation problems, we apply a differential operator, see equations (35) and (36), so we get terms involving both the original Wendland function and its derivatives. The derivative of a Wendland function can be written in terms of a lower order Wendland function, using equation (39), and when doing this it is necessary to keep track of the constants $C_{l,k+1}$ for the derivatives. It is only the constant for the base function that can be chosen arbitrarily. After the choice has been made, we must keep track of it through all calculations.

The Wendland functions have several important properties, cf. e.g. [Giesl 2007b, Prop. 3.10]:

(i) $\psi_{l,k}(r)$ is a polynomial of degree $l+2k$ for $r \in [0,1]$ and $\mathrm{supp}(\psi_{l,k}) = [0,1]$.

(ii) The radial function $\Psi(\mathbf{x}) := \psi_{l,k}(\|\mathbf{x}\|)$ is C^{2k} at 0.

(iii) $\psi_{l,k}$ is C^{k+l-1} at 1.

Frequently we fix the parameter $l := \left\lfloor \frac{d}{2} \right\rfloor + k + 1$, where d is the spacial dimension we are working in, and a constant $c > 0$ to fix the support. By the properties stated above, the radial function $\Psi(\mathbf{x}) := \psi_{l,k}(c\|\mathbf{x}\|)$ is then a C^{2k} function with $\mathrm{supp}(\Psi) = \mathscr{B}^d(0, c^{-1}) \subset \mathbf{R}^n$, where $\mathscr{B}^n(0, c^{-1})$ is the closed n-dimensional ball around the origin with radius c^{-1}.

4.4 Verification

By using meshless collocation we obtain a function V that satisfies the PDE at each collocation point, see equation (33).

Let $\mathscr{A}, \mathscr{B}$, and $\mathscr{U}$ be as in the definition of non-local Lyapunov functions (Definition 3.2) and let $V(x)$ be a numerical approximation obtained using meshless collocation as described in previous sections. Now, by [Bjornsson, Hafstein, et al. 2019, Theorem 4.3] if

$$\nu := \max_{y \in Y_{\mathscr{U}}} LV(y) + C_u \frac{d^2}{4} h^2 < 0, \tag{40}$$

then V is a true non-local Lyapunov function for the system. Here d is the dimension of the system, $h > 0$ is a parameter controlling the density of the evaluation grid, and $Y_{\mathscr{U}}$ is the evaluation grid that covers $\mathscr{U}$. Finally the constant $C_{\mathscr{U}}$ is an upper estimate on the second order derivatives of our function LV, obtained using Taylor-type estimates, for further details see [Bjornsson, Hafstein, et al. 2019] and [Mohammed and Giesl 2015].

This means that the equation $LV(\mathbf{x}) < 0$ holds for any point $\mathbf{x} \in \mathscr{U}$. This, along with a local Lyapunov function W as described in (Definition. 3.1) allows us to get rigorous estimate of the stochastic γ-basin of attraction for our SDE.

This method is described in detail in paper [Bjornsson, Hafstein, et al. 2019] which is the main result of this thesis project.

4.5 Computational challenges

To compute a Lyapunov function using the meshless collocation discussed above, a large number of evaluations of the RBF function ψ and its derivatives are necessary. When verifying the properties of the Lyapunov function we need to do even more evaluations. Therefore it is essential that these evaluations can be carried out in an efficient and accurate way.

Since we used Wendland functions as our radial basis functions, we studied different methods to evaluate these functions, which results are summarised in papers [Bjornsson and Hafstein 2019a; Bjornsson and Hafstein 2019b; Bjornsson and Hafstein 2018b (Paper IV,V and VI)]. It turned out that the most optimal way to evaluate these functions was in a specific factorised form. Previous methods, see e.g [Argáez, Hafstein, and Giesl 2017], relied on calculating the coefficients of the expanded polynomials and evaluating the polynomials using Horner's method [Burrus et al. 2003] but this turned out to be not optimal.

We instead developed an algorithm and software to calculate the Wendland functions of arbitrary degree in this factorised format, replacing tedious and error prone calculations by hand. This method and the corresponding software library is summarised in (Paper IV). The software library is available at:

`https://gitlab.com/hjortur/wendland-function-generator`.

5 Contributions

Here is a list of the papers included in this thesis and a short description of their contributions.

Paper I:

Lyapunov Functions for Almost Sure Exponential Stability. The paper builds up on the work of [Mao 2008] by generalising theorems on almost sure exponential stability. In his book, X. Mao shows that a certain Lyapunov function is sufficient for a solution of a stochastic differential equation to be almost surely exponentially stable. We generalize this result to a larger class of Lyapunov functions. Chapter 2.3 provides further discussion.

Paper II:

Computation of the stochastic basin of attraction by rigorous construction of a Lyapunov function. The paper uses a method to calculate the γ basin of attraction for a stochastic differential equation by combining two different Lyapunov functions as was done in paper [Gudmundsson and Hafstein 2018]. Furthermore the paper provides a method to calculate the non-local Lyapunov function using Meshless collocation and a way to verify the properties of the resulting function. This gives us rigorous lower bounds on the γ basin of attraction.

Paper III:

Local Lyapunov Functions for Nonlinear Stochastic Differential Equations by Linearization. The paper shows how a local Lyapunov function for a stochastic differential equation can be calculated by a linearisation of the SDE. The result is an explicit lower bound on the domain where the resulting local Lyapunov function is a true Lyapunov function for the original system.

Paper IV:

Advanced algorithm for interpolation with Wendland functions. The paper builds up on previous work in [Bjornsson and Hafstein 2019b(Paper V)] and [Bjornsson and Hafstein 2018b(Paper VI)]. The paper describes an algorithm to generate Wendland functions in an efficient form and a software library that can be used to calculate non-local Lyapunov functions for both deterministic and stochastic autonomous differential equations.

Paper V:

Algorithm and Software to Generate Code for Wendland Functions in Factorised Form. The paper describes an algorithm to generate Wendland functions in a specific factorised form and describes software that generates these Wendland functions, in factorised form. This software was later improved as described in [Bjornsson and Hafstein 2019a(Paper IV)].

Paper VI:

Verification of a Numerical Solution to a Collocation Problem. The paper describes some of the computational challenges when trying to verify the properties of a non-local Lyapunov function calculated using meshless collocation of Wendland radial basis functions. Different methods to evaluate Wendland functions were compared and a specific factorised form of Wendland functions was found to be the most efficient, both in terms of speed of evaluation and numerical accuracy.

6 Conclusions

In this dissertation we have given an overview of the concept of stability of stochastic differential equations. We described Lyapunov functions and their application for analysing the stability of an equilibrium for a stochastic system, and looked at meshless collocation, which is a numerical method for calculating such Lyapunov functions.

We have made contributions in a few different, but related areas of this project. First of all we have generalized a previous result on Lyapunov functions for almost sure exponential stability to a larger class of functions.

We have also contributed to a method to calculate non-local Lyapunov functions numerically for the purpose of estimating the γ basin of attraction for autonomous stochastic differential equations. A meshless collocation method delivers us a non-local Lyapunov function in the form of a sum of Wendland functions, and computational challenges we faced in this project led us to explore various methods of evaluating Wendland functions.

We found an efficient method to evaluate Wendland functions and designed an algorithm that generates Wendland functions of arbitrary degree in an efficient form. We developed a software library that calculates Lyapunov functions using Wendland functions in an efficient form, for both stochastic and deterministic systems.

Lastly, we contributed to a method of calculating local Lypunov functions for stochastic differential systems using *linearisation*.

Paper I

Lyapunov Functions for Almost Sure Exponential Stability

Bjornsson, H. and Hafstein, S.

Dynamical Systems in Theoretical Perspective, Springer Proceedings in Mathematics and Statistics 248

Lyapunov Functions for Almost Sure Exponential Stability

Hjortur Björnsson and Sigurdur Freyr Hafstein

Abstract We present a generalization of results obtained by X. Mao in his book "Stochastic Differential Equations and Applications" (2008). When studying what Mao calls "almost sure exponential stability", essentially a negative upper bound on the almost sure Lyapunov exponents, he works with Lyapunov functions that are twice continuously differentiable in the spatial variable and continuously differentiable in time. Mao gives sufficient conditions in terms of such a Lyapunov function for a solution of a stochastic differential equation to be almost surely exponentially stable. Further, he gives sufficient conditions of a similar kind for the solution to be almost surely exponentially unstable. Unfortunately, this class of Lyapunov functions is too restrictive. Indeed, R. Khasminskii showed in his book "Stochastic Stability of Differential Equations" (1979/2012) that even for an autonomous stochastic differential equation with constant coefficients, of which the solution is stochastically stable and such that the deterministic part has an unstable equilibrium, there cannot exists a Lyapunov function that is differentiable at the origin. These restrictions are inherited by Mao's Lyapunov functions. We therefore consider Lyapunov functions that are not necessarily differentiable at the origin and we show that the sufficiency conditions Mao proves can be generalized to Lyapunov functions of this form.

Keywords Almost sure exponential stability · Lyapunov function · Almost sure Lyapunov exponent

1 Introduction

Lyapunov methods, as first described in [1], have been widely used to study the behaviour of various dynamical systems, both real-world examples or purely theo-

H. Björnsson (✉) · S. F. Hafstein
University of Iceland, The Science Institute, Dunhagi 5, 107 Reykjavík, Iceland
e-mail: hjb6@hi.is

S. F. Hafstein
e-mail: shafstein@hi.is

J. Awrejcewicz (ed.), *Dynamical Systems in Theoretical Perspective*,
Springer Proceedings in Mathematics & Statistics 248,
https://doi.org/10.1007/978-3-319-96598-7_5

retical ones. This is a very active field due to the complicated dynamics exhibited in several real-world systems, as for example the wobblestone model presented in [2]. Other specific examples include the dynamics of the double [3] or triple pendulum [4, 5], where Lyapunov exponents were used to study the chaotic behavior of the systems. Often it is necessary to modify a dynamical system to include either an unknown force, or to consider the perturbation of the system by some noise, and that is where stochastic differential equations (SDEs) are commonly used. Here in this paper, we are concerned with applying Lyapunov methods for classical dynamical systems to the stochastic framework, as done by Khasminskii [8].

We work in a complete probability space $(\Omega, \mathscr{F}, \mathbf{P})$ with a right continuous filtration $\{\mathscr{F}_t\}_{t\geq 0}$ and such that $\mathscr{F}_0$ contains all $\mathbf{P}$ null sets. In this paper we consider strong solutions of the d-dimensional stochastic differential equation

$$\mathrm{d}x(t) = f(x(t), t)\mathrm{d}t + g(x(t), t)\mathrm{d}B(t) \quad \text{on } t \geq t_0 \tag{1}$$

where $B(t)$ is an m-dimensional Brownian motion. For a more detailed description of the setting cf. [9, Sec. 2.1]. We assume that for any given initial value $x(t_0) = x_0 \in \mathbf{R}^d$ there exists a unique global solution, denoted by $t \mapsto x(t, t_0, x_0)$, with continuous sample paths. Furthermore, we assume that

$$f(0, t) = 0 \quad \text{and} \quad g(0, t) = 0 \quad \text{for all } t \geq t_0.$$

Sufficient condition for the existence of such solutions are, for example, given by the following statement, cf. [9, Thm. 2.3.6].

For any real number $T > 0$ and integer $n \geq 1$, the following hold true:

1. There exists a positive constant $K_{T,n}$, such that for all $t \in [t_0, T]$ and all $x, y \in \mathbf{R}^d$ with $|x| \vee |y| \leq n$,

$$|f(x,t) - f(y,t)|^2 \bigvee |g(x,t) - g(y,t)|^2 \leq K_{T,n}|x-y|^2.$$

2. There exists a positive constant K_T, such that for all $(x, t) \in \mathbf{R}^d \times [t_0, T]$

$$x^\top f(x,t) + \frac{1}{2}|g(x,t)|^2 \leq K_T(1 + |x|^2).$$

Here $|\cdot|$ is the Euclidean norm and the symbols $\wedge$ and $\vee$ are defined to be the minimum and the maximum respectively:

$$a \wedge b := \min(a, b) \quad \text{and} \quad a \vee b := \max(a, b).$$

Corresponding to the initial value $x(t_0) = 0$, we have the solution $x(t) = 0$ for all t. This solution is called the trivial solution. In this paper we are studying the stability of the trivial solution and, more specifically, when it is almost surely exponentially stable. This definition is taken from Mao's book [9, Def. 4.3.1], see also e.g. [6, 11].

Definition 1 The trivial solution of (1) is said to be *almost surely exponentially stable* if

$$\limsup_{t\to\infty} \frac{1}{t} \log |x(t, t_0, x_0)| < 0$$

almost surely, for all $x_0 \in \mathbf{R}^d$.

First, we clarify some of the notation used in the paper. For our purposes all integrals in this paper of the form $\int \cdot \, \mathrm{d}B(s)$ are to be interpreted in the Itô sense. We write $b_n \uparrow a$ if the sequence b_n is increasing and has limit a. We denote by $\mathscr{L}^2(\mathbf{R}_+, \mathbf{R}^{d\times m})$ the family of all $(d \times m)$-matrix valued measurable $\{\mathscr{F}_t\}_{t\geq 0}$-adapted processes $f = \{f(t)\}_{t\geq 0}$ such that

$$\int_0^T |f(t)|^2 \mathrm{d}t < \infty \quad \text{a.s. for every } T > 0$$

and by $\mathscr{M}^2(\mathbf{R}_+, \mathbf{R}^{d\times m})$ the family of all processes $f \in \mathscr{L}^2(\mathbf{R}_+, \mathbf{R}^{d\times m})$ such that

$$\mathbf{E}\left\{\int_0^T |f(t)|^2 \mathrm{d}t\right\} < \infty \quad \text{for every } T > 0.$$

Here $\mathbf{E}$ denotes the expectation and a.s. is an abbreviation for *almost surely* as usual. Let $f \in \mathscr{M}^2(\mathbf{R}_+, \mathbf{R}^{d\times m})$ and consider the process

$$M_t = \int_0^t f(s) \mathrm{d}B(s)$$

then there exists a t-continuous version of the process M_t. Furthermore the process is $\{\mathscr{F}_t\}$ adapted and is a square integrable martingale [10, Thm. 3.2.5]. By the preceding remark, we will assume that $\int_0^t f(s)\mathrm{d}B(s)$ refers to a t-continuous version of the integral.

A sequence of stopping times $\{\tau_k\}_{k\geq 1}$ is called a *localization* if it is non-decreasing and $\tau_k \uparrow \infty$ almost surely. A right continuous adapted process $M = \{M_t\}_{t\geq 0}$ is called a *local martingale* if there exists a localization $\{\tau_k\}_{k\geq 1}$ such that the process $\{M_{\tau_k \wedge t} - M_0\}_{t\geq 0}$ is a martingale for every $k \geq 1$. We denote the quadratic variation of a continuous local martingale M by $\langle M, M\rangle_t$, which is the unique continuous adapted process of finite variation, such that $\{M_t^2 - \langle M, M\rangle_t\}_{t\geq 0}$ is a continuous local martingale which takes the value 0 at $t = 0$.

Let M_t be a continuous martingale of the form

$$M_t = \int_0^t f(s) \mathrm{d}B(s).$$

Then the quadratic variation $\langle M, M\rangle_t$ is given by

$$\langle M, M\rangle_t = \int_0^t |f(s)|^2 \mathrm{d}s$$

almost surely [9, Thm. 1.5.14].

Let τ be a stopping time and let $[[0, \tau]]$ be the stochastic interval

$$[[0, \tau]] = \{(t, \omega) \in \mathbf{R}_+ \times \Omega : 0 \leq t \leq \tau(\omega)\}.$$

We now list a few facts needed to give rigid proofs of our results. For any $f \in \mathscr{L}^2(\mathbf{R}_+, \mathbf{R}^{d\times m})$ we can define a sequence of stopping times

$$\tau_n := n \wedge \inf\{t \geq 0 : \int_0^t |f(s)|^2 \mathrm{d}s \geq n\}.$$

It is easy to see that $\tau_n \uparrow \infty$ almost surely. Let I_A, for $A \subset \mathbf{R}_+ \times \Omega$, be the indicator function, that is $I_A(x) = 1$ if $x \in A$ and zero otherwise. Then we can define the process $g_n(t) = f(t)I_{[[0,\tau_n]]}(t)$. We see that $g_n \in \mathscr{M}^2(\mathbf{R}_+, \mathbf{R}^{d\times m})$ so the integral

$$J_n(t) = \int_0^t g_n(s)\mathrm{d}B(s)$$

is a martingale. That is to say, the process

$$J(t) := \int_0^t f(s)\mathrm{d}B(s)$$

is a local martingale with localization $\{\tau_n\}$, since for any $n \geq 1$

$$J(t\wedge\tau_n) = \int_0^{t\wedge\tau_n} f(s)\mathrm{d}B(s) = \int_0^t f(s)I_{[[0,\tau_n]]}(s)\mathrm{d}B(s) = \int_0^t g_n(s)\mathrm{d}B(s) = J_n(t)$$

is a martingale.

In his book [9], Mao considers Lyapunov functions $V(x, t) \in C^{2,1}(\mathbf{R}^d \times [t_0, \infty[; \mathbf{R}_+)$ where $C^{2,1}(\mathbf{R}^d \times [t_0, \infty[\,; \mathbf{R}_+)$ is the set of all continuous functions $\mathbf{R}^d \times [t_0, \infty[\to \mathbf{R}_+$, which are continuously differentiable twice in the first coordinate x, with $x \in \mathbf{R}^d$, and once in t with $t \in [t_0, \infty[$. Now define a differential operator L associated with (1) by

$$L = \frac{\partial}{\partial t} + \sum_{i=1}^d f_i(x, t)\frac{\partial}{\partial x_i} + \frac{1}{2}\sum_{i,j=1}^d [g(x, t)g^\top(x, t)]_{ij}\frac{\partial^2}{\partial x_i \partial x_j}, \tag{2}$$

where $[g(x,t)g^\top(x,t)]_{ij}$ is the (i,j)-th component of the $(d \times d)$-matrix $gg^\top$ at (x,t). If $x(t)$ is a solution of (1) then by Itô's formula

$$dV(x(t),t) = LV(x(t),t)\mathrm{d}t + V_x(x(t),t)\mathrm{d}B(t)$$

where $V_x \in \mathbf{R}^{1\times d}$ is the derivative (gradient) of V with respect to x.

Khasminskii showed in his book [8, p. 154–155] that even for SDEs with constant coefficients there cannot exist Lyapunov functions that are differentiable at 0 unless the deterministic part of the SDE is already stable. Therefore we extend the results from Mao's book using the larger class of functions $C_0^{2,1}(\mathbf{R}^d \times [t_0,\infty[\,;\mathbf{R}_+)$ which are continuous, continuously differentiable in t, and twice continuously differentiable in x except at the point $x=0$.

Below is a theorem taken from Mao's book [9] which we will use in the next chapter. For completeness we give a more worked out proof than in the book.

Theorem 1 [9, Thm. 1.7.4]
Let $g = (g_1,\dots,g_m) \in \mathscr{L}^2(\mathbf{R}_+, \mathbf{R}^{d\times m})$, and T,α,β be any numbers ≥ 0. Then

$$\mathbf{P}\left\{\sup_{0\le t\le T}\left[\int_0^t g(s)\mathrm{d}B(s) - \frac{\alpha}{2}\int_0^t |g(s)|^2\mathrm{d}s\right] > \beta\right\} \le \mathrm{e}^{-\alpha\beta}. \tag{3}$$

Proof Define the process

$$x(t) = \alpha\int_0^t g(s)\mathrm{d}B(s) - \frac{\alpha^2}{2}\int_0^t |g(s)|^2\mathrm{d}s$$

and for every integer $n \geq 1$, define the stopping time

$$\tau_n = \inf\left\{t \ge 0 : \left|\int_0^t g(s)\mathrm{d}B(s)\right| + \int_0^t |g(s)|^2\mathrm{d}s \ge n\right\}.$$

Then τ_n is a localization, and since

$$\begin{aligned}|x_n(t)| &\le \alpha\left|\int_0^t g(s)I_{[[0,\tau_n]]}(s)\mathrm{d}B(s)\right| + \frac{\alpha^2}{2}\int_0^t |g(s)|^2 I_{[[0,\tau_n]]}\mathrm{d}s\\ &\le \alpha n + \frac{\alpha^2}{2}n = n\frac{2\alpha+\alpha^2}{2}\end{aligned}$$

we see that the process $x_n(t) := x(t\wedge\tau_n)$ is bounded.

Apply Itô's formula to $\exp(x_n(t))$ and we obtain

$$\begin{aligned}\exp(x_n(t)) &= 1 + \int_0^t \exp(x_n(s))\mathrm{d}x_n(s) + \frac{\alpha^2}{2}\int_0^t \exp(x_n(s))|g(s)|^2 I_{[[0,\tau_n]]}(s)\mathrm{d}s\\ &= 1 + \Bigg(\alpha \int_0^t \exp(x_n(s))g(s)I_{[[0,\tau_n]]}(s)\mathrm{d}B(s)\\ &\qquad -\frac{\alpha^2}{2}\int_0^t \exp(x_n(s))|g(s)|^2 I_{[[0,\tau_n]]}(s)\mathrm{d}s\Bigg)\\ &\qquad + \frac{\alpha^2}{2}\int_0^t \exp(x_n(s))|g(s)|^2 I_{[[0,\tau_n]]}(s)\mathrm{d}s\\ &= 1 + \alpha \int_0^t \exp(x_n(s))g(s)I_{[[0,\tau_n]]}(s)\mathrm{d}B(s).\end{aligned}$$

The term inside the integral is bounded by $n^2 \frac{2\alpha+\alpha^2}{2}$ almost surely, therefore the process $\exp(x_n)$ is a non negative martingale with $\mathbf{E}\{\exp(x_n(T))\} = 1$, for all $n \geq 1$. This construction is known as the Doléans-Dade exponential of the local martingale $Y_t := \int_0^t g(s)\mathrm{d}B(s)$, see [8, Thm. 26.8].
By Doob's martingale inequality [9, Thm. 1.3.8] we get that

$$\mathbf{P}\left\{\sup_{0\leq t\leq T} \exp[x_n(t)] \geq \mathrm{e}^{\alpha\beta}\right\} \leq \mathrm{e}^{-\alpha\beta}\mathbf{E}\{\exp(x_n(T))\} = \mathrm{e}^{-\alpha\beta}.$$

Then it follows that

$$\mathbf{P}\left\{\sup_{0\leq t\leq T} \frac{x_n(t)}{\alpha} > \beta\right\} \leq \mathrm{e}^{-\alpha\beta}.$$

Since this inequality holds for any $n \geq 1$, and

$$\lim_{n\to\infty} x_n(t) = x(t)$$

almost surely, we get by the dominated convergence theorem that

$$\mathbf{P}\left\{\sup_{0\leq t\leq T} \frac{x(t)}{\alpha} > \beta\right\} \leq \mathrm{e}^{-\alpha\beta}$$

and the proof is complete. □

2 The Theorems and Their Proofs

As discussed above, we state two theorems from Mao's book [9], more specifically Theorem 4.3.3 and Theorem 4.3.5, except we allow the Lyapunov functions V to be in the $C_0^{2,1}$ space instead of the too restrictive space $C^{2,1}$, like Mao does. The difference

is that in the former space the functions are not required to be differentiable at the origin, while functions in the latter one are smooth everywhere. As already explained before, this makes the results much more relevant and useful.
First we state and proof Theorem 4.3.3 from [9] with the weaker conditions. Note that, like above, $V_x \in \mathbf{R}^{1\times d}$ is the derivative (gradient) of V with respect to x.

Theorem 2 (advancement of Thm. 4.3.3 in Mao)
Assume there exists a function $V \in C_0^{2,1}(\mathbf{R}^d \times [t_0, \infty); \mathbf{R}_+)$ and constants $p > 0$, $c_1 > 0$, $c_2 \in \mathbf{R}$, $c_3 \geq 0$, such that for all $x \neq 0$ and $t \geq t_0$:

1. $c_1|x|^p \leq V(x,t)$,
2. $LV(x,t) \leq c_2 V(x,t)$,
3. $|V_x(x,t)g(x,t)|^2 \geq c_3 V^2(x,t)$.

Then

$$\limsup_{t\to\infty} \frac{1}{t} \log|x(t; t_0, x_0)| \leq -\frac{c_3 - 2c_2}{2p} \quad a.s.$$

for all $x_0 \in \mathbf{R}^d$. In particular, if $c_3 > 2c_2$, the trivial solution of Eq. (1) is almost surely exponentially stable, see Definition 1.

The proof here mostly follows Mao's original argument, but with some modifications, since the process $M(t)$ below isn't necessarily a martingale.

Proof Clearly the inequality holds for $x_0 = 0$ since $x(t, t_0, 0) = 0$ for all t. We only need to show the inequality for all $x_0 \neq 0$. Fix any $x_0 \neq 0$ and write $x(t) := x(t; t_0, x_0)$. It is well known that 0 is an inaccessible point, cf. e.g. [9, Lemma 4.3.2], that is to say, $x(t) \neq 0$ for all $t \geq t_0$ almost surely. Thus one can apply Itô's formula and get

$$\begin{aligned}
&\log V(x(t), t)\\
&= \log V(x_0, t_0) + \int_{t_0}^{t} \frac{LV(x(s), s)}{V(x(s), s)} \mathrm{d}s + M(t) - \frac{1}{2}\int_{t_0}^{t} \frac{|V_x(x(s), s)g(x(s), s)|^2}{(V(x(s), s))^2} \mathrm{d}s\\
&\leq \log V(x_0, t_0) + c_2(t - t_0) + M(t) - \frac{1}{2}\int_{t_0}^{t} \frac{|V_x(x(s), s)g(x(s), s)|^2}{(V(x(s), s))^2} \mathrm{d}s
\end{aligned}$$

where we used condition 2 for the last inequality and

$$M(t) := \int_{t_0}^{t} \frac{V_x(x(s), s)g(x(s), s)}{V(x(s), s)} \mathrm{d}B(s).$$

We claim the process

$$h(s) := \frac{V_x(x(s), s)g(x(s), s)}{V(x(s), s)}$$

is in $\mathscr{L}^2([t_0, \infty[, \mathbf{R}^d)$. Indeed, for almost all $\omega \in \Omega$, the trajectory of $x(t)(\omega)$, $t_0 \leq t \leq T$, is a compact subset of $\mathbf{R}^d \setminus \{0\}$. Hence, for almost all ω, the function $h(s)(\omega)$

is continuous on the compact set $t_0 \leq s \leq T$ and thus bounded. Since this holds true for all T, we have $h(s) \in \mathscr{L}^2([t_0, \infty[, \mathbf{R}^d)$.

Fix an arbitrary $\varepsilon > 0$. We can now use Theorem 1 and get for all $n \in \mathbf{N}$:

$$\mathbf{P}\left\{\sup_{t_0 \leq t \leq t_0+n}\left[M(t) - \frac{\varepsilon}{2}\int_{t_0}^{t}\frac{|V_x(x(s), s)g(x(s), s)|^2}{(V(x(s), s))^2}\mathrm{d}s\right] > \frac{2}{\varepsilon}\log(n)\right\} \leq \frac{1}{n^2}$$

By the Borel Cantelli theorem, cf. e.g. [7, Thm. 3.18], there exists an $n_0(\omega) > 0$ for almost all ω, such that

$$M(t) \leq \frac{2}{\varepsilon}\log(n) + \frac{\varepsilon}{2}\int_{t_0}^{t}\frac{|V_x(x(s), s)g(x(s), s)|^2}{(V(x(s), s))^2}\mathrm{d}s.$$

for all $t_0 \leq t \leq t_0 + n$ if $n > n_0$. By condition 3,

$$\begin{aligned}
&\log V(x(t), t)\\
&\leq \log V(x_0, t_0) + c_2(t - t_0) + \frac{1}{2}(\varepsilon - 1)\int_{t_0}^{t}\frac{|V_x(x(s), s)g(x(s), s)|^2}{(V(x(s), s))^2}\mathrm{d}s + \frac{2}{\varepsilon}\log(n)\\
&\leq \log V(x_0, t_0) + c_2(t - t_0) - \frac{1}{2}(1 - \varepsilon)c_3(t - t_0) + \frac{2}{\varepsilon}\log(n)\\
&= \log V(x_0, t_0) - \frac{1}{2}((1 - \varepsilon)c_3 - 2c_2)(t - t_0) + \frac{2}{\varepsilon}\log(n)
\end{aligned}$$

for all $t_0 \leq t \leq t_0 + n$ if $n > n_0$ for almost all ω. Therefore we have for almost all ω, that

$$\frac{1}{t}\log V(x(t), t) \leq -\frac{t - t_0}{2t}[(1 - \varepsilon)c_3 - 2c_2] + \frac{\log V(x_0, t_0) + 2\log(n)/\varepsilon}{t_0 + n - 1}$$

if $t_0 + n - 1 \leq t \leq t_0 + n$ and $n > n_0$.

Fix ω and let $n \to \infty$, then

$$\limsup_{t\to\infty}\frac{1}{t}\log V(x(t), t) \leq -\frac{1}{2}((1 - \varepsilon)c_3 - 2c_2)$$

holds point-wise for almost all ω. Finally using condition 1 we have

$$\limsup_{t\to\infty}\frac{1}{t}\log|x(t)| \leq -\frac{(1 - \varepsilon)c_3 - 2c_2}{2p}$$

for almost all ω. Since $\varepsilon > 0$ was arbitrary we have the conclusion. □

Now we state and proof Theorem 4.3.5 from [9] with the weaker conditions.

Theorem 3 (advancement of Thm. 4.3.5 in Mao)
Assume that there exists a function $V \in C_0^{2,1}(\mathbf{R}^d \times [t_0, \infty); \mathbf{R}_+)$*, and constants* $p > 0$, $c_1 > 0$, $c_2 \in \mathbf{R}$, $c_3 > 0$, *such that for all* $x \neq 0$ *and* $t \geq t_0$,

1. $c_1|x^p| \geq V(x, t) > 0$,
2. $LV(x, t) \geq c_2 V(x, t)$,
3. $|V_x(x, t)g(x, t)|^2 \leq c_3 V^2(x, t)$.

Then

$$\liminf_{t\to\infty} \frac{1}{t} \log |x(t; t_0, x_0)| \geq \frac{2c_2 - c_3}{2p} \quad a.s.$$

for all $x_0 \neq 0$ *in* $\mathbf{R}^d$.

The proof again follows the same method Mao used in his book, but here it works without modifications for our weaker assumptions on the function V. For completeness we, however, give a more worked out proof than given in [9].

Proof Just like in the proof of Theorem 2 we fix some $x_0 \neq 0$ and we write $x(t) = x(t; t_0, x_0)$. Furthermore we define $M(t)$ and $h(s)$ as in the proof of Theorem 2, and by Itô's formula we have that

$$\begin{aligned} &\log V(x(t), t) \qquad (4)\\ &= \log V(x_0, t_0) + \int_{t_0}^{t} \frac{LV(x(s), s)}{V(x(s), s)} \mathrm{d}s + M(t) - \frac{1}{2} \int_{t_0}^{t} \frac{|V_x(x(s), s)g(x(s), s)|^2}{(V(x(s), s))^2} \mathrm{d}s. \end{aligned}$$

By condition 3, we have that $|h(s)|^2 < c_3$, so $h \in \mathscr{M}^2(\mathbf{R}_+, \mathbf{R}^{1\times m})$ and $M(t) = \int_{t_0}^t h(s)\mathrm{d}B(s)$ is a martingale. By Eq. (4) and condition 2

$$\begin{aligned} \log V(x(t), t) &\geq \log V(x_0, t_0) + c_2(t - t_0) - \frac{c_3}{2}(t - t_0) + M(t) \\ &= \log V(x_0, t_0) + \frac{1}{2}(2c_2 - c_3)(t - t_0) + M(t). \end{aligned} \qquad (5)$$

Since $M(t)$ is a martingale with quadratic variation

$$\langle M(t), M(t)\rangle = \int_{t_0}^{t} |h(s)|^2 \mathrm{d}s \leq c_3(t - t_0),$$

we have by the strong law of large numbers, cf. e.g. [9, Thm 1.3.4], that $\lim_{t\to\infty} M(t)/t = 0$ a.s. It therefore follows from (5) that

$$\liminf_{t\to\infty} \frac{1}{t} \log V(x(t), t) \geq \frac{1}{2}(2c_2 - c_3) \quad \text{a.s.}$$

Finally by condition 1 then

$$\liminf_{t\to\infty} \frac{1}{t} \log |x(t; t_0, x_0)| \geq \frac{2c_2 - c_3}{2p}.$$

□

Remark If in the last theorem we have $2c_2 > c_3$, then almost all the sample paths of $t \mapsto |x(t; t_0, x_0)|$ will tend to infinity, and in this case the trivial solution of Eq. (1) is said to be *almost surely exponentially unstable*.

Example Consider the 1-dimensional SDE

$$dX(t) = f(X(t), t)dt + g(X(t), t)dB(t) := \frac{1}{4}X(t)dt + X(t)dB(t) \tag{6}$$

Set $V(x, t) = |x|^{1/2}$, then $V \in C_0^{2,1}$ and, by Eq. (2), the function $LV(x)$ is given by

$$LV(x) = \frac{1}{4}x \cdot (1/2)|x|^{-1/2} + \frac{1}{2}x^2 \cdot (-1/2)(1/2)|x|^{-3/2} = \frac{1}{8}|x|^{1/2} - \frac{1}{8}|x|^{1/2} = 0.$$

Furthermore we see that

$$|V_x(x)g(x, t)|^2 = |(1/2)|x|^{-1/2}x|^2 = \frac{1}{4}(|x|^{1/2})^2 = \frac{1}{4}V(x)^2.$$

Fixing constants $c_1 = 1$, $p = 1/2$, $c_2 = 0$ and $0 < c_3 < 1/4$, we see by Theorem 2 that for any solution $x(t)$ of Eq. (6) the following inequality holds

$$\limsup_{t\to\infty} \frac{1}{t} \log |x(t)| \leq -\frac{c_3 - 2c_2}{2p} = -c_3 < 0 \quad \text{a.s.}$$

In particular the trivial solution of system (6) is almost surely exponentially stable (in fact the solution is stable in probability, see [8, Thm. 5.3]), and the function V we used is not differentiable at 0. Moreover, as shown by Khasminskii [8, p. 154–155], there cannot exists a Lyapunov function for this system that is differentiable at the origin.

3 Conclusions

In his book [9] X. Mao states and proves two theorems, Theorem 4.3.3 and Theorem 4.3.5, where he shows that the existence of a certain auxiliary function, so-called Lyapunov function, implies the *almost sure exponential stability* or, for a different kind of function, the *almost sure exponential instability* respectively of the zero solution of a SDE. Unfortunately, the class of functions $C^{2,1}(\mathbf{R}^d \times [t_0, \infty); \mathbf{R}_+)$

he considers to serve as the foundation for Lyapunov functions is too restrictive as had already been pointed out in the literature [8, p. 154–155]. The adequate class of functions is given by $C_0^{2,1}(\mathbf{R}^d \times [t_0, \infty); \mathbf{R}_+)$ and we formulate and prove Mao's theorems for this wider class of functions. This renders these theorems much more useful for applications.

Acknowledgements This research was supported by the Icelandic Research Fund (Rannís) grant number 152429-051, Lyapunov Methods and Stochastic Stability.

References

1. Lyapunov, A.M.: The general problem of the stability of motion. Int. J. Control **55**(3), 531–534 (1992)
2. Awrejcewicz, J., Kudra, G.: Mathematical modelling and simulation of the bifurcational wobblestone dynamics. Discontinuity, Nonlinearity, and Complexity **3**(2), 123–132 (2014)
3. Awrejcewicza, J., Wasilewskia, G., Kudra, G., Reshmin, S.: An experiment with swinging up a double pendulum using feedback control. J. Comput. Syst. Sci. Int. **51**(2), 176–182 (2012)
4. Awrejcewicz, J., Kudra, G., Wasilewski, G.: Experimental and numerical investigation of chaotic regions in the triple physical pendulum. Nonlinear Dyn. **50**, 755–766 (2007)
5. Awrejcewicz, J., Kudra, G., Wasilewski, G.: Chaotic zones in triple pendulum dynamics observed experimentally and numerically. Appl. Mech. Mater. **9**, 1–17 (2008)
6. Higham, D.J., Mao, X., Yuan, C.: Almost sure and moment exponential stability in the numerical simulation of stochastic differential equations. SIAM. J. Numer. Anal. **45**(2), 592–609 (2007)
7. Kallenberg, O.: Foundations of Modern Probability. Springer Science & Business Media, (2006)
8. Khasminskii, R.: Stochastic stability of differential equations, vol. 66. Springer Science & Business Media (2007)
9. Mao, X.: Stochastic Differential Equations and Applications. Elsevier (2007)
10. Oksendal, B.: Stochastic Differential Equations. Springer (2003)
11. Wu, F., Mao, X., Szpruch, L.: Almost sure exponential stability of numerical solutions for stochastic delay differential equations. Numer. Math. (Heidelb) **115**(4), 681–697 (2010)

Paper II

Computation of the stochastic basin of attraction by rigorous construction of a Lyapunov function.

Bjornsson, H. and Hafstein, S. and Giesl, P. and Scalas, E. and Gudmundsson, S

Discrete and Continuous Dynamical Systems, Series B, 24-8, Issue 2

DISCRETE AND CONTINUOUS
DYNAMICAL SYSTEMS SERIES B
Volume **24**, Number **8**, August **2019**

doi:10.3934/dcdsb.2019080

pp. **4247–4269**

COMPUTATION OF THE STOCHASTIC BASIN OF ATTRACTION BY RIGOROUS CONSTRUCTION OF A LYAPUNOV FUNCTION

Hjörtur Björnsson and Sigurdur Hafstein

Faculty of Physical Sciences
University of Iceland
107 Reykjavik, Iceland

Peter Giesl and Enrico Scalas

Department of Mathematics
University of Sussex
Falmer BN1 9QH, United Kingdom

Skuli Gudmundsson

Svensk Exportkredit
Klarabergsviadukten 61-63
111 64 Stockholm, Sweden

(Communicated by Xiaoying Han)

Abstract. The γ-basin of attraction of the zero solution of a nonlinear stochastic differential equation can be determined through a pair of a local and a non-local Lyapunov function. In this paper, we construct a non-local Lyapunov function by solving a second-order PDE using meshless collocation. We provide a-posteriori error estimates which guarantee that the constructed function is indeed a non-local Lyapunov function. Combining this method with the computation of a local Lyapunov function for the linearisation around an equilibrium of the stochastic differential equation in question, a problem which is much more manageable than computing a Lyapunov function in a large area containing the equilibrium, we provide a rigorous estimate of the stochastic γ-basin of attraction of the equilibrium.

1. Introduction.

In deterministic dynamical systems given by autonomous ordinary differential equations (ODE), the basin of attraction of an asymptotically stable equilibrium is the set of all initial conditions, such that the corresponding solutions converge to the equilibrium as time tends to infinity. When considering a stochastic differential equation (SDE), this notion can be replaced by the γ-basin of attraction, i.e. the set of all initial conditions, such that sample paths will converge to the equilibrium as time tends to infinity with probability at least γ. This concept will be defined in Section 2, Definition 2.2.

2010 *Mathematics Subject Classification.* Primary: 37B25, 65P40, 93E03, 93E15, 93D30; Secondary: 60H35, 65C30.

Key words and phrases. Nonlinear stochastic differential equation, Lyapunov function, Radial basis function, Basin of attraction.

The research for this paper was supported by the Icelandic Research Fund (Rannís) in the project 'Lyapunov Methods and Stochastic Stability' (152429-051), which is gratefully acknowledged.

It turns out that the γ-basin of attraction can be determined using Lyapunov functions. In [8], a combination of a local and a non-local Lyapunov function was used to determine a subset of the γ-basin of attraction. A Lyapunov function $V\colon \mathbb{R}^d \to \mathbb{R}$ for a SDE satisfies $LV(\mathbf{x}) \leq 0$, where L is a second-order differential operator, which arises from the SDE. A local Lyapunov function is only defined in a small neighborhood of the equilibrium and can often be determined by linearisation. A non-local Lyapunov function, however, is defined on a superset $\widetilde{\mathcal{U}} \subset \mathbb{R}^d$ of the γ-basin of attraction apart from a small neighborhood, where the negativity condition is not necessarily satisfied. Local Lyapunov functions will be defined in Section 2, Definition 2.3, and non-local ones in Section 2, Definition 2.4.

In this paper, we present a constructive method to compute a non-local Lyapunov function for a general SDE. In particular, we use meshless collocation to solve a PDE boundary value problem of the form $LV(\mathbf{x}) = \tilde{\nu} < 0$ for all $\mathbf{x} \in \widetilde{\mathcal{U}}$ and with fixed boundary values for $V(\mathbf{x})$ at all $\mathbf{x} \in \partial\widetilde{\mathcal{U}}$. After choosing a kernel, in particular a Radial Basis Function, as well as collocation points in $\widetilde{\mathcal{U}}$ and $\partial\widetilde{\mathcal{U}}$, the approximate solution v to the problem is determined by using a certain ansatz and by computing coefficients by solving a linear equation.

To ensure that the approximation v is itself a valid Lyapunov function, we provide rigorous a-posteriori estimates on $Lv(\mathbf{x})$. This is achieved by evaluating $Lv(\mathbf{x})$ at all $\mathbf{x}$ in a test grid and using Taylor-type estimates for the points in between. These make use of the specific ansatz and corresponding estimates. The method is applied to two examples in one and two dimensions, respectively.

The outline of the paper is as follows: In Section 2 we recall the definition of the γ-basin of attraction and its determination using a pair of a local and a non-local Lyapunov function. In Section 3 we discuss meshless collocation for general PDE boundary value problems and in particular for the PDE related to the SDE under study. Moreover, we present a-posteriori error estimates based on first and second derivatives of Lv. Section 4 applies these results to the construction of a non-local Lyapunov function. Finally, we apply the method to two examples in Section 5. The appendix contains explicit formulas for the ansatz using meshless collocation, as well as tables for the estimates.

Note on notations:

If not specified, we use the Euclidean norm of a vector $\mathbf{x} \in \mathbb{R}^d$, i.e. $\|\mathbf{x}\| := \|\mathbf{x}\|_2$. We denote the closed ϵ-neighborhood with respect to the $\|\cdot\|_1$ norm of a compact set $K \subset \mathbb{R}^d$ by

$$K_{\epsilon,\|\cdot\|_1} = \{\mathbf{x} \in \mathbb{R}^d \,:\, \underset{\|\cdot\|_1}{\operatorname{dist}}(\mathbf{x}, K) \leq \epsilon\},$$

where $\underset{\|\cdot\|_1}{\operatorname{dist}}(\mathbf{x}, K) = \min_{\mathbf{y}\in K} \|\mathbf{x} - \mathbf{y}\|_1$. We sometimes denote the i-th component of a vector $\mathbf{x} - \mathbf{y}$ by $(\mathbf{x} - \mathbf{y})_i$ to shorten formulas.

2. Stochastic basin of attraction and Lyapunov functions.

In this section we introduce the type of SDE that we study as well as the stochastic basin of attraction of the zero (trivial) solution. We also recall the definition of (stochastic) Lyapunov functions; in particular, we will consider an appropriate combination of a local and a non-local Lyapunov function to determine the stochastic basin of attraction.

We study the stability of the trivial solution of the SDE of Itô type

$$d\mathbf{X}(t) \quad = \quad \mathbf{f}(\mathbf{X}(t))\,dt + \mathbf{g}(\mathbf{X}(t))d\mathbf{W}(t), \tag{2.1}$$

where $\mathbf{W}(t)$ is a Q-dimensional Wiener process. The functions $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$ and $\mathbf{g} : \mathbb{R}^d \to \mathbb{R}^{d\times Q}$ are Lipschitz continuous on a neighbourhood of the origin $\mathcal{O}$, i.e. there exists a $K > 0$ such that

$$\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\| + \|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{y})\| \le K\|\mathbf{x} - \mathbf{y}\| \quad \text{for all } \mathbf{x}, \mathbf{y} \in \mathcal{O}.$$

Moreover, we assume that $\mathbf{f}(\mathbf{0}) = \mathbf{0}$ and $\mathbf{g}(\mathbf{0}) = 0$, so that $\mathbf{X}(t) = \mathbf{0}$ is a solution of (2.1) for all $t \ge 0$.

Since we are interested in local stability, i.e. γ-basins of attraction within $\mathcal{O}$, we can extend $\mathbf{f}$ and $\mathbf{g}$ to Lipschitz continuous functions on $\mathbb{R}^d$ and consider *strong solutions* to (2.1) on $[0, \infty)$. This simplifies technical matters considerably, cf. [8, §2].

For the SDE (2.1) the associated *generator* is given by

$$LV(\mathbf{x}) := \nabla V(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}) + \frac{1}{2} \sum_{i,j=1}^{d} \left[\mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^\top\right]_{ij} \frac{\partial^2 V}{\partial x_i \partial x_j}(\mathbf{x}), \tag{2.2}$$

for $V : \mathcal{U} \to \mathbb{R}$ with $\mathcal{U} \subset \mathbb{R}^d$.

Remark 2.1. If the matrix $\mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^\top$ is positive definite for all $\mathbf{x} \in \mathcal{U}$ in a compact set $\mathcal{U} \subset \mathbb{R}^d$, then the second-order linear differential operator L is strictly elliptic in $\mathcal{U}$. In this (non-degenerate) case, results about the existence of classical solutions are available, however, in this paper we will discuss the general case and make no requirement on the positive definiteness of the matrix.

Let us now define the γ-basin of attraction which describes the set of initial conditions so that sample paths converge to the origin with probability at least γ, see [8, Definition 2.4].

Definition 2.2 (γ-basin of attraction (γ-BOA))**.** Consider the system (2.1) and let $0 < \gamma \le 1$. We refer to the set

$$\left\{\mathbf{x} \in \mathbb{R}^d \ : \ \mathbb{P}\left\{\lim_{t\to\infty} \|\mathbf{X}^{\mathbf{x}}(t)\| = 0\right\} \ge \gamma\right\} \qquad (\gamma\text{-BOA})$$

as the *γ-basin of attraction* or short γ-BOA of the origin. Here, $\mathbf{X}^{\mathbf{x}}(t)$ denotes the unique strong solution (stochastic process) of the SDE with initial condition $\mathbf{x}$.

In the following definition [8, Definition 2.5], we introduce a local Lyapunov function in the set $\mathcal{N}$ (see also [8, Theorem 2.7]). A local Lyapunov function U is a positive definite function such that LU is negative definite in a (small) neighbourhood $\mathcal{N}$ of $\mathbf{0}$. This is most conveniently defined using so-called $\mathcal{K}_\infty$ functions; a function $\mu : \mathbb{R}_+ \to \mathbb{R}_+$ is said to be of class $\mathcal{K}_\infty$ if it is continuous, strictly increasing, $\mu(0) = 0$, and $\lim_{x\to\infty} \mu(x) = \infty$.

Definition 2.3 (Local Lyapunov function)**.** Consider the system (2.1). A function $U \in C(\mathcal{N}) \cap C^2(\mathcal{N} \setminus \{\mathbf{0}\})$, where $\mathbf{0} \in \mathcal{N} \subset \mathbb{R}^d$ is a domain, is called a (local) Lyapunov function for the system (2.1), if there are functions $\mu_1, \mu_2, \mu_3 \in \mathcal{K}_\infty$, such that U fulfills the properties:

(i) $\mu_1(\|\mathbf{x}\|) \le U(\mathbf{x}) \le \mu_2(\|\mathbf{x}\|)$ for all $\mathbf{x} \in \mathcal{N}$
(ii) $LU(\mathbf{x}) \le -\mu_3(\|\mathbf{x}\|)$ for all $\mathbf{x} \in \mathcal{N} \setminus \{\mathbf{0}\}$

Let $U_{\max} > 0$ be such that $U^{-1}([0, U_{\max}])$ is a compact subset of $\mathcal{N}$.

Next we introduce a non-local Lyapunov function in the set $\mathcal{U}$ as in [8, Definition 2.9, (2a)]; note that we have replaced 0 by b and 1 by a. A non-local Lyapunov function satisfies $LV < 0$ in a large set $\mathcal{U}$, not including a small neighborhood $\mathcal{B}$ of the equilibrium.

Definition 2.4 (Non-local Lyapunov function)**.** Let $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^d$, $\mathcal{B} \subset \mathcal{A}^\circ$, be simply connected compact neighbourhoods of the origin with C^2 boundaries and set $\mathcal{U} := \mathcal{A} \setminus \mathcal{B}^\circ$. A function $V \in C^2(\mathcal{U})$ for the system (2.1) such that

(1) $b \leq V(\mathbf{x}) \leq a$ for all $\mathbf{x} \in \mathcal{U}$, $V^{-1}(b) = \partial\mathcal{B}$, $V^{-1}(a) = \partial\mathcal{A}$ with $b < a$, and
(2) $LV(\mathbf{x}) < 0$ for all $\mathbf{x} \in \mathcal{U}$,

is called a *non-local Lyapunov function* for the system (2.1). We refer to ∂A as the outer boundary of $\mathcal{U}$ and ∂B as the inner boundary of $\mathcal{U}$.

The following result from [8, Theorem 2.11] shows how a local and a non-local Lyapunov function provide information about the γ-BOA. For an illustration of the various sets, see [8, Figure 1]. The proof uses the non-local Lyapunov function to estimate the probability that solutions starting in $\mathcal{U}$ leave the set through the boundary $\partial\mathcal{B}$, and then the local Lyapunov function estimates the probability that they converge to the origin once they are in $\mathcal{B}$. The combined probability can be bounded by γ.

Theorem 2.5. *Consider the system* (2.1) *and assume there exists a local Lyapunov function* $U : \mathcal{N} \to \mathbb{R}_+$ *as in Definition 2.3 with the constant* $U_{\max} > 0$ *and a non-local Lyapunov function* $V : \mathcal{U} \to \mathbb{R}_+$ *as in Definition 2.4. Let* $0 < \beta < 1$ *and* $b < \lambda < \alpha < a$ *and the set* $\mathcal{B}$ *from Definition 2.4 be such that*

$$U^{-1}(U_{\max}) \subset V^{-1}([b,\lambda]) \quad \textit{and} \quad \partial\mathcal{B} = V^{-1}(b) \subset U^{-1}([0, \beta U_{\max}]).$$

Then the set $V^{-1}([b,\alpha]) \cup \mathcal{B}$ *is a subset of the* γ*-BOA of the origin, where*

$$\gamma := \frac{(a-\alpha)(1-\beta)}{a-b-\beta(a-\lambda)}. \tag{2.3}$$

Note that the bound (2.3) has a different formula than in [8, Theorem 2.11], because here $\partial\mathcal{B} = V^{-1}(b)$ and $\partial\mathcal{A} = V^{-1}(a)$ with b and a not necessarily equal to 0 and 1, respectively. Thus our formula is the formula from [8, Theorem 2.11] with γ replaced by $(\gamma - b)/(a-b)$ and α replaced by $(\alpha - b)/(a-b)$.

In this paper, we focus on a general method to compute non-local Lyapunov functions. Local Lyapunov functions can often be found directly in specific examples: for example, if the noise is small and the origin is an asymptotically stable equilibrium of the corresponding deterministic system with no noise, then the deterministic Lyapunov function can serve as a local Lyapunov function. Another way to construct a local Lyapunov function is similar to the construction of local Lyapunov functions for deterministic systems: by linearising the system around the origin and constructing a Lyapunov function for the linearised system, which is a local Lyapunov function for the nonlinear system, see [1].

For the examples in this paper, we are able to construct local Lyapunov functions with one of these two approaches. For a more general discussion on the construction of Lyapunov functions for linear systems see also [9].

3. Meshless collocation.

In this section we will recall meshless collocation and its use to approximate solutions of boundary value problems for general linear PDEs of the form

$$\begin{cases} LV(\mathbf{x}) & = & r(\mathbf{x}) \text{ for } \mathbf{x} \in \Omega, \\ V(\mathbf{x}) & = & c(\mathbf{x}) \text{ for } \mathbf{x} \in \partial\Omega, \end{cases} \tag{3.1}$$

where L is a linear differential operator and Ω is a bounded domain in $\mathbb{R}^d$ with sufficiently smooth boundary. Meshless collocation seeks to find the solution v of an interpolation problem, which minimises the norm in a Reproducing Kernel Hilbert Space (RKHS), in our case a Sobolev space. The interpolation problem will ensure that v satisfies the PDE and the boundary values (3.1) at given collocation points.

If the PDE boundary value problem has a solution V, then v approximates V and we have error estimates of $\|V(\mathbf{x}) - v(\mathbf{x})\|_{L_\infty(\partial\Omega)}$ as well as $\|LV(\mathbf{x}) - Lv(\mathbf{x})\|_{L_\infty(\Omega)}$. The error estimates involve the fill distance of the collocation points, measuring how dense they are in Ω and $\partial\Omega$, respectively. Unfortunately, these estimates also involve unknown quantities, such as the norm of V. Thus, these error estimates ensure that by adding more and more collocation points the error converges to zero, but they do not provide explicit, computable bounds on the error.

We can, however, compute explicit a-posteriori bounds on the errors $\|V(\mathbf{x}) - v(\mathbf{x})\|_{L_\infty(\partial\Omega)}$ as well as $\|LV(\mathbf{x}) - Lv(\mathbf{x})\|_{L_\infty(\Omega)}$ by first computing $|LV(\mathbf{x}) - Lv(\mathbf{x})|$ for a finite, but large set of points $Y \subset \Omega$. Taylor's theorem and estimates on the first and second derivatives by using the explicit form of v provide us with explicit bounds on these errors as shown in Section 3.2.

3.1. Meshless collocation: PDE boundary value problems.

Meshless collocation, in particular by Radial Basis Functions, is a powerful method to solve linear PDEs [11, 2, 12]. For a general introduction to meshless collocation and RKHS, see [14]. Meshless collocation has been applied to the computation of Lyapunov functions in deterministic systems [4, 7]. For an overview of this and other methods to compute Lyapunov functions, see the review [5].

In this section, we will outline the method, apply it to our particular case, and recall known results, in particular error estimates from [4].

We consider a general linear operator L of order m given by

$$LV(\mathbf{x}) = \sum_{|\boldsymbol{\alpha}|\le m} c_{\boldsymbol{\alpha}}(\mathbf{x})\partial_{\boldsymbol{\alpha}} V(\mathbf{x}). \tag{3.2}$$

In our case, $m = 2$ and the operator is given by

$$Lv(\mathbf{x}) = \frac{1}{2}\sum_{i,j=1}^{d} m_{ij}(\mathbf{x})\frac{\partial^2}{\partial x_i \partial x_j} v(\mathbf{x}) + \sum_{i=1}^{d} f_i(\mathbf{x})\frac{\partial}{\partial x_i} v(\mathbf{x}), \tag{3.3}$$

where $(m_{ij}(\mathbf{x}))_{i,j=1,\ldots,d} = \mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^\top$, i.e. $m_{ij}(\mathbf{x}) = \sum_{q=1}^{Q} g_{iq}(\mathbf{x}) g_{jq}(\mathbf{x})$. We denote the q-th column of $\mathbf{g}$ by g^q.

Hence, our operator is of the form (3.2) with $c_{\mathbf{e}_i}(\mathbf{x}) = f_i(\mathbf{x})$ and $c_{\mathbf{e}_i+\mathbf{e}_j}(\mathbf{x}) = \frac{1}{2} m_{ij}(\mathbf{x})$. A singular point of L is a point $\mathbf{x}$ with $c_{\boldsymbol{\alpha}}(\mathbf{x}) = 0$ for all $|\boldsymbol{\alpha}| \le 2$, see [4, Definition 3.2].

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with smooth boundary $\Gamma := \partial\Omega$. Our goal is to (approximately) solve the boundary value problem with a PDE given by:

$$\begin{cases} Lv(\mathbf{x}) &= r(\mathbf{x}) \text{ for } \mathbf{x} \in \Omega, \\ v(\mathbf{x}) &= c(\mathbf{x}) \text{ for } \mathbf{x} \in \Gamma. \end{cases} \tag{3.4}$$

Our approximation will be a function in a RKHS, which is a Hilbert space H of functions $\Omega \to \mathbb{R}$ with inner product $\langle \cdot, \cdot \rangle_H$, and a kernel $\Phi\colon \Omega \times \Omega \to \mathbb{R}$ such that

1. $\Phi(\cdot, \mathbf{x}) \in H$ for all $\mathbf{x} \in \Omega$,
2. $g(\mathbf{x}) = \langle g, \Phi(\cdot, \mathbf{x}) \rangle_H$ for all $\mathbf{x} \in \Omega$ and $g \in H$.

In our case, we choose the radially symmetric kernel $\Phi(\mathbf{x},\mathbf{y}) = \psi(\|\mathbf{x}-\mathbf{y}\|)$, where $\psi = \psi_{\ell,k}$ is given by a Wendland function [13], see also Table 1. Setting $\ell = \lfloor \frac{d}{2} \rfloor + k + 1$, the parameter $k \in \mathbb{N}$ is a smoothness index and the function $\Phi(\mathbf{x},\mathbf{y})$ is a C^{2k} function in $\mathbf{x}$ for fixed $\mathbf{y}$, and the RKHS with this kernel is norm-equivalent to the Sobolev space W_2^τ with $\tau = k + \frac{d+1}{2}$.

Given sets of pairwise distinct points $X_1 = \{\mathbf{x}_1,\ldots,\mathbf{x}_N\} \subset \Omega \subset \mathbb{R}^d$, none of which is a singular point of L, and pairwise distinct points $X_2 = \{\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_M\} \subset \Gamma = \partial\Omega$, we seek to find the (unique) solution v to the interpolation problem

$$\begin{aligned} Lv(\mathbf{x}_i) &= r(\mathbf{x}_i) \text{ for all } i = 1,\ldots,N, \\ v(\boldsymbol{\xi}_i) &= c(\boldsymbol{\xi}_i) \text{ for all } i = 1,\ldots,M, \end{aligned}$$

which minimises the norm of the RKHS. It turns out that the solution is given by

$$\begin{aligned} v(\mathbf{x}) &= \sum_{k=1}^{N} \alpha_k (\delta_{\mathbf{x}_k} \circ L)^{\mathbf{y}} \psi(\|\mathbf{x}-\mathbf{y}\|) \\ &\quad + \sum_{k=1}^{M} \alpha_{N+k} (\delta_{\boldsymbol{\xi}_k} \circ L^0)^{\mathbf{y}} \psi(\|\mathbf{x}-\mathbf{y}\|), \end{aligned} \tag{3.5}$$

where $L^0 = \mathrm{id}$, $\delta_{\mathbf{y}} v(\mathbf{x}) = v(\mathbf{y})$, the superscript $\mathbf{y}$ denotes that the operator is applied with respect to the variable $\mathbf{y}$, and the coefficients α_k are computed by solving the linear system $A\alpha = \beta$, where $\beta_k = r(\mathbf{x}_k)$ for $k = 1,\ldots,N$ and $\beta_{N+k} = c(\boldsymbol{\xi}_k)$ for $k = 1,\ldots,M$. $A = (a_{kl})$ is a symmetric $(N+M)\times(N+M)$ matrix given by $A = \begin{pmatrix} B & C \\ C^\top & D \end{pmatrix}$ with $B \in \mathbb{R}^{N\times N}$, $C \in \mathbb{R}^{N\times M}$, $D \in \mathbb{R}^{M\times M}$, where

$$\begin{aligned} &\text{for } k,l = 1,\ldots,N: \\ &\quad b_{kl} = (\delta_{\mathbf{x}_k} \circ L)^{\mathbf{x}} (\delta_{\mathbf{x}_l} \circ L)^{\mathbf{y}} \psi(\|\mathbf{x}-\mathbf{y}\|), \\ &\text{for } k = 1,\ldots,N, l = 1,\ldots,M: \\ &\quad c_{kl} = (\delta_{\mathbf{x}_k} \circ L)^{\mathbf{x}} (\delta_{\boldsymbol{\xi}_l} \circ L^0)^{\mathbf{y}} \psi(\|\mathbf{x}-\mathbf{y}\|) = (\delta_{\mathbf{x}_k} \circ L)^{\mathbf{x}} \psi(\|\mathbf{x}-\boldsymbol{\xi}_l\|), \\ &\text{for } k,l = 1,\ldots,M: \\ &\quad d_{kl} = (\delta_{\boldsymbol{\xi}_k} \circ L^0)^{\mathbf{x}} (\delta_{\boldsymbol{\xi}_l} \circ L^0)^{\mathbf{y}} \psi(\|\mathbf{x}-\mathbf{y}\|) = \psi(\|\boldsymbol{\xi}_k - \boldsymbol{\xi}_l\|). \end{aligned}$$

Explicit formulas for v and Lv are given in the Appendix A.

If the PDE has a solution V, then error estimates imply that the function v is an approximation to V as stated in Theorem 3.1 below. Note that the mesh norms measure how dense the points in X_1 and X_2 are in the domain and boundary, respectively. The following is [4, Corollary 3.12] adapted to our linear operator.

Theorem 3.1. *Let $k > 3/2$, if d is odd, and $k > 2$, if d is even. Let $f_i, m_{ij} \in W_\infty^{k-1+\lfloor \frac{d+1}{2} \rfloor}(\Omega)$ and let the solution V of (3.4) satisfy $V \in W^{k+(d+1)/2}(\Omega)$. Then the approximation v as above, for sufficiently small mesh norms, satisfies*

$$\|LV - Lv\|_{L_\infty(\Omega)} \le C h_{X_1,\Omega}^{k-3/2} \|V\|_{W_2^{k+(d+1)/2}(\Omega)}, \tag{3.6}$$

$$\|V - v\|_{L_\infty(\partial\Omega)} \le C h_{X_2,\partial\Omega}^{k+1/2} \|V\|_{W_2^{k+(d+1)/2}(\Omega)}, \tag{3.7}$$

where $h_{X_1,\Omega} = \sup_{\mathbf{x}\in\Omega} \min_{\mathbf{x}_j \in X_1} \|\mathbf{x}-\mathbf{x}_j\|$ and the constant $h_{X_2,\partial\Omega}$ is the mesh norm for the boundary part, for the precise definition see [4].

3.2. **A-posteriori error estimates.** Note that, unless L is non-degenerate, we have no results on the existence of classical solutions and thus we cannot use Theorem 3.1. Even in that case, the error estimates in Theorem 3.1 contain quantities that are not known explicitly, such as $\|V\|_{W_2^{k+(d+1)/2}(\Omega)}$.

Hence, in this section we derive estimates that only contain explicitly computable constants. They do not require us to prove the existence of a solution, but are a verification that the computed function satisfies an inequality at all points. The main idea is to evaluate the function at many points on a test grid and then use a Taylor-type argument in between. As we have an explicit formula for the approximation, we can derive explicit bounds on the derivatives. As these are multiplied by the mesh norm of the test grid, which can be made arbitrarily small, we can make the estimate as accurate as necessary.

Let us first present the Taylor-type estimates for a general function u, which later will be either the approximation v or Lv. These theorems, as well as a more detailed discussion of a suitable choice of the test grid are taken from [10], see also [6].

As test grids, we will use the following:

Definition 3.2. Define the following grids in $\mathbb{R}^d$ with $h > 0$:

- $S_h = h\mathbb{Z}^d$
- $C_h = S_h \cup \left(\frac{h}{2}\mathbf{1} + S_h\right)$, where $\mathbf{1} = (1, \dots, 1) \in \mathbb{R}^d$

The following theorem is based on the mean-value theorem and uses the specific structure of the grid points S_h.

Theorem 3.3 (First derivative)**.** *Let $u \in C^1(\mathbb{R}^d, \mathbb{R})$ and let $K \subset \mathbb{R}^d$ be compact. Fix $h > 0$ and let $Y := C_h \cap K_{h\,d/4, \|\cdot\|_1}$.*

Define

$$e_h = \frac{d}{4} \max_{\mathbf{z} \in K_{h\,d/4,\|\cdot\|_1}} \max_{l \in \{1,\dots,d\}} \left| \frac{\partial u}{\partial x_l}(\mathbf{z}) \right| h.$$

Then we have for all $\mathbf{x} \in K$ that

$$\min_{\mathbf{y} \in Y} u(\mathbf{y}) - e_h \le u(\mathbf{x}) \le \max_{\mathbf{y} \in Y} u(\mathbf{y}) + e_h.$$

Proof. Let $\mathbf{x} \in K$. Then there is a $\mathbf{y} \in C_h$ with $\|\mathbf{x} - \mathbf{y}\|_1 \le \frac{d}{4}h$, see [10, Theorem 5.5], and thus $\mathbf{y} \in Y$. The mean value theorem shows that there is a $\theta \in [0, 1]$ such that

$$\begin{aligned} |u(\mathbf{x}) - u(\mathbf{y})| &= |\nabla u(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \cdot (\mathbf{x} - \mathbf{y})| \\ &\le \|\nabla u(\theta\mathbf{x} + (1-\theta)\mathbf{y})\|_\infty \|\mathbf{x} - \mathbf{y}\|_1 \\ &\le \max_{l \in \{1,\dots,d\}} \left| \frac{\partial u}{\partial x_l}(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \right| \frac{d}{4}h\,. \end{aligned}$$

Note that $\theta\mathbf{x} + (1-\theta)\mathbf{y} \in K_{h\,d/4,\|\cdot\|_1}$, since

$$\|\theta\mathbf{x} + (1-\theta)\mathbf{y} - \mathbf{x}\|_1 = (1-\theta)\|\mathbf{y} - \mathbf{x}\|_1 \le \frac{d}{4}h.$$

This shows the statement. □

The next theorem relies on a triangulation of the phase space with vertices in S_h, C_h, respectively. Using Taylor's theorem in each simplex, we can derive the estimates below. Note that, as discussed in [10], depending on odd or even dimension, we use either S_h or C_h to obtain an estimate with as few points as possible.

Theorem 3.4 (Second derivative)**.** *Let $u \in C^2(\mathbb{R}^d, \mathbb{R})$ and let $K \subset \mathbb{R}^d$ be compact. Fix $h > 0$. If $d > 1$ define*

$$e_h = \frac{d^2}{4} \max_{\mathbf{z} \in K_{d\,h, \|\cdot\|_1}} \max_{l,p \in \{1,\dots,d\}} \left| \frac{\partial^2 u}{\partial x_p \partial x_l}(\mathbf{z}) \right| h^2.$$

- *If d is even, then let $Y := S_h \cap K_{d\,h, \|\cdot\|_1}$.*
- *If $d \geq 3$ is odd, then let $Y := C_h \cap K_{(d-1)\,h, \|\cdot\|_1}$.*

In the case $d = 1$ let $Y := C_h \cap K_{h/2, \|\cdot\|_1}$ and define

$$e_h = \frac{1}{4} \max_{\mathbf{z} \in K_{h/2, \|\cdot\|_1}} |u''(\mathbf{z})| \, h^2.$$

In all cases we then have for all $\mathbf{x} \in K$

$$\min_{\mathbf{y} \in Y} u(\mathbf{y}) - e_h \quad \leq \quad u(\mathbf{x}) \quad \leq \quad \max_{\mathbf{y} \in Y} u(\mathbf{y}) + e_h.$$

Proof. We consider the case where d is even. Let $\mathbf{x} \in K$. Then there is a simplex S with vertices $\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_d\} \subset S_h$, such that $\mathbf{x} = \sum_{i=0}^d \lambda_i \mathbf{x}_i \in S$, where $\sum_{i=0}^d \lambda_i = 1$ and $0 \leq \lambda_i \leq 1$. Since $\max_{\mathbf{y},\mathbf{z} \in S} \|\mathbf{y} - \mathbf{z}\|_1 = dh$, we have $S \subset K_{d\,h, \|\cdot\|_1}$ and thus $\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_d\} \subset Y$.

Now we use the following result from [10, Proposition 5.2]: Denote by $h^* := \max_{j=0,\dots,d} \|\mathbf{x}_0 - \mathbf{x}_j\|_1$ the maximal distance from any vertex to the fixed vertex $\mathbf{x}_0$. For $w \in C^2(\mathbb{R}^d, \mathbb{R})$ we have for all $0 \leq \lambda_i \leq 1$ with $\sum_{i=0}^d \lambda_i = 1$ that

$$\left| w\left(\sum_{i=0}^d \lambda_i \mathbf{x}_i \right) - \sum_{i=0}^d \lambda_i w(\mathbf{x}_i) \right| \quad \leq \quad \max_{\mathbf{z} \in S} \max_{l,p \in \{1,\dots,d\}} \left| \frac{\partial^2 w(\mathbf{z})}{\partial x_p \partial x_l} \right| (h^*)^2. \tag{3.8}$$

In our case, we can choose the vertex $\mathbf{x}_0$ such that $h^* = \frac{d}{2} h$. As $\mathbf{x} \in S$ there are $0 \leq \lambda_i \leq 1$ with $\sum_{i=0}^d \lambda_i = 1$ such that $\mathbf{x} = \sum_{i=0}^d \lambda_i \mathbf{x}_i$. Hence, by (3.8)

$$\left| u\left(\mathbf{x}\right) - \sum_{i=0}^d \lambda_i u(\mathbf{x}_i) \right| \quad \leq \quad \max_{\mathbf{z} \in K_{d\,h, \|\cdot\|_1}} \max_{l,p \in \{1,\dots,d\}} \left| \frac{\partial^2 u(\mathbf{z})}{\partial x_p \partial x_l} \right| \frac{d^2}{4} h^2$$

and then

$$u\left(\mathbf{x}\right) \leq \max_{\mathbf{y} \in Y} u(\mathbf{y}) \underbrace{\sum_{i=0}^d \lambda_i}_{=1} + \max_{\mathbf{z} \in K_{d\,h, \|\cdot\|_1}} \max_{l,p \in \{1,\dots,d\}} \left| \frac{\partial^2 u(\mathbf{z})}{\partial x_p \partial x_l} \right| \frac{d^2}{4} h^2$$

and similarly for the other inequality.

The result for odd dimensions follows in a similar way, noting that we choose a simplex S with vertices $\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_d\} \subset C_h$. Since $\max_{\mathbf{y},\mathbf{z} \in S} \|\mathbf{y} - \mathbf{z}\|_1 = (d-1)h$ for $d \geq 2$ and $\frac{h}{2}$ for $d = 1$, and for a simplex with vertices in C_h we can choose the vertex $\mathbf{x}_0$ such that $h^* := \max_{j=0,\dots,d} \|\mathbf{x}_0 - \mathbf{x}_j\|_1 = \frac{d}{2} h$, see [10, Theorem 5.8], the result follows. □

The following theorem provides us with explicit bounds on the first and second derivatives of both v and Lv, as required in Theorems 3.3 and 3.4 for $u = v$ and $u = Lv$, respectively. Note that they involve quantities depending on $\mathbf{f}$ and $\mathbf{g}$ as well as their first and second derivatives, and the (computed) coefficients α_i, $i = 1, \dots, N + M$. Moreover, the bounds $\psi_{i,k}$ as defined below are calculated for specific Wendland functions ψ_0 in the appendix. Note that the requirement on ψ_i is satisfied for Wendland functions with smoothness index $k \geq 6$.

Theorem 3.5. *Let $v \in C^4(\mathbb{R}^d, \mathbb{R})$ be given by (3.5) with kernel $\psi(r) =: \psi_0(r) \in C^6$. Let $C \subset \mathbb{R}^d$ be a compact set. Denote*

- $\psi_i(r) = \frac{1}{r}\frac{d}{dr}\psi_{i-1}(r)$ *for* $r > 0$ *and* $i = 1, \ldots, 6$ *and assume that* $\psi_i(r)$ *can be continuously extended to* $r = 0$,
- $\psi_{i,k} = \sup_{r\in[0,\infty)} |\psi_i(r)| r^k < \infty$ *for* $i, k \in \mathbb{N}_0$,
- $F = \max_{\mathbf{x}\in C} \|\mathbf{f}(\mathbf{x})\|$,
  $F_1 = \max_{\mathbf{x}\in C} \max_{l\in\{1,\ldots,d\}} \left\| \frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_l} \right\|$, *and*
  $F_2 = \max_{\mathbf{x}\in C} \max_{l,p\in\{1,\ldots,d\}} \left\| \frac{\partial^2 \mathbf{f}(\mathbf{x})}{\partial x_p \partial x_l} \right\|$.
- $$\begin{aligned} G &= \frac{1}{2}\sum_{q=1}^{Q} \max_{\mathbf{x}\in C} \|g^q(\mathbf{x})\|^2, \\ G_1 &= \sum_{q=1}^{Q} \max_{\mathbf{x}\in C} \max_{l\in\{1,\ldots,d\}} \|g^q(\mathbf{x})\| \left\| \frac{\partial g^q(\mathbf{x})}{\partial x_l} \right\|, \text{ and} \\ G_2 &= \sum_{q=1}^{Q} \max_{\substack{\mathbf{x}\in C \\ l,p\in\{1,\ldots,d\}}} \left[\left\| \frac{\partial^2 g^q(\mathbf{x})}{\partial x_p \partial x_l} \right\| \|g^q(\mathbf{x})\| + \left\| \frac{\partial g^q(\mathbf{x})}{\partial x_p} \right\| \left\| \frac{\partial g^q(\mathbf{x})}{\partial x_l} \right\| \right], \end{aligned}$$
  where $g^q(\mathbf{x}) \in \mathbb{R}^d$ *denotes the vector* $(g_{iq}(\mathbf{x}))_{i=1,\ldots,d}$ *for all* $q = 1, \ldots, Q$.
- $\boldsymbol{\alpha}_1 = \sum_{k=1}^{N} |\alpha_k|$ *and* $\boldsymbol{\alpha}_2 = \sum_{k=N+1}^{N+M} |\alpha_k|$.

Then we have the following bounds for all $\mathbf{x} \in C$ *and all* $l, p \in \{1, \ldots, d\}$:

$$\begin{aligned} \left| \frac{\partial v}{\partial x_l}(\mathbf{x}) \right| &\leq \boldsymbol{\alpha}_1\{G[\psi_{3,3} + 3\psi_{2,1}] + F[\psi_{2,2} + \psi_{1,0}]\} + \boldsymbol{\alpha}_2\psi_{1,1}, \\ \left| \frac{\partial^2 v}{\partial x_p \partial x_l}(\mathbf{x}) \right| &\leq \boldsymbol{\alpha}_1\{G[\psi_{4,4} + 6\psi_{3,2} + 3\psi_{2,0}] + F[\psi_{3,3} + 3\psi_{2,1}]\} \\ &\quad + \boldsymbol{\alpha}_2[\psi_{2,2} + \psi_{1,0}], \end{aligned}$$

$$\begin{aligned} \left| \frac{\partial Lv}{\partial x_l}(\mathbf{x}) \right| \leq \boldsymbol{\alpha}_1 \Big\{ & G^2[\psi_{5,5} + 10\psi_{4,3} + 15\psi_{3,1}] \\ & + (2F + G_1)G[\psi_{4,4} + 6\psi_{3,2} + 3\psi_{2,0}] \\ & + (F_1 G + F G_1 + F^2)[\psi_{3,3} + 3\psi_{2,1}] \\ & + F F_1[\psi_{2,2} + \psi_{1,0}] \Big\} \\ + \boldsymbol{\alpha}_2 \Big\{ & G[\psi_{3,3} + 3\psi_{2,1}] + (F + G_1)[\psi_{2,2} + \psi_{1,0}] + F_1\psi_{1,1} \Big\}, \text{ and} \end{aligned}$$

$$\begin{aligned} \left| \frac{\partial^2 Lv}{\partial x_p \partial x_l}(\mathbf{x}) \right| \leq \boldsymbol{\alpha}_1 \Big\{ & G^2[\psi_{6,6} + 15\psi_{5,4} + 45\psi_{4,2} + 15\psi_{3,0}] \\ & + 2(F + G_1)G[\psi_{5,5} + 10\psi_{4,3} + 15\psi_{3,1}] \\ & + (F^2 + G G_2 + 2F_1 G + 2F G_1)[\psi_{4,4} + 6\psi_{3,2} + 3\psi_{2,0}] \\ & + (2F F_1 + F_2 G + F G_2)[\psi_{3,3} + 3\psi_{2,1}] \\ & + F F_2[\psi_{2,2} + \psi_{1,0}] \Big\} \end{aligned}$$

$$+\boldsymbol{\alpha}_2\Big\{G[\psi_{4,4}+6\psi_{3,2}+3\psi_{2,0}] \\ +(F+2G_1)[\psi_{3,3}+3\psi_{2,1}]+(2F_1+G_2)[\psi_{2,2}+\psi_{1,0}]\Big\}.$$

Proof. The proof follows directly by differentiation and estimating terms of similar type, where we use the explicit formulas derived in the appendix. Using formula (A.1) for v we get

$$\begin{aligned}
&\frac{\partial v}{\partial x_j}(\mathbf{x})\\
&= \sum_{k=1}^{N}\alpha_k\Big\{-\psi_2(\|\mathbf{x}-\mathbf{x}_k\|)(\mathbf{x}-\mathbf{x}_k)_l\langle\mathbf{x}-\mathbf{x}_k,\mathbf{f}(\mathbf{x}_k)\rangle\\
&\qquad -\psi_1(\|\mathbf{x}-\mathbf{x}_k\|)f_l(\mathbf{x}_k)\\
&\qquad +\frac{1}{2}\sum_{i,j=1}^{d}m_{ij}(\mathbf{x}_k)\Big[\psi_3(\|\mathbf{x}-\mathbf{x}_k\|)(\mathbf{x}-\mathbf{x}_k)_l(\mathbf{x}-\mathbf{x}_k)_i(\mathbf{x}-\mathbf{x}_k)_j\\
&\qquad +\psi_2(\|\mathbf{x}-\mathbf{x}_k\|)[\delta_{il}(\mathbf{x}-\mathbf{x}_k)_j+(\mathbf{x}-\mathbf{x}_k)_i\delta_{lj}+(\mathbf{x}-\mathbf{x}_k)_l\delta_{ij}]\Big]\Big\}\\
&\quad +\sum_{k=1}^{M}\alpha_{N+k}\psi_1(\|\mathbf{x}-\boldsymbol{\xi}_k\|)(\mathbf{x}-\boldsymbol{\xi}_k)_l\\
&\leq \sum_{k=1}^{N}|\alpha_k|\Big\{|\psi_2(\|\mathbf{x}-\mathbf{x}_k\|)|\ \|\mathbf{x}-\mathbf{x}_k\|^2F+|\psi_1(\|\mathbf{x}-\mathbf{x}_k\|)|F\\
&\qquad +\frac{1}{2}\sum_{i,j=1}^{d}\sum_{q=1}^{Q}g_{iq}(\mathbf{x}_k)g_{jq}(\mathbf{x}_k)\Big[|\psi_3(\|\mathbf{x}-\mathbf{x}_k\|)|\|\mathbf{x}-\mathbf{x}_k\|^3\\
&\qquad +3|\psi_2(\|\mathbf{x}-\mathbf{x}_k\|)|\|\mathbf{x}-\mathbf{x}_k\|\Big]\Big\}\\
&\quad +\sum_{k=1}^{M}|\alpha_{N+k}|\ |\psi_1(\|\mathbf{x}-\boldsymbol{\xi}_k\|)|\ \|\mathbf{x}-\boldsymbol{\xi}_k\|.
\end{aligned}$$

This shows the first estimate, using the definitions of $\boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_2$, and $\psi_{l,k}$. The other estimates are proved in a similar way. □

4. **Non-local Lyapunov function.** In this section we will present a method to use meshless collocation, as discussed in the previous section, to compute a non-local Lyapunov function and combine it with a given, local Lyapunov function.

We seek to find a non-local Lyapunov function v satisfying $Lv(\mathbf{x})<0$, see Definition 2.4. This is done by finding an approximate solution of the PDE $LV(\mathbf{x})=\tilde{\nu}$ with $\tilde{\nu}<0$ in $\widetilde{\mathcal{U}}$ by meshless collocation and using the a-posteriori estimates for Lv to show that v satisfies $Lv(\mathbf{x})\leq\nu<0$. Note, however, that the boundary of $\widetilde{\mathcal{U}}$ is only approximately given by the level sets with level 0 and 1 of v, apart from the case $d=1$. Hence, we compute the minimum of v at the outer boundary of $\widetilde{\mathcal{U}}$ and the maximum of v at the inner boundary of $\widetilde{\mathcal{U}}$, using the a-posteriori estimates for

v. Then we can define $\mathcal{U} = \mathcal{A}\setminus\mathcal{B}^\circ$ via $\mathcal{A}$ and $\mathcal{B}$ through the level sets of v with levels a and b, respectively, and thus show that v satisfies the conditions in Definition 2.4. Theorem 2.5 applied to v then gives us a rigorous result for the stochastic basin of attraction of the equilibrium at the origin.

Let v be the approximate solution of the following boundary-value problem:

$$LV(\mathbf{x}) \quad = \quad \tilde{\nu} \quad \text{for all} \quad \mathbf{x} \in \widetilde{\mathcal{U}}^\circ, \tag{4.1}$$

$$V(\mathbf{x}) \quad = \quad \begin{cases} 0 & \text{for all} \quad \mathbf{x} \in \partial\widetilde{\mathcal{B}}, \\ 1 & \text{for all} \quad \mathbf{x} \in \partial\widetilde{\mathcal{A}}, \end{cases} \tag{4.2}$$

where L is given by (2.2), $\tilde{\nu} < 0$ and $\widetilde{\mathcal{U}} = \widetilde{\mathcal{A}} \setminus \widetilde{\mathcal{B}}^\circ$, where $\widetilde{\mathcal{B}} \subset \widetilde{\mathcal{A}}^\circ$ and $\widetilde{\mathcal{A}}$ and $\widetilde{\mathcal{B}}$ are both simply connected compact neighborhoods of the origin with C^2 boundaries.

We use Theorem 3.3 or Theorem 3.4 with the set $K = \partial\widetilde{\mathcal{B}}$ and fixed $h > 0$ for the function v. We set

$$m := \max_{\mathbf{x}\in Y} v(\mathbf{x}) + e_h,$$

where e_h and Y are defined in Theorem 3.3 or Theorem 3.4, respectively.

We use Theorem 3.3 or Theorem 3.4 with the set $K = \partial\widetilde{\mathcal{A}}$ and fixed $h > 0$ for the function v. We set

$$M := \min_{\mathbf{x}\in Y} v(\mathbf{x}) - e_h,$$

where e_h and Y are defined in Theorem 3.3 or Theorem 3.4, respectively.

Lemma 4.1. *In the situation described above, assume that $v \in C^2$ and $m < M$, and choose $m < b < a < M$. Define $\mathcal{A} = v^{-1}((-\infty, a])$, $\mathcal{B} = v^{-1}((-\infty, b])$ and $\mathcal{U} = \mathcal{A} \setminus \mathcal{B}^\circ$. Assume that $\mathcal{A}$ and $\mathcal{B}$ are simply connected compact neighborhoods of the origin, and assume that $\mathbb{R}^d\setminus\mathcal{B}$ is connected. Then $\mathcal{A}$ and $\mathcal{B}$ have C^2 boundaries, $\mathcal{B} \subset \mathcal{A}^\circ$, and $\mathcal{U} \subset \widetilde{\mathcal{U}}$.*

Proof. The sets $\mathcal{A}$ and $\mathcal{B}$ have C^2 boundaries since $v \in C^2$. $\mathcal{B} \subset \mathcal{A}^\circ$ follows from $b < a$.

We first show now that $\mathcal{A} \subset \widetilde{\mathcal{A}}$. Assuming the opposite, there is a point $\mathbf{x}^* \in \mathcal{A}\setminus\widetilde{\mathcal{A}}$ and, since $\mathcal{A}$ is a connected neighborhood of the origin, there is a continuous path from $\mathbf{x}^*$ to the origin within $\mathcal{A}$, which has to intersect with $\partial\widetilde{\mathcal{A}}$ as the origin is in $\widetilde{\mathcal{A}}$. Hence, there is a point $\mathbf{x} \in \mathcal{A} \cap \partial\widetilde{\mathcal{A}}$. This means that $v(\mathbf{x}) \leq a$ and, because of Theorem 3.3 or 3.4 and the arguments above, that $v(\mathbf{x}) \geq \min_{\mathbf{y}\in Y} v(\mathbf{y}) - e_h = M > a$, which is a contradiction.

Next we show that $\widetilde{\mathcal{B}} \subset \mathcal{B}$. Since both $\widetilde{\mathcal{B}}$ and $\mathcal{B}$ are compact, there is a point $\tilde{\mathbf{x}} \in \mathbb{R}^d$ with $\tilde{\mathbf{x}} \notin \widetilde{\mathcal{B}}$ and $\tilde{\mathbf{x}} \notin \mathcal{B}$. Now assume the opposite to the statement $\widetilde{\mathcal{B}} \subset \mathcal{B}$, namely that there is a point $\mathbf{x}^* \in \widetilde{\mathcal{B}}\setminus\mathcal{B}$ and, since $\mathbb{R}^d\setminus\mathcal{B}$ is a connected neighborhood of $\tilde{\mathbf{x}}$, there is a continuous path from $\mathbf{x}^*$ to $\tilde{\mathbf{x}}$ within $\mathbb{R}^d \setminus \mathcal{B}$, which has to intersect with $\partial\widetilde{\mathcal{B}}$ as $\tilde{\mathbf{x}}$ is in $\mathbb{R}^d \setminus \widetilde{\mathcal{B}}$. Hence, there is a point $\mathbf{x} \in (\mathbb{R}^d \setminus \mathcal{B}) \cap \partial\widetilde{\mathcal{B}}$. This means that $v(\mathbf{x}) > b$ and, because of Theorem 3.3 or 3.4 and the arguments above, that $v(\mathbf{x}) \leq \max_{\mathbf{y}\in Y} v(\mathbf{y}) + e_h = m < b$, which is a contradiction. □

Now we use Theorem 3.3 or 3.4, respectively, to establish that v is a non-local Lyapunov function. To estimate $C_{\mathcal{U}}$ we use Theorem 3.5 with $C = \mathcal{U}_{h\,d/4,\|\cdot\|_1}$. Together with a local Lyapunov function, we can then use Theorem 2.5 to determine a γ-basin of attraction.

Theorem 4.2 (First derivative)**.** *Let $v \in C^3$ be a function given by meshless collocation as described above.*

Let $\mathcal{A} = v^{-1}((-\infty, a])$ and $\mathcal{B} = v^{-1}((-\infty, b])$ and assume that $\mathcal{B} \subset \mathcal{A}^\circ$ and that $\mathcal{A}$ and $\mathcal{B}$ are simply connected compact neighbourhoods of the origin with C^2 boundaries. Set $\mathcal{U} := \mathcal{A} \setminus \mathcal{B}^\circ$.

Fix $h > 0$ and define $Y_\mathcal{U} := C_h \cap \mathcal{U}_{h\,d/4\|\cdot\|_1}$,

$$C_\mathcal{U} := \max_{\mathbf{z}\in\mathcal{U}_{h\,d/4,\|\cdot\|_1}} \max_{l\in\{1,\ldots,d\}} \left|\frac{\partial Lv}{\partial x_l}(\mathbf{z})\right|$$

and

$$\nu \quad := \quad \max_{\mathbf{y}\in Y_\mathcal{U}} Lv(\mathbf{y}) + C_\mathcal{U}\frac{d}{4}h.$$

If $\nu < 0$, then v is a non-local Lyapunov function.

Proof. For all $\mathbf{x} \in \mathcal{U}$ we have by Theorem 3.3 for $u = Lv$

$$Lv(\mathbf{x}) \quad \leq \quad \max_{\mathbf{y}\in Y_\mathcal{U}} Lv(\mathbf{y}) + C_\mathcal{U}\frac{d}{4}h = \nu < 0.$$

Hence, v satisfies the assumptions of Definition 2.4. □

Theorem 4.3 (Second derivative)**.** *Let $v \in C^4$ be a function given by meshless collocation as described above. Let $\mathcal{A} = v^{-1}((-\infty, a])$ and $\mathcal{B} = v^{-1}((-\infty, b])$ and assume that $\mathcal{B} \subset \mathcal{A}^\circ$ and that $\mathcal{A}$ and $\mathcal{B}$ are simply connected compact compact neighbourhoods of the origin with C^2 boundaries. Set $\mathcal{U} := \mathcal{A} \setminus \mathcal{B}^\circ$. Fix $h > 0$.*

- *If $d = 1$, then let*

$$Y_\mathcal{U} := C_h \cap \mathcal{U}_{h/2,\|\cdot\|_1} \quad \text{and} \quad C_\mathcal{U} := \max_{\mathbf{z}\in\mathcal{U}_{h/2,\|\cdot\|_1}} |(Lv)''(\mathbf{z})|\,.$$

- *If d is even, then let*

$$Y_\mathcal{U} := S_h \cap \mathcal{U}_{d\,h,\|\cdot\|_1} \quad \text{and} \quad C_\mathcal{U} := \max_{\mathbf{z}\in\mathcal{U}_{d\,h,\|\cdot\|_1}} \max_{p,l\in\{1,\ldots,d\}} \left|\frac{\partial^2 Lv}{\partial x_p \partial x_l}(\mathbf{z})\right|.$$

- *If $d \geq 3$ is odd, then let $Y_\mathcal{U} := C_h \cap \mathcal{U}_{(d-1)\,h,\|\cdot\|_1}$ and*

$$C_\mathcal{U} := \max_{\mathbf{z}\in\mathcal{U}_{(d-1)\,h,\|\cdot\|_1}} \max_{p,l\in\{1,\ldots,d\}} \left|\frac{\partial^2 Lv}{\partial x_p \partial x_l}(\mathbf{z})\right|.$$

Let

$$\nu \quad := \quad \max_{\mathbf{y}\in Y_\mathcal{U}} Lv(\mathbf{y}) + C_\mathcal{U}\frac{d^2}{4}h^2$$

If $\nu < 0$, then v is a non-local Lyapunov function.

Proof. For all $\mathbf{x} \in \mathcal{U}$ we have with Theorem 3.4 for $u = Lv$

$$Lv(\mathbf{x}) \quad \leq \quad \max_{\mathbf{y}\in Y_\mathcal{U}} Lv(\mathbf{y}) + C_\mathcal{U}\frac{d^2}{4}h^2 = \nu < 0.$$

Hence, v satisfies the assumptions of Definition 2.4. □

Remark 4.4. Note that due to Lemma 4.1 we have $\mathcal{U} \subset \widetilde{\mathcal{U}}$ and thus we can replace $\mathcal{U}$ in the previous two theorems by $\widetilde{\mathcal{U}}$. However, we can use Theorems 4.2 and 4.3 directly with suitable a and b, without employing Lemma 4.1 as well.

5. Examples.

5.1. One-dimensional example.

We consider the example from [8]:

$$dx \quad = \quad \sin x \, dt + \frac{3x}{1+x^2} \, dW, \tag{5.1}$$

where W is a one-dimensional Wiener-process. As $\sin x$ and $3x/(1+x^2)$ are Lipschitz, this equation has a unique strong solution. As local Lyapunov function we take $U(x) = |x|^{1/2}$ as in [8]. Then

$$LU(x) = -\frac{1}{2}|x|^{1/2}\left(3^2 \frac{\frac{1}{2}}{2(1+x^2)^2} - \frac{\sin(x)}{x}\right)$$

and $LU(x) < 0$ for all $x \in [-2^{-1/2}, 2^{-1/2}] \setminus \{0\} =: \mathcal{B} \setminus \{0\}$. Therefore we can choose $\{\pm 2^{-1/2}\} = U^{-1}(U_{\max})$ with $U_{\max} = 2^{-1/4}$.

For the non-local Lyapunov function we just consider $x \geq 0$, since the SDE is symmetric. We use the Wendland function $\phi_{7,6}$ with coefficient $c = 2$. We set $\rho_1 = 10^{-2}$ and $\rho_2 = 8$ and determine an approximate solution to the equation

$$LV(x) = -10^{-3} \quad \text{ on } (\rho_1, \rho_2)$$

such that $V(\rho_1) = 0$ and $V(\rho_2) = 1$. We have chosen 700 collocation points evenly spaced in the interval $[1.1 \cdot 10^{-2}, 7.99]$.

The approximating function v and Lv are displayed in Figure 1. We obtain the values $\boldsymbol{\alpha}_1 = 653.0140$ and $\boldsymbol{\alpha}_2 = 0.9440$. Since in the 1-dimensional case the boundary values for the approximation are $v(\rho_1) = 0$ and $v(\rho_2) = 1$, we choose $a = 1$ and $b = 0$ and hence $\mathcal{U} = [\rho_1, \rho_2]$. We first use Theorem 3.5 on any compact set C with $F = F_1 = F_2 = 1$, $G = 9/8$, $G_1 = 1.9566$, and $G_2 = 9$ to obtain $\max_{z\in\mathbb{R}} |(Lv)''(z)| = 1.6846 \cdot 10^{12} =: C_{\mathcal{U}}$; for the values $\psi_{k,l}$ see Table 3.

We now use Theorem 4.3 and choose $h = 2.1307 \cdot 10^{-8}$, which corresponds to $7.5 \cdot 10^8$ evenly spaced points $Y_{\mathcal{U}} = C_h \cap [\rho_1 - h/2, \rho_2 + h/2] = \frac{h}{2}\mathbb{Z} \cap [\rho_1 - h/2, \rho_2 + h/2]$ on the interval. We obtain a maximum value of $\max_{y\in Y_{\mathcal{U}}} Lv(y) = -0.281 \cdot 10^{-3}$ and thus

$$\nu = \max_{y\in Y_{\mathcal{U}}} Lv(y) + C_{\mathcal{U}}\frac{h^2}{4} = -0.281 \cdot 10^{-3} + 0.19119 \cdot 10^{-3} < 0.$$

By Theorem 4.3, v is a non-local Lyapunov function.

Now we need to determine constants $0 < \beta < 1$ and $0 < \lambda < \alpha < 1$, see Theorem 2.5, such that

$$U^{-1}(U_{\max}) \subset v^{-1}([0, \lambda]) \quad \text{ and } \quad \partial\mathcal{B} = v^{-1}(0) \subset U^{-1}([0, \beta U_{\max}]).$$

Following calculations from [8] we compute a lower estimate $[-r_{1-\beta}, r_{1-\beta}]$ for the $(1-\beta)$-BOA of the equilibrium, by solving $U(r_{1-\beta}) = \beta U_{\max}$. Thus $r_{1-\beta} = \beta^2 2^{-1/2}$. Theorem 2.5 requires

$$\rho_1 = v^{-1}(0) \subset U^{-1}([0, \beta U_{\max}]),$$

which is equivalent to $\rho_1 = 10^{-2} < r_{1-\beta} = \beta^2 2^{-1/2}$, i.e. $\beta > 0.1189$. We need to find λ such that

$$U^{-1}(U_{\max}) \subset v^{-1}([0, \lambda])$$

which is equivalent to $V(2^{-1/2}) \leq \lambda$. We now fix $\beta = 0.1247$, $\lambda = 0.0421$ and we are free to choose $\alpha > \lambda$.

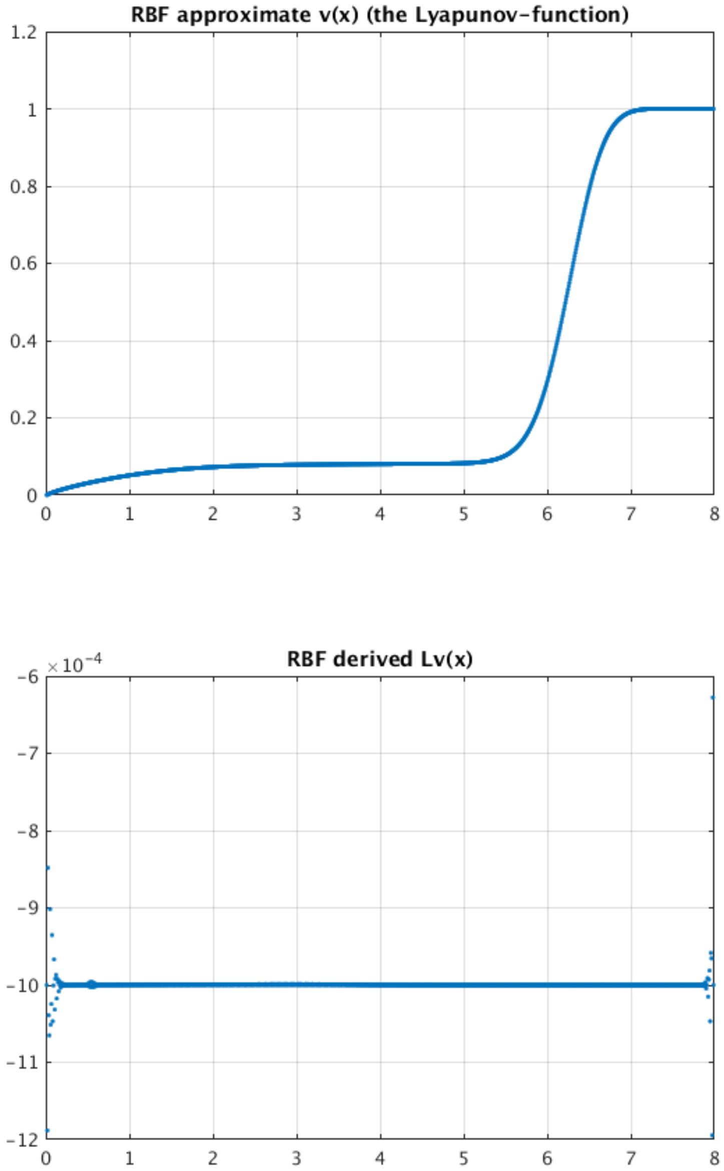


FIGURE 1. Above: the computed non-local Lyapunov function v for system (5.1). Below: the function Lv, approximating -10^{-3}.

Corresponding to our choice of α, we have that the set $v^{-1}([0,\alpha]) \cup \mathcal{B}$ is a subset of the γ-BOA by Theorem 2.5 (note that $b = 0$ and $a = 1$) with

$$\gamma = \frac{(1-\alpha)(1-\beta)}{1-\beta(1-\lambda)}.$$

For $\alpha = 0.044$ we have $V^{-1}([0,\alpha]) \cup \mathcal{B} \approx [-0.803, 0.803]$ and $\gamma \approx 0.95$.

For $\alpha = 0.09$ we have $V^{-1}([0,\alpha]) \cup \mathcal{B} \approx [-5.33, 5.33]$ and $\gamma \approx 0.90$.

Let us compare these results first to the local Lyapunov function: here we obtain $[-0.00177, 0.00177]$ and $[-0.00707, 0.00707]$ as lower estimates of the 0.95- and 0.90-BOAs. By comparing those values with the estimates obtained above we see a very substantial increase.

Our results are comparable to the results in [8] in that we obtained similarly sized γ-BOA, however, our method includes a rigorous verification (numerical proof) that v is indeed a non-local Lyapunov function. This verification is missing in [8] and one can only hope that the computed non-local function is a Lyapunov function for the system.

Lastly, we set up a simple Monte-Carlo simulation using the First-order stochastic Runge-Kutta method to generate 1000 approximate realisations of sample paths, starting at the point $x = 5.33$. We then check when they leave the interval $[10^{-4}, 8]$ and at which end. The result is that 98% of simulations leave through the inner boundary and 2% through the outer, which is close to what we expected since the point 5.33 is inside the 0.9-BOA. Note that this a larger value than predicted by our method. On the one hand, our estimate is indeed just a lower bound and exiting $[10^{-4}, 8]$ through the lower boundary is not the same as the sample trajectories converging to the origin as time tends to infinity. It confirms, however, the validity of our estimate.

5.2. **Two-dimensional example.** We consider the first example from [3, Section 4], namely

$$d\mathbf{x} \quad = \quad (M + \rho(\mathbf{x})I)\mathbf{x}dt + \mathbf{g}(\mathbf{x})dW, \tag{5.2}$$

where W is a one-dimensional Wiener-process, I is the 2×2 identity matrix, and with

$$M = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \ \rho(\mathbf{x}) = \|\mathbf{x}\| - 1, \text{ and } \mathbf{g}(\mathbf{x}) = \theta\|\mathbf{x}\| \left(\|\mathbf{x}\| - \frac{1}{2}\right)\left(\|\mathbf{x}\| - \frac{3}{2}\right)\mathbf{x}.$$

To assert the existence of unique strong solutions we use these formulas for $\mathbf{f}(\mathbf{x}) = (M + \rho(\mathbf{x})I)\mathbf{x}$ and $\mathbf{g}(\mathbf{x})$ inside of a ball, centered at the origin and with radius 4 and outside of this ball we extend $\mathbf{f}$ and $\mathbf{g}$ as Lipschitz functions. For this SDE the generator is

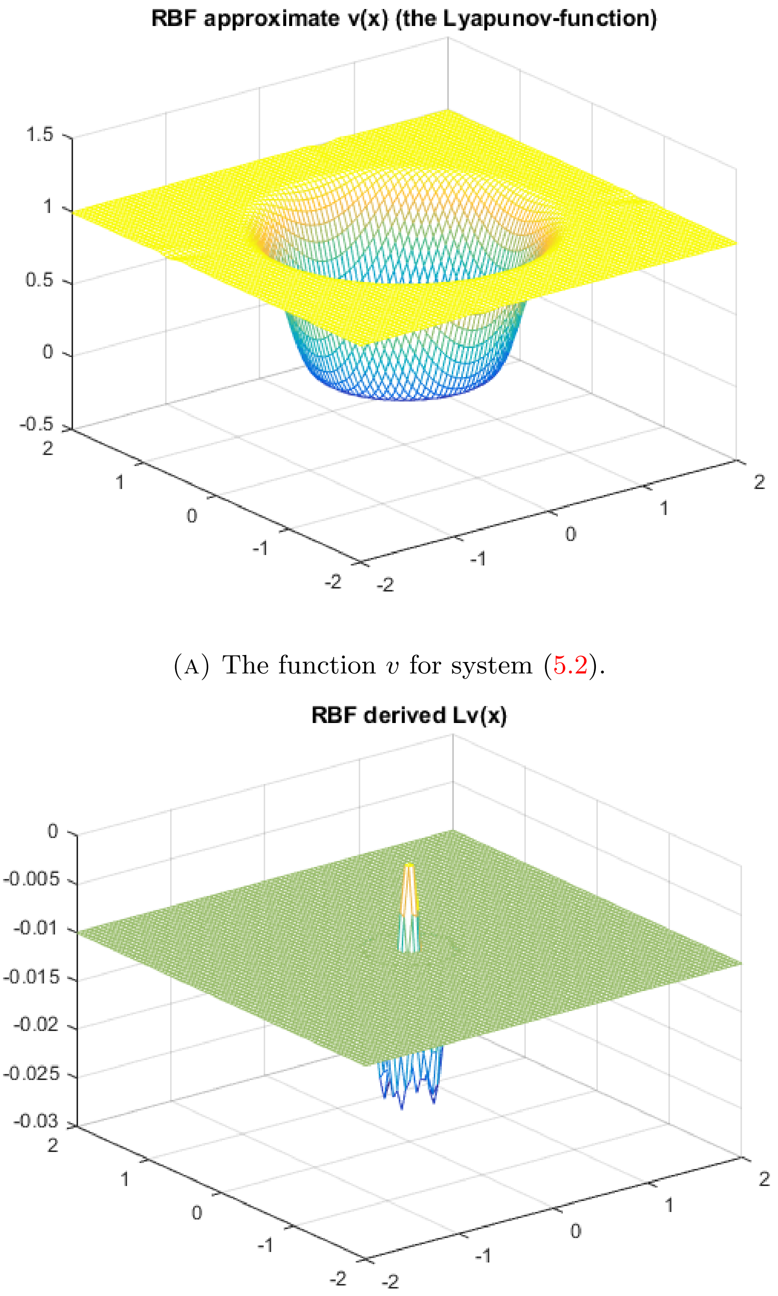
$$L := \frac{1}{2}\sum_{i,j=1}^{2} a_{ij}(\mathbf{x})\frac{\partial^2}{\partial x_i \partial x_j} + \sum_{i=1}^{2} f_i(\mathbf{x})\frac{\partial}{\partial x_i}, \text{ where } \ a(\mathbf{x}) := \mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^\top.$$

By solving the continuous time Lyapunov equation $J^\top P + PJ = -2I$ for the deterministic linearised system

$$\mathbf{x}' = J\mathbf{x} \ \text{ with } \ J = \begin{pmatrix} -1 & 1 \\ -1 & -1 \end{pmatrix} = D\mathbf{f}(\mathbf{0}),$$

we get the Lyapunov function $U(\mathbf{x}) = \|\mathbf{x}\|^2$. For our system this delivers with $\mathbf{x} = (x, y)$:

$$\begin{aligned} & U(\mathbf{x}) \\ & \quad = \frac{1}{2}\theta^2\|\mathbf{x}\|^2\left(\|\mathbf{x}\| - \frac{1}{2}\right)^2\left(\|\mathbf{x}\| - \frac{3}{2}\right)^2 (x^2 \cdot 2 + xy \cdot 0 + yx \cdot 0 + y^2 \cdot 2) \\ & \qquad + [(\|\mathbf{x}\| - 1)x + y] \cdot 2x + [-x + (\|\mathbf{x}\| - 1)y]2y \\ & \quad = \theta^2\|\mathbf{x}\|^4\left(\|\mathbf{x}\| - \frac{1}{2}\right)^2\left(\|\mathbf{x}\| - \frac{3}{2}\right)^2 + (\|\mathbf{x}\| - 1)(2x^2 + 2y^2) \\ & \quad = -\|\mathbf{x}\|^2\left(2 - 2\|\mathbf{x}\| - \theta^2\|\mathbf{x}\|^2\left(\|\mathbf{x}\| - \frac{1}{2}\right)^2\left(\|\mathbf{x}\| - \frac{3}{2}\right)^2\right). \end{aligned}$$



(A) The function v for system (5.2).

(B) The function Lv for system (5.2), approximating -10^{-2}.

FIGURE 2. Non-local Lyapunov function for system (5.2) with $\theta =$ 1. The non-local Lyapunov functions looks very similar to the one computed in [3].

Set

$$h_\theta(r) = 2 - 2r - \theta^2 r^2 \left(r - \frac{1}{2}\right)^2 \left(r - \frac{3}{2}\right)^2 .$$

Then $LU(\mathbf{x}) = -\|\mathbf{x}\|^2 h_\theta(\|\mathbf{x}\|)$ and routine calculations show that on the interval $[0, 1/2]$ the function $r \mapsto r^2(r - \frac{1}{2})^2(r - \frac{3}{2})^2$ takes its largest value at $r^* = (4 -$

$\sqrt{7})/6 \approx 0.22571$ and that $h_\theta(r^*) > 1.55 - 6.3 \cdot 10^{-3}\theta^2$, so for any $0 \leq \theta \leq 15.56$ the function $U(\mathbf{x}) = \|\mathbf{x}\|^2$ is a Lyapunov function for the system on $B_{1/2}(0)$. It is not difficult to verify that if $0 \leq \theta \leq 1$, then U is a (local) Lyapunov function on $B_{0.9}(0)$.

Now we calculate the constants for $K = \{\mathbf{x} \in \mathbb{R}^2 : R_1 \leq \|\mathbf{x}\| \leq R_2\}$ with $R_2 = 2$. We have, see appendix, $F = R_2\sqrt{1 + (R_2 - 1)^2} = 2\sqrt{2}$, $F_1 = \sqrt{12}$, $F_2 = 2$, $G = \frac{9}{2}\theta^2$, $G_1 = 33\theta^2$, and $G_2 = 197.5\theta^2$.

We used the Wendland function $\phi_{8,6}$ with $c = 1$, for the system (5.2) with $\theta = 1$. We choose $\rho_1 = 0.4$ and $\rho_2 = 1.9$ and use a 80×80 grid of collocation points on $[-2, 2] \times [-2, 2]$ to calculate a non-local Lyapunov function, approximating $LV(\mathbf{x}) = -10^{-2}$, see Figure 2. With $\boldsymbol{\alpha}_1 = 401.4572$ and $\boldsymbol{\alpha}_2 = 5.8372$ we obtain the value $C_{\mathcal{U}} = 4.3220 \cdot 10^{12}$. By evaluating LV on a relatively coarse 1000×1000 grid of points on $[-2, 2] \times [-2, 2]$, we estimated the maximum value of LV not to exceed -0.005. Hence, we require a checking grid with $h = 3.4013 \cdot 10^{-8}$ and thus we need to evaluate LV at $(1.1760 \cdot 10^8)^2 \approx 10^{16}$ points. Our current software and computer setup is not adequate to complete those calculations in a reasonable time frame, but we note that the verification workload is perfectly parallel which can be used to speed up the calculations. The necessary estimates for these computations are included in the appendix for future reference.

Now similarly to Example 5.1, we have to determine constants $0 < \beta < 1$ and $0 < \lambda < \alpha < 1$ (see Theorem 2.5), such that

$$U^{-1}(U_{\max}) \subset v^{-1}([0, \lambda]) \quad \text{and} \quad \partial\mathcal{B} = v^{-1}(0) \subset U^{-1}([0, \beta U_{\max}]),$$

where $U(\mathbf{x}) = \|\mathbf{x}\|^2$ is the local Lyapunov function on $B_{1/2}(0)$. We calculate a lower estimate for the $(1-\beta)$-BOA, $\{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\| \leq r_{1-\beta}\}$, of the equilibrium by choosing $\overline{B_{1/2}(0)} = U^{-1}([0, U_{\max}])$, i.e. $U_{\max} = 1/4$, and solving $\overline{B_{r_{1-\beta}}(0)} = U^{-1}([0, \beta U_{\max}])$. Thus $r_{1-\beta} = \frac{\sqrt{\beta}}{2}$. Theorem 2.5 requires

$$v^{-1}(0) \subset U^{-1}([0, \beta U_{\max}])$$

which is equivalent to $0.4 \leq r_{1-\beta} = \sqrt{\beta}2$, i.e. $\beta > 0.64$. Now we need to find λ such that

$$U^{-1}(U_{\max}) \subset v^{-1}([0, \lambda]).$$

We now fix $\beta = 0.65$, $\lambda = 0.005$ and we are free to choose $\alpha > \lambda$. Corresponding to our choice of α we have that the set $v^{-1}((0, \alpha))$ is a subset of the γ-BOA by Theorem 2.5 (with $b = 0$ and $a = 0$) with

$$\gamma = \frac{(1 - \alpha)(1 - \beta)}{1 - \beta(1 - \lambda)}.$$

For $\alpha = 0.01$ we have $v^{-1}([0, \alpha]) \cup \mathcal{B} \approx B_{0.6454}(0)$ and $\gamma \approx 0.9809$

For $\alpha = 0.09$ we have $v^{-1}([0, \alpha]) \cup \mathcal{B} \approx B_{0.839}(0)$ and $\gamma \approx 0.90$.

Let us compare these results to the local Lyapunov function: Here we obtain $B_{0.0707}(0)$ and $B_{0.1581}(0)$ as lower estimates of the 0.98 and 0.90-BOAs. By comparing the estimates above we see a substantial increase.

Our results are comparable to the results in [3] in that we obtained similarly sized γ-BOA. However, our method includes a framework for rigorous verification (numerical proof) that v is indeed a non-local Lyapunov function, although this verification could not be performed at this point due to its huge computational demand.

Acknowledgment. This research and, in particular, Björnsson and Gudmundsson were supported by The Icelandic Research Fund with the project *Lyapunov Methods and Stochastic Stability* (grant nr. 152429-051).

Appendix A. **Explicit formulas for meshless collocation.** We calculate $v(\mathbf{x})$, $Lv(\mathbf{x})$, and the collocation matrix A for the specific operator L given in (3.3). We denote recursively $\psi_{i+1}(r) = \frac{1}{r}\frac{\partial}{\partial r}\psi_i(r)$ for $i = 0, 1, \ldots, 5$ and $\psi_0 = \psi$, where ψ is a certain Wendland functions that can be found below in Tables 1 and 2. Recall that $\|\cdot\| = \|\cdot\|_2$.

We have, see (3.5), that

$$\begin{aligned} v(\mathbf{x}) \quad = \quad & \sum_{k=1}^{N} \alpha_k \Big[-\psi_1(\|\mathbf{x}-\mathbf{x}_k\|)\langle \mathbf{x}-\mathbf{x}_k, f(\mathbf{x}_k)\rangle \\ & \qquad + \frac{1}{2}\sum_{i,j=1}^{d} m_{ij}(\mathbf{x}_k)[\psi_2(\|\mathbf{x}-\mathbf{x}_k\|)(\mathbf{x}-\mathbf{x}_k)_i(\mathbf{x}-\mathbf{x}_k)_j \\ & \qquad\qquad + \delta_{ij}\psi_1(\|\mathbf{x}-\mathbf{x}_k\|)]\Big] \\ & + \sum_{k=1}^{M} \alpha_{N+k}\psi_0(\|\mathbf{x}-\boldsymbol{\xi}_k\|). \end{aligned} \tag{A.1}$$

The formula for $Lv(x)$ is, abbreviating $\boldsymbol{\beta} = \mathbf{x} - \mathbf{x}_k$,

$$\begin{aligned} & Lv(\mathbf{x}) \\ = \; & \sum_{k=1}^{N} \alpha_k \Big\{ -\psi_2(\|\boldsymbol{\beta}\|)\langle \boldsymbol{\beta}, f(\mathbf{x})\rangle\langle \boldsymbol{\beta}, f(\mathbf{x}_k)\rangle - \psi_1(\|\boldsymbol{\beta}\|)\langle f(\mathbf{x}), f(\mathbf{x}_k)\rangle \\ & + \frac{1}{2}\sum_{i,j=1}^{d} m_{ij}(\mathbf{x}_k)\Big[\psi_3(\|\boldsymbol{\beta}\|)\langle \boldsymbol{\beta}, f(\mathbf{x})\rangle\beta_i\beta_j + \psi_2(\|\boldsymbol{\beta}\|)f_j(\mathbf{x})\beta_i \\ & \qquad + \psi_2(\|\boldsymbol{\beta}\|)f_i(\mathbf{x})\beta_j + \delta_{ij}\psi_2(\|\boldsymbol{\beta}\|)\langle \boldsymbol{\beta}, f(\mathbf{x})\rangle\Big] \\ & + \frac{1}{2}\sum_{i,j=1}^{d} m_{ij}(\mathbf{x})\Big[-\psi_3(\|\boldsymbol{\beta}\|)\langle \boldsymbol{\beta}, f(\mathbf{x}_k)\rangle\beta_i\beta_j - \psi_2(\|\boldsymbol{\beta}\|)f_j(\mathbf{x}_k)\beta_i \\ & \qquad - \psi_2(\|\boldsymbol{\beta}\|)f_i(\mathbf{x}_k)\beta_j - \delta_{ij}\psi_2(\|\boldsymbol{\beta}\|)\langle \boldsymbol{\beta}, f(\mathbf{x}_k)\rangle\Big] \\ & + \frac{1}{4}\sum_{r,s=1}^{d}\sum_{i,j=1}^{d} m_{rs}(\mathbf{x})m_{ij}(\mathbf{x}_k)\Big[\psi_4(\|\boldsymbol{\beta}\|)\beta_i\beta_j\beta_r\beta_s \\ & \qquad + \psi_3(\|\boldsymbol{\beta}\|)[\delta_{ij}\beta_r\beta_s + \delta_{ir}\beta_j\beta_s \\ & \qquad\qquad + \delta_{is}\beta_j\beta_r + \delta_{jr}\beta_i\beta_s + \delta_{js}\beta_i\beta_r + \delta_{rs}\beta_i\beta_j] \\ & \qquad + \psi_2(\|\boldsymbol{\beta}\|)[\delta_{ij}\delta_{rs} + \delta_{ir}\delta_{js} + \delta_{is}\delta_{jr}]\Big]\Big\} \\ & + \sum_{k=1}^{M} \alpha_{N+k}\Big\{ -\psi_1(\|\boldsymbol{\xi}_k - \mathbf{x}\|)\langle \boldsymbol{\xi}_k - \mathbf{x}, f(\mathbf{x})\rangle \end{aligned}$$

$$+\frac{1}{2}\sum_{i,j=1}^{d} m_{ij}(\mathbf{x})[\psi_2(\|\boldsymbol{\xi}_k-\mathbf{x}\|)(\boldsymbol{\xi}_k-\mathbf{x})_i(\boldsymbol{\xi}_k-\mathbf{x})_j$$
$$\left.+\delta_{ij}\psi_1(\|\boldsymbol{\xi}_k-\mathbf{x}\|)]\right\}.$$

The formulas for the matrix elements are

$$\begin{aligned}
d_{kl} &= \psi_0(\|\boldsymbol{\xi}_k-\boldsymbol{\xi}_l\|),\\
c_{kl} &= -\psi_1(\|\boldsymbol{\xi}_l-\mathbf{x}_k\|)\langle\boldsymbol{\xi}_l-\mathbf{x}_k, f(\mathbf{x}_k)\rangle\\
&\quad+\frac{1}{2}\sum_{i,j=1}^{d} m_{ij}(\mathbf{x}_k)[\psi_2(\|\boldsymbol{\xi}_l-\mathbf{x}_k\|)(\boldsymbol{\xi}_l-\mathbf{x}_k)_i(\boldsymbol{\xi}_l-\mathbf{x}_k)_j\\
&\quad+\delta_{ij}\psi_1(\|\boldsymbol{\xi}_l-\mathbf{x}_k\|)],
\end{aligned}$$

and, abbreviating $\boldsymbol{\beta}=\mathbf{x}_k-\mathbf{x}_l$,

$$\begin{aligned}
b_{kl} &= -\psi_2(\|\boldsymbol{\beta}\|)\langle\boldsymbol{\beta}, f(\mathbf{x}_k)\rangle\langle\boldsymbol{\beta}, f(\mathbf{x}_l)\rangle-\psi_1(\|\boldsymbol{\beta}\|)\langle f(\mathbf{x}_k), f(\mathbf{x}_l)\rangle\\
&\quad+\frac{1}{2}\sum_{i,j=1}^{d} m_{ij}(\mathbf{x}_l)\Big[\psi_3(\|\boldsymbol{\beta}\|)\langle\boldsymbol{\beta}, f(\mathbf{x}_k)\rangle\beta_i\beta_j+\psi_2(\|\ \boldsymbol{\beta}\|)f_j(\mathbf{x}_k)\beta_i\\
&\qquad+\psi_2(\|\boldsymbol{\beta}\|)f_i(\mathbf{x}_k)\beta_j+\delta_{ij}\psi_2(\|\boldsymbol{\beta}\|)\langle\boldsymbol{\beta}, f(\mathbf{x}_k)\rangle\Big]\\
&\quad+\frac{1}{2}\sum_{i,j=1}^{d} m_{ij}(\mathbf{x}_k)\Big[-\psi_3(\|\boldsymbol{\beta}\|)\langle\boldsymbol{\beta}, f(\mathbf{x}_l)\rangle\beta_i\beta_j-\psi_2(\|\boldsymbol{\beta}\|)f_j(\mathbf{x}_l)\beta_i\\
&\qquad-\psi_2(\|\boldsymbol{\beta}\|)f_i(\mathbf{x}_l)\beta_j-\delta_{ij}\psi_2(\|\boldsymbol{\beta}\|)\langle\boldsymbol{\beta}, f(\mathbf{x}_l)\rangle\Big]\\
&\quad+\frac{1}{4}\sum_{r,s=1}^{d}\sum_{i,j=1}^{d} m_{rs}(\mathbf{x}_k)m_{ij}(\mathbf{x}_l)\Big[\psi_4(\|\boldsymbol{\beta}\|)\beta_i\beta_j\beta_r\beta_s\\
&\qquad+\psi_3(\|\boldsymbol{\beta}\|)[\delta_{ij}\beta_r\beta_s+\delta_{ir}\beta_j\beta_s+\delta_{is}\beta_j\beta_r\\
&\qquad\qquad+\delta_{jr}\beta_i\beta_s+\delta_{js}\beta_i\beta_r+\delta_{rs}\beta_i\beta_j]\\
&\qquad+\psi_2(\|\boldsymbol{\beta}\|)[\delta_{ij}\delta_{rs}+\delta_{ir}\delta_{js}+\delta_{is}\delta_{jr}]\Big].
\end{aligned}$$

Appendix B. **Two-dimensional example.** In this section we give the details of the estimates for F_i and G_i of the 2-dimensional example from Section 5.2.

With $\mathbf{f}(x_1,x_2)=\begin{pmatrix}(\|\mathbf{x}\|-1)x_1+x_2\\-x_1+(\|\mathbf{x}\|-1)x_2\end{pmatrix}$ we obtain

$$\begin{aligned}
F &= \Big((\|\mathbf{x}\|-1)^2x_1^2+x_2^2+2x_1x_2(\|\mathbf{x}\|-1)\\
&\qquad+x_1^2-2x_1x_2(\|\mathbf{x}\|-1)+x_2^2(\|\mathbf{x}\|-1)^2\Big)^{1/2}\\
&= \|\mathbf{x}\|\sqrt{(\|\mathbf{x}\|-1)^2+1},\\
\frac{\partial\mathbf{f}}{\partial x_1} &= \begin{pmatrix}\frac{x_1^2}{\|\mathbf{x}\|}+\|\mathbf{x}\|-1\\-1+\frac{x_1x_2}{\|\mathbf{x}\|}\end{pmatrix}=\begin{pmatrix}\frac{2x_1^2+x_2^2}{\|\mathbf{x}\|}-1\\-1+\frac{x_1x_2}{\|\mathbf{x}\|}\end{pmatrix},
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathbf{f}}{\partial x_2} &= \begin{pmatrix} \frac{x_1x_2}{\|\mathbf{x}\|}+1 \\ \frac{x_2^2}{\|\mathbf{x}\|}+\|\mathbf{x}\|-1 \end{pmatrix} = \begin{pmatrix} \frac{x_1x_2}{\|\mathbf{x}\|}+1 \\ \frac{x_1^2+2x_2^2}{\|\mathbf{x}\|}-1 \end{pmatrix}, \\
\left\|\frac{\partial \mathbf{f}}{\partial x_2}\right\|^2 &= \frac{x_1^2x_2^2+x_1^4+4x_1^2x_2^2+4x_2^4}{\|\mathbf{x}\|^2} + \frac{2x_1x_2-2x_1^2-4x_2^2}{\|\mathbf{x}\|}+2 \\
&\leq x_1^2+4x_2^2-\frac{x_1^2+3x_2^2}{\|\mathbf{x}\|}+2 \\
&= x_1^2\left(1-\frac{1}{\|\mathbf{x}\|}\right)+x_2^2\left(4-\frac{3}{\|\mathbf{x}\|}\right)+2 \\
&\leq x_1^2\left(1-\frac{1}{R_2}\right)+x_2^2\left(4-\frac{3}{R_2}\right)+2 \\
&\leq R_2^2 \max\left(0,\left(1-\frac{1}{R_2}\right),\left(4-\frac{3}{R_2}\right)\right)+2, \\
\frac{\partial^2 \mathbf{f}}{\partial x_1^2} &= \begin{pmatrix} \frac{x_1(2x_1^2+3x_2^2)}{\|\mathbf{x}\|^3} \\ \frac{x_2^3}{\|\mathbf{x}\|^3} \end{pmatrix}, \\
\frac{\partial^2 \mathbf{f}}{\partial x_1 \partial x_2} &= \begin{pmatrix} \frac{x_2^3}{\|\mathbf{x}\|^3} \\ \frac{x_1^3}{\|\mathbf{x}\|^3} \end{pmatrix}, \\
\frac{\partial^2 \mathbf{f}}{\partial x_2^2} &= \begin{pmatrix} \frac{x_1^3}{\|\mathbf{x}\|^3} \\ \frac{x_2(3x_1^2+2x_2^2)}{\|\mathbf{x}\|^3} \end{pmatrix}, \quad \text{and} \\
\left\|\frac{\partial^2 \mathbf{f}}{\partial x_2^2}\right\| &= \frac{\sqrt{x_1^6+9x_1^4x_2^2+12x_1^2x_2^4+4x_2^6}}{\|\mathbf{x}\|^3} \\
&\leq \frac{\sqrt{4x_1^6+12x_1^4x_2^2+12x_1^2x_2^4+4x_2^6}}{\|\mathbf{x}\|^3} \\
&= \frac{\sqrt{4(x_1^2+x_2^2)^3}}{\|\mathbf{x}\|^3} = 2.
\end{aligned}$$

We now calculate the estimates for $\mathbf{g}(\mathbf{x}) = \theta r(r-0.5)(r-1.5)\mathbf{x}$, denoting $\|\mathbf{x}\| = r$. For $r \in [0,2]$ we have

$$\|\mathbf{g}(\mathbf{x})\| \leq \theta 4 \cdot \frac{3}{2} \cdot \frac{1}{2} = 3\theta.$$

Furthermore,

$$\frac{\partial \mathbf{g}}{\partial x_i} = \theta\left\{\frac{x_i}{\|\mathbf{x}\|}\left[3r^2-4r+\frac{3}{4}\right]\mathbf{x}+r(r-0.5)(r-1.5)\mathbf{e}_i\right\}$$

Hence,

$$\begin{aligned}
\left\|\frac{\partial \mathbf{g}}{\partial x_i}\right\| &\leq \theta\left(\left|3r^2-4r+\frac{3}{4}\right|r+|r(r-0.5)(r-1.5)|\right) \\
&\leq 11\theta
\end{aligned}$$

for $r \in [0,2]$.

Finally,

$$\begin{aligned}
\frac{\partial^2 \mathbf{g}}{\partial x_1^2} &= \theta\left[3r^2-4r+\frac{3}{4}\right]\left(\frac{x_2^2}{\|\mathbf{x}\|^3}\mathbf{x}+2\frac{x_1}{\|\mathbf{x}\|}\mathbf{e}_1\right)+\theta\frac{x_1^2}{\|\mathbf{x}\|^2}(6r-4)\mathbf{x}\\
&= \frac{\theta}{r^3}\left[\left(3r^2-4r+\frac{3}{4}\right)x_2^2+(6r^2-4r)x_1^2\right]\mathbf{x}\\
&\quad +2\theta\left(3r^2-4r+\frac{3}{4}\right)\frac{x_1}{\|\mathbf{x}\|}\mathbf{e}_1,\\
\left\|\frac{\partial^2 \mathbf{g}}{\partial x_1^2}\right\| &\leq \theta\max\left(\left|3r^2-4r+\frac{3}{4}\right|,\left|6r^2-4r\right|\right)\\
&\quad +2\theta\left|3r^2-4r+\frac{3}{4}\right|\\
&\leq 25.5\theta \text{ for } r\in[0,2],\\
\frac{\partial^2 \mathbf{g}}{\partial x_1\partial x_2} &= \theta\left[3r^2-4r+\frac{3}{4}\right]\left(\frac{-x_1x_2}{\|\mathbf{x}\|^3}\mathbf{x}+\frac{x_2}{\|\mathbf{x}\|}\mathbf{e}_1+\frac{x_1}{\|\mathbf{x}\|}\mathbf{e}_2\right)\\
&\quad +\theta\frac{x_1x_2}{\|\mathbf{x}\|^2}(6r-4)\mathbf{x}\\
&= \frac{\theta}{r^3}\left[-3r^2+4r-\frac{3}{4}+6r^2-4r\right]x_1x_2\mathbf{x}\\
&\quad +\frac{\theta}{r}\left(3r^2-4r+\frac{3}{4}\right)(x_2\mathbf{e}_1+x_1\mathbf{e}_2),\quad\text{and}\\
\left\|\frac{\partial^2 \mathbf{g}}{\partial x_1\partial x_2}\right\| &\leq \theta\left|3r^2-\frac{3}{4}\right|+\theta\left|3r^2-4r+\frac{3}{4}\right|\\
&\leq 16\theta.
\end{aligned}$$

Appendix C. **Wendland functions.** In this appendix we give the explicit formulas of the Wendland functions $\phi_{8,6}$ and $\phi_{7,6}$ as well as the corresponding auxiliary functions ψ_i, $i=1,\ldots,6$. Furthermore, we give the relevant estimates for $\psi_{k,i}=\sup_{r\in[0,\infty)}|\psi_k(r)|r^i$.

In particular, in Table 1 and Table 2, we give the formulas for the Wendland function $\psi_0(r)=\phi_{8,6}(cr)$ and $\psi_0(r)=\phi_{7,6}(cr)$, respectively, as well as ψ_i, $i=1,\ldots,6$. In Table 3 we give the formulas for the expressions $\psi_{k,i}=\sup_{r\in[0,\infty)}\psi_k(r)r^i$, required for the estimates for the same Wendland functions $\phi_{8,6}$ and $\phi_{7,6}$. Note that $x_+:=\max\{x,0\}$.

REFERENCES

[1] 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 2018) - Volume 1*, 2018, 579–586, .

[2] M. Buhmann, *Radial Basis Functions: Theory and Implementations*, volume 12 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2003.

[3] F. Camilli and L. Grüne, Characterizing attraction probabilities via the stochastic Zubov equation, *Discrete Contin. Dyn. Syst. Ser. B*, **3** (2003), 457–468.

[4] P. Giesl, *Construction of Global Lyapunov functions using Radial Basis Functions*, volume 1904 of Lecture Notes in Mathematics, Springer, Berlin, 2007.

[5] P. Giesl and S. Hafstein, Review of computational methods for Lyapunov functions, *Discrete Contin. Dyn. Syst. Ser. B*, **20** (2015), 2291–2331.

	$\phi_{8,6}$
$\psi_0(r)$	$[46,189(cr)^6 + 73,206(cr)^5 + 54,915(cr)^4 + 24,500(cr)^3$ $+6,755(cr)^2 + 1,078cr + 77]\,(1-cr)_+^{14}$
$\psi_1(r)$	$-380\,c^2\,[2,431(cr)^5 + 2,931(cr)^4 + 1,638(cr)^3 + 518(cr)^2$ $+91cr + 7]\,(1-cr)_+^{13}$
$\psi_2(r)$	$12,920\,c^4\,[1,287(cr)^4 + 1,108(cr)^3 + 426(cr)^2$ $+84cr + 7]\,(1-cr)_+^{12}$
$\psi_3(r)$	$-620,160\,c^6\,[429(cr)^3 + 239(cr)^2 + 55cr + 5]\,(1-cr)_+^{11}$
$\psi_4(r)$	$112,869,120\,c^8\,[33(cr)^2 + 10cr + 1]\,(1-cr)_+^{10}$
$\psi_5(r)$	$-4,966,241,280\,c^{10}\,[9cr + 1]\,(1-cr)_+^{9}$
$\psi_6(r)$	$446,961,715,200\,c^{12}\,(1-cr)_+^{8}$

TABLE 1. The table shows the Wendland function $\psi_0(r) := \phi_{8,6}(cr)$ as well as the related functions ψ_1 to ψ_6, defined recursively by $\psi_{k+1}(r) := \frac{\partial_r \psi_k(r)}{r}$ for $k = 0, 1, \ldots, 5$.

	$\phi_{7,6}$
$\psi_0(r)$	$[4,096(cr)^6 + 7,059(cr)^5 + 5,751(cr)^4 + 2,782(cr)^3 + 830(cr)^2$ $+143cr + 11]\,(1-cr)_+^{13}$
$\psi_1(r)$	$-38\,c^2\,[2,048(cr)^5 + 2,697(cr)^4 + 1,644(cr)^3 + 566(cr)^2$ $+108cr + 9]\,(1-cr)_+^{12}$
$\psi_2(r)$	$10,336\,c^4\,[128(cr)^4 + 121(cr)^3 + 51(cr)^2 + 11cr + 1]\,(1-cr)_+^{11}$
$\psi_3(r)$	$-62,016\,c^6\,[320(cr)^3 + 197(cr)^2 + 50cr + 5]\,(1-cr)_+^{10}$
$\psi_4(r)$	$3,224,832\,c^8\,[80(cr)^2 + 27cr + 3]\,(1-cr)_+^{9}$
$\psi_5(r)$	$-354,731,520\,c^{10}\,[8cr + 1]\,(1-cr)_+^{8}$
$\psi_6(r)$	$25,540,669,440\,c^{12}\,(1-cr)_+^{7}$

TABLE 2. The table shows the Wendland function $\psi_0(r) := \phi_{7,6}(cr)$ as well as the related functions ψ_1 to ψ_6, defined recursively by $\psi_{k+1}(r) := \frac{\partial_r \psi_k(r)}{r}$ for $k = 0, 1, \ldots, 5$.

[6] P. Giesl and N. Mohammed, Verification estimates for the construction of Lyapunov functions using meshfree collocation, Discrete Contin. Dyn. Syst. Ser. B, in press.

[7] P. Giesl and H. Wendland, Meshless collocation: Error estimates with application to dynamical systems, *SIAM J. Numer. Anal.*, **45** 92007), 1723–1741.

[8] S. Gudmundsson and S. Hafstein, Probabilistic basin of attraction and its estimation using two Lyapunov functions, *Complexity*, **2018** (2018), Article ID 2895658, 9 pages.

[9] 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*, **2** (2018), 939–956.

[10] N. Mohammed, *Grid Refinement and Verification Estimates for the RBF Construction Method of Lyapunov Functions*, PhD thesis, University of Sussex, 2016.

[11] M. J. D. Powell, The theory of radial basis function approximation in 1990, In *Advances in Numerical Analysis, Vol. II (Lancaster, 1990)*, Oxford Sci. Publ., pages 105–210. Oxford Univ. Press, New York, 1992.

[12] R. Schaback and H. Wendland, Kernel techniques: From machine learning to meshless methods, *Acta Numer.*, **15** (2006), 543–639.

$\psi_{k,i}$	$\phi_{8,6}$	$\phi_{7,6}$
$\psi_{6,6}$	$3.148511062 \cdot 10^7 \cdot c^6$	$3.240130299 \cdot 10^6 \cdot c^6$
$\psi_{5,5}$	$2.363249538 \cdot 10^6 \cdot c^5$	$2.588617377 \cdot 10^5 \cdot c^5$
$\psi_{5,4}$	$6.409097287 \cdot 10^6 \cdot c^6$	$6.534280933 \cdot 10^5 \cdot c^6$
$\psi_{4,4}$	$1.947997580 \cdot 10^5 \cdot c^4$	$2.262550039 \cdot 10^4 \cdot c^4$
$\psi_{4,3}$	$6.000016519 \cdot 10^5 \cdot c^5$	$6.515237949 \cdot 10^4 \cdot c^5$
$\psi_{4,2}$	$2.215560450 \cdot 10^6 \cdot c^6$	$2.237953342 \cdot 10^5 \cdot c^6$
$\psi_{3,3}$	$1.807542870 \cdot 10^4 \cdot c^3$	$2.219149087 \cdot 10^3 \cdot c^3$
$\psi_{3,2}$	$6.618581621 \cdot 10^4 \cdot c^4$	$7.625999381 \cdot 10^3 \cdot c^4$
$\psi_{3,1}$	$3.172360616 \cdot 10^5 \cdot c^5$	$3.414789975 \cdot 10^4 \cdot c^5$
$\psi_{3,0}$	$3.1008 \cdot 10^6 \cdot c^6$	$3.1008 \cdot 10^5 \cdot c^6$
$\psi_{2,2}$	$1.970990855 \cdot 10^3 \cdot c^2$	$2.550970282 \cdot 10^2 \cdot c^2$
$\psi_{2,1}$	$9.418422390 \cdot 10^3 \cdot c^3$	$1.147899628 \cdot 10^3 \cdot c^3$
$\psi_{2,0}$	$9.044 \cdot 10^4 \cdot c^4$	$1.0336 \cdot 10^4 \cdot c^4$
$\psi_{1,1}$	$2.767275907 \cdot 10^2 \cdot c$	$3.766803387 \cdot 10^1 \cdot c$
$\psi_{1,0}$	$2.66 \cdot 10^3 \cdot c^2$	$3.42 \cdot 10^2 \cdot c^2$

TABLE 3. The table shows values for $\psi_{k,i} := \sup_{r\in[0,\infty)} |\psi_i(r)|r^k$ for the Wendland functions $\psi_0(r) := \phi_{8,6}(cr)$ and $\psi_0(r) := \phi_{7,6}(cr)$.

[13] H. Wendland, Error estimates for interpolation by compactly supported radial basis functions of minimal degree, *J. Approx. Theory*, **93** (1998), 258–272.

[14] H. Wendland, *Scattered Data Approximation*, volume 17 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2005.

Received April 2018; revised October 2018.

E-mail address: hjb6@hi.is
E-mail address: P.A.Giesl@sussex.ac.uk
E-mail address: skgu@sek.se
E-mail address: shafstein@hi.is
E-mail address: E.Scalas@sussex.ac.uk

Paper III

Local Lyapunov Functions for Nonlinear Stochastic Differential Equations by Linearization

Bjornsson, H. and Giesl, P. and Gudmundsson, S. and Hafstein, S.

Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO)

Reproduced with permission SciTePress, Science and Technology Publications, Lda

Local Lyapunov Functions for Nonlinear Stochastic Differential Equations by Linearization

Hjörtur Björnsson[1], Peter Giesl[2], Skuli Gudmundsson[3] and Sigurdur Hafstein[1]

[1]*Science Institute and Faculty of Physical Sciences, University of Iceland, Dunhagi 5, 107 Reykjavík, Iceland*

[2]*Department of Mathematics, University of Sussex, Falmer, BN1 9QH, U.K.*

[3]*Svensk Exportkredit, Klarabergsviadukten 61-63, 11164 Stockholm, Sweden*

Keywords: Stochastic Differential Equation, Lyapunov Function, Linearization, Asymptotic Stability in Probability.

Abstract: We present a rigid estimate of the domain, on which a Lyapunov function for the linearization of a nonlinear stochastic differential equation is a Lyapunov function for the original system. By using this estimate the demanding task of computing a lower bound on the γ-basin of attraction for a nonlinear stochastic systems is greatly simplified and the application of a resent numerical method for the same purpose facilitated.

1 INTRODUCTION

When analysing the stability of an equilibrium of a nonlinear deterministic system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$, one often resorts to linearization around the equilibrium. Assuming, without restriction of generality, that the equilibrium in question is at the origin, then one analyzes the stability of the origin for the system $\dot{\mathbf{x}} = A\mathbf{x}$, where $A := D\mathbf{f}(\mathbf{0})$ is the Jacobian of $\mathbf{f}$ at the origin. Now, if the matrix A is Hurwitz, i.e. the real-parts of the eigenvalues of A are all strictly negative, then one can solve the Lyapunov equation $A^\top P + PA = -Q$, where $Q \in \mathbb{R}^{d\times d}$ is an arbitrary symmetric and positive definite matrix. The solution $P \in \mathbb{R}^{d\times d}$ is then symmetric and positive definite and $V(\mathbf{x}) = \mathbf{x}^\top P\mathbf{x}$ is a Lyapunov function for the system, i.e. V has a minimum at the equilibrium at the origin and the derivative of V along solution trajectories of the linearized system fulfills

$$\nabla V(\mathbf{x}) \bullet A\mathbf{x} = -\mathbf{x}^\top Q\mathbf{x}$$

and is thus negative on $\mathbb{R}^d \setminus \{\mathbf{0}\}$. The function V will also be a Lyapunov function for the original nonlinear system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ on a neighbourhood $\mathcal{N}$ of the origin where

$$V'(\mathbf{x}) = \nabla V(\mathbf{x}) \bullet \mathbf{f}(\mathbf{x}) < 0 \quad \text{for } \mathbf{x} \in \mathcal{N} \setminus \{\mathbf{0}\}.$$

Here V' denotes the orbital derivative of the system. The size of the set $\mathcal{N}$ is of great importance because compact sublevel sets of V that are within $\mathcal{N}$ are lower bounds on the equilibrium's basin of attraction, i.e. the set of points which converge to the equilibrium as time goes to infinity. Explicit bounds for the size of $\mathcal{N}$ are quite easily derived, cf. e.g. (Hafstein, 2004). In this paper we will derive such an estimate, but for the considerably more demanding case of stochastic differential equations.

Notation: We denote by $\|\mathbf{x}\|$ the Euclidian norm of a vector $\mathbf{x} \in \mathbb{R}^d$ and for $A \in \mathbb{R}^{d\times d}$ by $\|A\| = \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|$ the matrix norm induced by the Euclidian vector norm. Vectors are assumed to be column vectors.We denote by $\kappa(A) := \|A\|\|A^{-1}\|$ the condition number with respect to the $\|\cdot\|$ norm of the nonsingular matrix $A \in \mathbb{R}^{d\times d}$. For a symmetric and positive definite $Q \in \mathbb{R}^{d\times d}$ we define the energetic norm $\|\mathbf{x}\|_Q := \sqrt{\mathbf{x}^\top Q\mathbf{x}}$ and the corresponding induced matrix norm $\|A\|_Q := \max_{\|\mathbf{x}\|_Q=1} \|A\mathbf{x}\|_Q$. Recall that a symmetric and positive definite $Q \in \mathbb{R}^{d\times d}$ can be factorized as $Q = ODO^\top$ where $O \in \mathbb{R}^{d\times d}$ is orthogonal, i.e. $O^\top O = O^\top O = I$ and $D = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d) \in \mathbb{R}^{d\times d}$ is a diagonal matrix with $0 < \lambda_1 \le \lambda_2 \le \ldots \le \lambda_d$. For every $a \in \mathbb{R}$ we define the matrix $Q^a = O\operatorname{diag}(\lambda_1^a, \lambda_2^a, \ldots, \lambda_d^a)O^\top$. It is not difficult to see that for $a > 0$ we have $\|Q^a\| = \lambda_d^a$ and $\|Q^{-a}\| = \lambda_1^{-a}$. Further,

$$\begin{aligned}\|Q^{-\frac{1}{2}}\|^{-1}\|\mathbf{x}\| \le \|\mathbf{x}\|_Q &= \sqrt{\mathbf{x}^\top Q\mathbf{x}} \\ &= \|Q^{\frac{1}{2}}\mathbf{x}\| \le \|Q^{\frac{1}{2}}\|\|\mathbf{x}\|.\end{aligned}$$

We consider d-dimensional systems and in all sums where the upper and lower bounds of the sum are omitted they are assumed to be 1 and d respectively, i.e. $\sum_i := \sum_{i=1}^d$, $\sum_{i,j} := \sum_{i,j=1}^d$ etc.

A function $\alpha : \mathbb{R}_+ \to \mathbb{R}_+$ is said to be of class $\mathcal{K}_\infty$

Björnsson, H., Giesl, P., Gudmundsson, S. and Hafstein, S.
Local Lyapunov Functions for Nonlinear Stochastic Differential Equations by Linearization.
DOI: 10.5220/0006944505790586
In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2018) - Volume 1*, pages 579-586
ISBN: 978-989-758-321-6

if it is continuous, monotonically increasing, $\alpha(0) = 0$, and $\lim_{x\to\infty}\alpha(x) = \infty$.

We write $\mathbb{P}$ and $\mathbb{E}$ for probability and expectation respectively. The underlying probability spaces should always be clear from the context. The abbreviation *a.s.* stands for *almost surely*, i.e. with probability one, and $\overset{\text{a.s.}}{=}$ means equal a.s.

2 THE PROBLEM SETTING

We give a short discussion of the setup and the problem at hand. For a more detailed discussion of the setup see (Gudmundsson and Hafstein, 2018, §2). The general d-dimensional stochastic differential equation (SDE) of Itô type we consider is of the form:

$$\mathrm{d}\mathbf{X}(t) = \mathbf{f}(\mathbf{X}(t))\mathrm{d}t + \mathbf{g}(\mathbf{X}(t))\cdot \mathrm{d}\mathbf{W}(t) \tag{1}$$

or equivalently

$$\mathrm{d}\mathbf{X}_i(t) = \mathbf{f}_i(\mathbf{X}(t))\mathrm{d}t + \sum_{u=1}^{U}\mathbf{g}^u(\mathbf{X}(t))\cdot \mathrm{d}W_u(t)$$

for $i = 1,2,\dots,d$. Thus $\mathbf{f} = (f_1, f_2,\dots,f_d)^\top$, $\mathbf{g} = (\mathbf{g}^1,\mathbf{g}^2,\dots,\mathbf{g}^U)$, and $\mathbf{g}^u = (g_1^u, g_2^u,\dots,g_d^u)^\top$, where $f_i, g_i^u : \mathbb{R}^d \to \mathbb{R}$. We assume that the origin is an equilibrium of the system, i.e. $\mathbf{f}(\mathbf{0}) = \mathbf{0}$ and $\mathbf{g}^\mathbf{u}(\mathbf{0}) = \mathbf{0}$ for $u = 1,2,\dots,U$ and we consider *strong solutions* to (1). For deterministic initial value solutions, i.e. $\mathbf{X}(0) = \mathbf{x} \in \mathbb{R}^d$ a.s., we write $\mathbf{X}^\mathbf{x}$ for the solution, i.e.

$$\mathbf{X}^\mathbf{x}(t) = \mathbf{x} + \int_0^t \mathbf{f}(\mathbf{X}(s))\mathrm{d}s + \int_0^t \mathbf{g}(\mathbf{X}(s))\mathrm{d}\mathbf{W}(s),$$

where the second integral is interpreted in the Itô sense. As shown in (Mao, 2008) it suffices to consider deterministic initial value solutions when studying the stability of an equilibrium.

Numerous concepts are in use concerning the stability of equilibria of SDEs. Here we will be concerned with the so-called *asymptotic stability in probability* of the zero solution (Khasminskii, 2012, (5.15)), also referred to as *stochastic asymptotic stability* (Mao, 2008, Definition 4.2.1). For a more detailed discussion of the stability of SDEs see the books by Khasminskii (Khasminskii, 2012) or Mao (Mao, 2008). We recall a few definitions:

Definition 2.1 (Stability in Probability (*SiP*)). *The null solution* $\mathbf{X}(t) \overset{a.s.}{=} \mathbf{0}$ *to the SDE* (1) *is said to be* stable in probability *(SiP) if for every* $r > 0$ *and* $0 < \varepsilon < 1$ *there exists a* $\delta > 0$ *such that*:

$$\|\mathbf{x}\| \le \delta \textit{ implies } \mathbb{P}\left\{\sup_{t\ge 0}\|\mathbf{X}^\mathbf{x}(t)\| \le r\right\} \ge 1-\varepsilon. \quad \square$$

Definition 2.2 (Asymptotic Stability in Probability (*ASiP*)). *The null solution* $\mathbf{X}(t) \overset{a.s.}{=} \mathbf{0}$ *to the SDE* (1) *is said to be* asymptotically stable in probability *(ASiP) if it is SiP and in addition for every* $0 < \varepsilon < 1$ *there exists a* $\delta > 0$ *such that*:

$$\|\mathbf{x}\| \le \delta \textit{ implies } \mathbb{P}\left\{\lim_{t\to\infty}\|\mathbf{X}^\mathbf{x}(t)\| = 0\right\} \ge 1-\varepsilon.$$

□

Our definitions of SiP and ASiP are equivalent to the more common

$$\lim_{\|\mathbf{x}\|\to 0}\mathbb{P}\left\{\sup_{t>0}\|\mathbf{X}^\mathbf{x}(t)\| \le r\right\} = 1 \quad \text{for all } r > 0$$

for SiP and additionally

$$\lim_{\|\mathbf{x}\|\to 0}\mathbb{P}\left\{\limsup_{t\to\infty}\|\mathbf{X}^\mathbf{x}(t)\| = 0\right\} = 1$$

for ASiP, which can be seen by fixing $r > 0$ and writing down the definition of a limit: *for every* $\varepsilon > 0$ *there exists a* $\delta > 0$.

The reason for our formulation is that we want to look at a more practical concept related to such stability, namely a stochastic analog of the basin of attraction (BOA) in the stability theory for deterministic systems, cf. (Gudmundsson and Hafstein, 2018). Instead of the limit $\|\mathbf{x}\| \to 0$ we consider: Given some *confidence* $0 < \gamma \le 1$ how far from the origin can sample paths start and still approach the equilibrium as $t \to \infty$ with probability greater than or equal to γ. This is the motivation for the next definition.

Definition 2.3 (γ-Basin Of Attraction (γ-BOA)). *Consider the system* (1) *and let* $0 < \gamma \le 1$. *We refer to the set*

$$\left\{\mathbf{x} \in \mathbb{R}^d : \mathbb{P}\left\{\lim_{t\to\infty}\|\mathbf{X}^\mathbf{x}(t)\| = 0\right\} \ge \gamma\right\} \quad (\gamma\text{-}BOA)$$

as the γ-basin of attraction*, or short* γ*-BOA, of the equilibrium at the origin.*

□

Note that a 1-BOA corresponds to the usual BOA for deterministic systems.

For the SDE (1) the associated *generator* is given by

$$LV(\mathbf{x}) := \tag{2}$$
$$\nabla V(\mathbf{x})\bullet\mathbf{f}(\mathbf{x}) + \frac{1}{2}\sum_{i,j}\left[\mathbf{g}(\mathbf{x})\mathbf{g}(\mathbf{x})^\top\right]_{ij}\frac{\partial^2 V}{\partial x_i \partial x_j}(\mathbf{x})$$

for some appropriately differentiable $V : \mathcal{U} \to \mathbb{R}$ with $\mathcal{U} \subset \mathbb{R}^d$. Notice that this is just the drift term in the expression for the stochastic differential of the process $t \mapsto V(\mathbf{X}(t))$. The generator for a stochastic system corresponds to the orbital derivative of a deterministic system.

Definition 2.4 (Local Lyapunov function)**.** *Consider the system* (1)*. A function $V \in C(\mathcal{U}) \cap C^2(\mathcal{U} \setminus \{\mathbf{0}\})$, where $\mathbf{0} \in \mathcal{U} \subset \mathbb{R}^d$ is a domain, is called a (local) Lyapunov function for the the system* (1) *if there are functions $\mu_1, \mu_2, \mu_3 \in \mathcal{K}_\infty$, such that V fulfills the properties:*

(i) $\mu_1(\|\mathbf{x}\|) \leq V(\mathbf{x}) \leq \mu_2(\|\mathbf{x}\|)$ *for all* $\mathbf{x} \in \mathcal{U}$

(ii) $LV(\mathbf{x}) \leq -\mu_3(\|\mathbf{x}\|)$ *for all* $\mathbf{x} \in \mathcal{U} \setminus \{\mathbf{0}\}$

Remark 2.5. *It is of vital importance that V is not necessarily differentiable at the equilibrium, because otherwise a large number of systems with an ASiP null solution do not possess a Lyapunov function, cf. (Khasminskii, 2012, Remark 5.5).*

The following theorem provides the first centerpiece of Lyapunov stability theory for our application, cf. (Khasminskii, 2012, Theorem 5.5 and Corollary 5.1):

Theorem 2.6. *If there exists a local Lyapunov functions as in Definition 2.4 for the system* (1)*, then the null solution is ASiP. Further, let $V_{\max} > 0$ and assume that $V^{-1}([0, V_{\max}])$ is a compact subset of $\mathcal{U}$. Then, for every $0 < \beta < 1$ the set $V^{-1}([0, \beta V_{\max}])$ is a subset of the $(1 - \beta)$-BOA of the origin.*

This concludes our discussion of the setup. In the next section we discuss Lyapunov functions for the linearization of (1) and prove the main contribution of this paper, a lower bound on the area where a Lyapunov function for the linearization is also a Lyapunov function for the nonlinear system.

3 MAIN RESULTS

We now consider the linearization of system (1). A Lyapunov function for the linearized system can then be constructed, e.g. with the method form (Hafstein et al., 2018), much more easily than for the nonlinear system (1). In addition to $\mathbf{f}$ and $\mathbf{g}$ satisfying the usual sufficient SDE solution-theory conditions *locally Lipschitz and the linear-growth conditions*, cf. e.g. (Mao, 2008, §2.3) or (Kallenberg, 2002, §21), we assume $\mathbf{f}$ and $\mathbf{g}$ are C^2 on a convex neighbourhood $\mathcal{U} \subset \mathbb{R}^d$ of the origin. The second order Taylor expansion for the components f_i of $\mathbf{f}$ at $\mathbf{x} \in \mathcal{U}$ reads

$$\begin{aligned} f_i(\mathbf{x}) &= \sum_j x_j F_{ij} + \frac{1}{2} \sum_{j,k} x_j x_k R^i_{jk}(\mathbf{x}) \\ &= \big(F\mathbf{x}\big)_i + \frac{1}{2} \mathbf{x}^\top R^i(\mathbf{x})\, \mathbf{x}, \end{aligned}$$

and the components g^u_i of $\mathbf{g}^u$,

$$\begin{aligned} g^u_i(\mathbf{x}) &= \sum_j x_j G^u_{ij} + \frac{1}{2} \sum_{j,k} x_j x_k R^{ui}_{jk}(\mathbf{x}) \\ &= \big(G^u\mathbf{x}\big)_i + \frac{1}{2} \mathbf{x}^\top R^{ui}(\mathbf{x})\, \mathbf{x} \end{aligned}$$

Here

$$F = \big(F_{ij}\big)_{i,j} \in \mathbb{R}^{d \times d} \text{ with } F_{ij} = \partial_j f_i(\mathbf{0})$$

and

$$G^u = \big(G^u_{ij}\big)_{i,j} \in \mathbb{R}^{d \times d} \text{ with } G^u_{ij} = \partial_j g^u_i(\mathbf{0})$$

and the matrices $R^i(\mathbf{x})$ and $R^{ui}(\mathbf{x})$ are the Taylor remainders

$$\begin{aligned} R^i(\mathbf{x}) &= \Big(R^i_{jk}(\mathbf{x})\Big)_{j,k} \in \mathbb{R}^{d \times d} \text{ and} \\ R^{ui}(\mathbf{x}) &= \Big(R^{ui}_{jk}(\mathbf{x})\Big)_{j,k} \in \mathbb{R}^{d \times d}. \end{aligned}$$

By abuse of notation we define the elements of upper bound matrices $R^i = \Big(R^i_{jk}\Big)_{j,k} \in \mathbb{R}^{d \times d}$ and $R^{ui} = \Big(R^{ui}_{jk}\Big)_{j,k} \in \mathbb{R}^{d \times d}$ as follows:

$$\big|\partial^2_{jk} f_i(\mathbf{x})\big| = |R^i_{jk}(\mathbf{x})| \leq R^i_{jk} \text{ and} \tag{3}$$

$$\big|\partial^2_{jk} g^u_i(\mathbf{x})\big| = |R^{ui}_{jk}(\mathbf{x})| \leq R^{ui}_{jk}, \tag{4}$$

for all $\mathbf{x} \in \mathcal{N}$, where $\mathcal{N}$ is a neighbourhood of the origin to be defined later. Finally we fix the constants $\mathcal{R}^i$ and $\mathcal{R}^{ui}$ as

$$\mathcal{R}^i := \|R^i\| \text{ and } \mathcal{R}^{ui} := \|R^{ui}\|. \tag{5}$$

The action of the generator (2) of the system (1) on some $V \in C(\mathcal{U}) \cap C^2(\mathcal{U} \setminus \{\mathbf{0}\})$ can be written as

$$\begin{aligned} LV(\mathbf{x}) &= \frac{1}{2} \sum_{i,j} m_{ij}(\mathbf{x}) \partial^2_{ij} V(\mathbf{x}) + \sum_i f_i(\mathbf{x}) \partial_i V(\mathbf{x}) \\ &= L_0 V(\mathbf{x}) + E(\mathbf{x}) \end{aligned}$$

where $L_0 V(\mathbf{x})$ is the generator of the linearized system defined below and $E(\mathbf{x})$ the rest (containing all the Taylor remainders). We will now work out the details, first notice that:

$$\begin{aligned} m_{ij}(\mathbf{x}) &= \sum_{u=1}^{U} g^u_i(\mathbf{x}) g^u_j(\mathbf{x}) \\ &= \sum_{k,l} x_k x_l \sum_{u=1}^{U} G^u_{ik} G^u_{jl} \\ &\quad + \frac{1}{2} \sum_{k,l,m} x_k x_l x_m \sum_{u=1}^{U} \Big(G^u_{ik} R^{uj}_{lm}(\mathbf{x}) + G^u_{jk} R^{ui}_{lm}(\mathbf{x})\Big) \\ &\quad + \frac{1}{4} \sum_{k,l,m,n} x_k x_l x_m x_n \sum_{u=1}^{U} R^{ui}_{kl}(\mathbf{x}) R^{uj}_{mn}(\mathbf{x}). \end{aligned}$$

We define L_0 as the generator associated to the *linearization* of the system (1), i.e. the system

$$\mathrm{d}\mathbf{X}(t) = F\,X(t)\,\mathrm{d}t + \sum_{u=1}^{U} G^u\,\mathbf{X}(t)\,\mathrm{d}W_u(t) \tag{6}$$

or equivalently

$$\mathrm{d}X_i(t) = \sum_j F_{ij}X_j(t)\,\mathrm{d}t + \sum_{u=1}^{U}\sum_j G^u_{ij}X_j(t)\,\mathrm{d}W_u(t)$$

for $i = 1,2,\dots,d$, which means that

$$L_0V(\mathbf{x}) = \tag{7}$$

$$\sum_{i,j} F_{ij}x_j\partial_i V(\mathbf{x}) + \frac{1}{2}\sum_{i,j}\left(\sum_{k,l}x_kx_l\sum_{u=1}^{U}G^u_{ik}G^u_{jl}\right)\partial^2_{ij}V(\mathbf{x}).$$

We gather together the nonlinear parts of the full SDE generator into the expression for $E(\mathbf{x})$:

$$E(\mathbf{x}) = \underbrace{\sum_s E_s(\mathbf{x})\partial_s V(\mathbf{x})}_{E_F(\mathbf{x})} + \underbrace{\frac{1}{2}\sum_{r,s}E_{rs}(\mathbf{x})\partial^2_{rs}V(\mathbf{x})}_{E_G(\mathbf{x})},$$

where

$$E_s(\mathbf{x}) = \frac{1}{2}\sum_{j,k}x_jx_kR^s_{jk}(\mathbf{x}) \qquad \text{and}$$

$$E_{rs}(\mathbf{x}) = \frac{1}{2}\sum_{k,l,m}x_kx_lx_m\sum_{u=1}^{U}\left(G^u_{rk}R^{us}_{lm}(\mathbf{x}) + G^u_{sk}R^{ur}_{lm}(\mathbf{x})\right)$$
$$+\frac{1}{4}\sum_{k,l,m,n}x_kx_lx_mx_n\sum_{u=1}^{U}R^{ur}_{kl}(\mathbf{x})R^{us}_{mn}(\mathbf{x}).$$

The plan for the rest of this section is as follows: With $LV(\mathbf{x})$ broken up into a linear part $L_0V(\mathbf{x})$ and a nonlinear correction $E(\mathbf{x})$, we take the explicit function

$$V(\mathbf{x}) = \|\mathbf{x}\|^p_Q = \left(\mathbf{x}^\top Q\mathbf{x}\right)^{\frac{p}{2}} \tag{8}$$

as the *ansatz* for the Lyapunov function candidate, where $Q \in \mathbb{R}^{d\times d}$ is a symmetric and positive definite matrix and $p > 0$. As argued in (Hafstein et al., 2018, §4) this is the expected form of a Lyapunov function for the linearized system (6) just as $\mathbf{x} \mapsto \mathbf{x}^\top P\mathbf{x}$ for a symmetric and positive definite P is the usual form for a Lyapunov function for a linear deterministic system $\dot{\mathbf{x}} = A\mathbf{x}$. Note that typically $p < 2$ so V is not differentiable at the origin. For this reason take $\mathbf{x} \neq \mathbf{0}$ in the calculations below. Assuming that we have fixed Q and $p > 0$ such that $L_0V(\mathbf{x}) < 0$ for all $\mathbf{x} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$, we derive a neighbourhood of the origin such that $|L_0V(\mathbf{x})| > |E(\mathbf{x})|$, which implies $LV(\mathbf{x}) < 0$.

From (Hafstein et al., 2018, Lemma 4.1) we can state the following: for $V(\mathbf{x}) = \|\mathbf{x}\|^p_Q$ we have

$$L_0V(\mathbf{x}) = -\frac{1}{2}p\|\mathbf{x}\|^{p-4}_Q H(\mathbf{x}) \;\text{ for all } \mathbf{x} \in \mathbb{R}^d \setminus \{\mathbf{0}\},$$

where

$$H(\mathbf{x}) = -\mathbf{x}^\top\left(F^\top Q + QF + \sum_{u=1}^{U}(G^u)^\top QG^u\right)\mathbf{x}\|\mathbf{x}\|^2_Q$$
$$+(2-p)\sum_{u=1}^{U}\left(\frac{1}{2}\mathbf{x}^\top(QG^u + (G^u)^\top Q)\mathbf{x}\right)^2.$$

This V is a Lyapunov function for the linear system (6) if there is a constant $C > 0$ such that

$$H(\mathbf{x}) \geq C\|\mathbf{x}\|^2_Q\|\mathbf{x}\|^2 \;\text{ for all } \mathbf{x} \in \mathbb{R}^d,$$

because then

$$L_0V(\mathbf{x}) \leq -\frac{1}{2}pC\|\mathbf{x}\|^{p-2}_Q\|\mathbf{x}\|^2 \tag{9}$$

for all $\mathbf{x} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$.

Before we state and prove our results we prove a simple but useful lemma:

Lemma 3.1. *Let $A = (A_{ij}), \widetilde{A} = (\widetilde{A}_{ij}) \in \mathbb{R}^{d\times d}$ be such that $|A_{ij}| \leq \widetilde{A}_{ij}$ for $i,j = 1,2,\dots,d$. Then*

$$\|A\| \leq \|\widetilde{A}\|. \tag{10}$$

In particular

$$\left|\sum_{i,j}x_iA_{ij}y_j\right| \leq \|\widetilde{A}\|\|\mathbf{x}\|\|\mathbf{y}\| \tag{11}$$

and

$$\left|\sum_{i,j,k}x_iQ_{ik}A_{kj}y_j\right| \leq \|\widetilde{A}\|\|Q^{\frac{1}{2}}\|\|\mathbf{x}\|_Q\|\mathbf{y}\| \tag{12}$$
$$\leq \|\widetilde{A}\|\kappa(Q)^{\frac{1}{2}}\|\mathbf{x}\|_Q\|\mathbf{y}\|_Q.$$

for every symmetric and positive definite $Q \in \mathbb{R}^{d\times d}$. If $AQ^{\frac{1}{2}} = Q^{\frac{1}{2}}A$ we even have

$$\left|\sum_{i,j,k}x_iQ_{ik}A_{kj}y_j\right| \leq \|\widetilde{A}\|\|\mathbf{x}\|_Q\|\mathbf{y}\|_Q. \tag{13}$$

Proof. For $\mathbf{x} = (x_1,x_2,\dots,x_d)^\top$ set $\tilde{\mathbf{x}} = (|x_1|,|x_2|,\dots,|x_d|)^\top$. Clearly $\|\mathbf{x}\| = \|\tilde{\mathbf{x}}\|$. The estimate (10) follows from

$$\begin{aligned}\|A\mathbf{x}\|^2 = \mathbf{x}^\top A^\top A\mathbf{x} &= \left|\sum_{i,j,k}x_iA_{ki}A_{kj}x_j\right|\\ &\leq \sum_{i,j,k}|x_i|\cdot|A_{ki}|\cdot|A_{kj}|\cdot|x_j|\\ &\leq \sum_{i,j,k}|x_i|\cdot\widetilde{A}_{ki}\widetilde{A}_{kj}|x_j| = \tilde{\mathbf{x}}^\top\widetilde{A}^\top\widetilde{A}\tilde{\mathbf{x}} = \|\widetilde{A}\tilde{\mathbf{x}}\|^2\\ &\leq \|\widetilde{A}\|^2\|\tilde{\mathbf{x}}\|^2 = \|\widetilde{A}\|^2\|\mathbf{x}\|^2\end{aligned}$$

and thus

$$\|A\| := \sup_{\mathbf{x}\neq\mathbf{0}}\frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} \leq \|\widetilde{A}\|.$$

The inequality (12) follow from

$$
\begin{aligned}
|\sum_{i,j,k} x_i Q_{ik} A_{kj} y_j| &= \left|\sum_{i,j} x_i \left(\sum_k Q_{ik} A_{kj}\right) y_j\right| \\
&= |\mathbf{x}^\top Q A \mathbf{y}| = |(Q^{\frac{1}{2}}\mathbf{x})^\top Q^{\frac{1}{2}} A \mathbf{y}| \\
&\le \|Q^{\frac{1}{2}}\mathbf{x}\| \|Q^{\frac{1}{2}} A\mathbf{y}\| = \|\mathbf{x}\|_Q \|Q^{\frac{1}{2}} A \mathbf{y}\| \\
&\le \|\mathbf{x}\|_Q \|Q^{\frac{1}{2}}\| \|A\| \|\mathbf{y}\| \\
&\le \|\widetilde{A}\| \|Q^{\frac{1}{2}}\| \|Q^{-\frac{1}{2}}\| \|\mathbf{x}\|_Q \|\mathbf{y}\|_Q
\end{aligned}
$$

and (11) follows form (12) with Q as the identity matrix. To see (13) just note that if $AQ^{\frac{1}{2}} = Q^{\frac{1}{2}}A$ we have

$$\|Q^{\frac{1}{2}} A \mathbf{y}\| = \|A Q^{\frac{1}{2}} \mathbf{y}\| \le \|A\| \|Q^{\frac{1}{2}}\mathbf{y}\| \le \|\widetilde{A}\| \|\mathbf{y}\|_Q$$

which can be used to improve the estimate above. □

Remark 3.2. *If A in* (12) *is symmetric we have*

$$\mathbf{x}^\top Q A \mathbf{y} = \sum_{i,j,k} x_i Q_{ik} A_{kj} y_j = \sum_{i,j,k} y_j A_{jk} Q_{ki} x_i = \mathbf{y}^\top A Q \mathbf{x}.$$

Remark 3.3. *For vectors $\mathbf{x}, \widetilde{\mathbf{x}} \in \mathbb{R}^d$, $|x_i| \le \widetilde{x}_i$ for $i = 1, 2, \ldots, d$, we obviously have $\|\mathbf{x}\| \le \|\widetilde{\mathbf{x}}\|$, but in general $\|\mathbf{x}\|_Q$ is not necessarily smaller than $\|\widetilde{\mathbf{x}}\|_Q$. Take for example $\mathbf{x} = (1, -1)^\top$, $\widetilde{\mathbf{x}} = (1, 1)^\top$, and $Q = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$. Then $\|\mathbf{x}\|_Q = \sqrt{\mathbf{x}^\top Q \mathbf{x}} = \sqrt{6}$ but $\|\mathbf{y}\|_Q = \sqrt{2}$. For this reason one cannot expect $|A_{ij}| \le \widetilde{A}_{ij}$ to imply $\|A\|_Q \le \|\widetilde{A}\|_Q$ for matrices $A, \widetilde{A} \in \mathbb{R}^{d \times d}$.*

We now come to the main contribution of this paper:

Theorem 3.4. *Consider the system* (1)*, assume that V as in* (8) *is a Lyapunov function for its linearization* (6)*, and let $C > 0$ be a constant as in* (9)*. Let $\rho^* > 0$ and assume the estimates* (3)*,* (4)*, and* (5) *hold true on $\mathcal{N} = \mathcal{D}^* := \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_Q \le \rho^*\}$. Define*

$$
\begin{aligned}
p^* &:= 1 + |p-2|, \\
\mathcal{R}^i &:= \|R^i\|, \\
\mathcal{R}^{ui} &:= \|R^{ui}\|, \\
\mathcal{R}_F &:= \|(\mathcal{R}^1, \mathcal{R}^2, \ldots, \mathcal{R}^d)\|, \\
\mathcal{R}^u_G &:= \|(\mathcal{R}^{u1}, \mathcal{R}^{u2}, \ldots, \mathcal{R}^{ud})\|, \\
\mathcal{R}_G &:= \|(\mathcal{R}^1_G, \mathcal{R}^2_G, \ldots, \mathcal{R}^U_G)\|^2, \\
\widetilde{E}_G &:= \|Q^{\frac{1}{2}}\| \left(\mathcal{R}_F + p^* \sum_{u=1}^{U} \mathcal{R}^u_G \|Q^{\frac{1}{2}} G^u Q^{-\frac{1}{2}}\|\right), \\
\widetilde{E}^*_G &:= \frac{1}{4} p^* \kappa(Q) \mathcal{R}_G.
\end{aligned}
$$

Then

$$LV(\mathbf{x}) = L_0 V(\mathbf{x}) + E(\mathbf{x})$$

where L_0V is defined in (7) *and*

$$|E(\mathbf{x})| \le \frac{1}{2} p \|\mathbf{x}\|_Q^{p-2} \|\mathbf{x}\|^2 \cdot \|\mathbf{x}\|_Q \left(\widetilde{E}_G + \widetilde{E}^*_G \|\mathbf{x}\|_Q\right)$$

for $\mathbf{x} \in \mathcal{D}^* := \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_Q \le \rho^*\}$. In particular, V is a Lyapunov function for the nonlinear system (1) *satisfying the condition of Definition 2.4 on*

$$\mathcal{U} = \mathcal{D} := \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_Q \le \rho\},$$

with

$$\rho < \min\left\{\rho^*, \frac{1}{2\widetilde{E}^*_G}\left(\sqrt{(\widetilde{E}_G)^2 + 4C\widetilde{E}^*_G} - \widetilde{E}_G\right)\right\}.$$

Proof. Let us first compute $\partial_s V(\mathbf{x})$ and $\partial^2_{rs} V(\mathbf{x})$,

$$
\begin{aligned}
\partial_s V(\mathbf{x}) &= \left(\sum_j Q_{sj} x_j + \sum_i Q_{is} x_i\right) \frac{p}{2} \left(\sum_{i,j} Q_{ij} x_i x_j\right)^{\frac{p}{2}-1} \\
&= p \sum_i x_i Q_{is} \|\mathbf{x}\|_Q^{p-2} \quad \text{and}
\end{aligned}
$$

$$
\begin{aligned}
\partial^2_{rs} V(\mathbf{x}) &= p Q_{rs} \|\mathbf{x}\|_Q^{p-2} + p \left(\sum_j x_j Q_{js}\right) \left(\frac{p}{2} - 1\right) \\
&\quad \times 2 \left(\sum_i x_i Q_{ir}\right) \left(\sum_{i,j} Q_{ij} x_i x_j\right)^{\frac{p}{2}-2} \\
&= p \|\mathbf{x}\|_Q^{p-2} Q_{rs} + p(p-2) \sum_{i,j} x_i x_j Q_{ir} Q_{js} \|\mathbf{x}\|_Q^{p-4} \\
&= p \|\mathbf{x}\|_Q^{p-4} \left(Q_{rs} \|\mathbf{x}\|_Q^2 + (p-2) \sum_{i,j} x_i x_j Q_{ir} Q_{js}\right).
\end{aligned}
$$

Now set $\mathbf{z} = (z_1, z_2, \ldots, z_d)^\top$ with $z_s := \mathbf{x}^\top R^s(\mathbf{x})\mathbf{x}$ and then $|z_s| \le \mathcal{R}^s \|\mathbf{x}\|^2$ and $\|\mathbf{z}\| \le \|\mathbf{x}\|^2 \mathcal{R}_F$ for $\mathbf{x} \in \mathcal{D}^*$. Then

$$
\begin{aligned}
|E_F(\mathbf{x})| &\le \left|\sum_s E_s(\mathbf{x}) \partial_s V(\mathbf{x})\right| \\
&\le \frac{p}{2} \|\mathbf{x}\|_Q^{p-2} \left|\sum_{s,i,j,k} x_j x_k R^s_{jk}(\mathbf{x}) x_i Q_{is}\right| \\
&= \frac{p}{2} \|\mathbf{x}\|_Q^{p-2} \left|\sum_{s,i} x_i Q_{is} \left(\sum_{j,k} x_j R^s_{jk}(\mathbf{x}) x_k\right)\right| \\
&= \frac{p}{2} \|\mathbf{x}\|_Q^{p-2} \left|\sum_{s,i} x_i Q_{is} \left(\mathbf{x}^\top R^s(\mathbf{x}) \mathbf{x}\right)\right| \\
&= \frac{p}{2} \|\mathbf{x}\|_Q^{p-2} \left|\sum_{s,i} x_i Q_{is} z_s\right| \\
&= \frac{p}{2} \|\mathbf{x}\|_Q^{p-2} \left|\mathbf{x}^\top Q \mathbf{z}\right| \\
&= \frac{p}{2} \|\mathbf{x}\|_Q^{p-2} \|Q^{\frac{1}{2}}\mathbf{x}\| \|Q^{\frac{1}{2}}\mathbf{z}\| \\
&\le \frac{p}{2} \|\mathbf{x}\|_Q^{p-2} \|\mathbf{x}\|_Q \|Q^{\frac{1}{2}}\| \|\mathbf{z}\| \\
&\le \frac{p}{2} \|\mathbf{x}\|_Q^{p-1} \|\mathbf{x}\|^2 \|Q^{\frac{1}{2}}\| \mathcal{R}_F.
\end{aligned}
$$

Since $E_G(\mathbf{x}) = \frac{1}{2}\sum_{r,s} E_{rs}(\mathbf{x})\partial^2_{rs}V(\mathbf{x})$ and by using our expressions for E_{rs} and $\partial^2_{rs}V(\mathbf{x})$ we obtain:

$$
\begin{aligned}
|E_G(\mathbf{x})| \le \frac{1}{4}p\|\mathbf{x}\|_Q^{p-4} & \Bigg| \sum_{r,s,k,l,m} x_k x_l x_m \\
&\times \sum_{u=1}^{U} \left(G^u_{rk}R^{us}_{lm}(\mathbf{x}) + G^u_{sk}R^{ur}_{lm}(\mathbf{x})\right) \\
&\times \left(Q_{rs}\|\mathbf{x}\|_Q^2 + (p-2)\sum_{i,j} x_i x_j Q_{ir} Q_{js}\right) \Bigg| \\
+ \frac{1}{8}p\|\mathbf{x}\|_Q^{p-4} & \Bigg| \sum_{r,s,k,l,m,n} x_k x_l x_m x_n \\
&\times \left(Q_{rs}\|\mathbf{x}\|_Q^2 + (p-2)\sum_{i,j} x_i x_j Q_{ir} Q_{js}\right) \\
&\times \sum_{u=1}^{U} R^{ur}_{kl}(\mathbf{x}) R^{us}_{mn}(\mathbf{x}) \Bigg|.
\end{aligned}
$$

We now estimate the expression on the right-hand side term by term: Set $\mathbf{z}^u = (z^u_1, z^u_2, \ldots, z^u_d)^\top$, where $z^u_i := \mathbf{x}^\top R^{ui}(\mathbf{x})\mathbf{x}$, and then $|z^u_i| \le \mathcal{R}^{ui}\|\mathbf{x}\|^2$ and $\|\mathbf{z}^u\| \le \|\mathbf{x}\|^2\mathcal{R}^u_G$ for $\mathbf{x} \in \mathcal{D}^*$. Then

$$
\begin{aligned}
&\sum_{r,s,k,l,m} x_k x_l x_m \sum_{u=1}^{U} G^u_{rk} R^{us}_{lm}(\mathbf{x}) Q_{rs} \|\mathbf{x}\|_Q^2 \\
&= \|\mathbf{x}\|_Q^2 \sum_{u=1}^{U} \sum_{r,s,k} x_k \left(\sum_{l,m} x_l R^{us}_{lm} x_m\right) Q_{sr} G^u_{rk} \\
&= \|\mathbf{x}\|_Q^2 \sum_{u=1}^{U} \sum_{r,s,k} x_k \left(\mathbf{x}^\top R^{us}\mathbf{x}\right) Q_{sr} G^u_{rk} \\
&= \|\mathbf{x}\|_Q^2 \sum_{u=1}^{U} \sum_{r,s,k} z^u_s Q_{sr} G^u_{rk} x_k \\
&= \|\mathbf{x}\|_Q^2 \sum_{u=1}^{U} (\mathbf{z}^u)^\top Q G^u \mathbf{x} \\
&= \|\mathbf{x}\|_Q^2 \sum_{u=1}^{U} (\mathbf{z}^u)^\top Q G^u Q^{-\frac{1}{2}} Q^{\frac{1}{2}} \mathbf{x} \\
&\le \|\mathbf{x}\|_Q^3 \|\mathbf{x}\|^2 \sum_{u=1}^{U} \|Q G^u Q^{-\frac{1}{2}}\| \mathcal{R}^u_G \\
&\le \|\mathbf{x}\|_Q^3 \|\mathbf{x}\|^2 \|Q^{\frac{1}{2}}\| \sum_{u=1}^{U} \|Q^{\frac{1}{2}} G^u Q^{-\frac{1}{2}}\| \mathcal{R}^u_G
\end{aligned}
$$

and similarly

$$
\begin{aligned}
&\sum_{r,s,k,l,m} x_k x_l x_m \sum_{u=1}^{U} G^u_{sk} R^{ur}_{lm}(\mathbf{x}) Q_{rs} \|\mathbf{x}\|_Q^2 \\
&= \|\mathbf{x}\|_Q^2 \sum_{u=1}^{U} \sum_{r,s,k} \left(\mathbf{x}^\top R^{ur}\mathbf{x}\right) Q_{rs} G^u_{sk} x_k \\
&= \|\mathbf{x}\|_Q^2 \sum_{u=1}^{U} \sum_{r,s,k} z^u_r Q_{rs} G^u_{sk} x_k \\
&= \|\mathbf{x}\|_Q^2 \sum_{u=1}^{U} (\mathbf{z}^u)^\top Q G^u \mathbf{x} \\
&\le \|\mathbf{x}\|_Q^3 \|\mathbf{x}\|^2 \sum_{u=1}^{U} \|Q G^u Q^{-\frac{1}{2}}\| \mathcal{R}^u_G \\
&\le \|\mathbf{x}\|_Q^3 \|\mathbf{x}\|^2 \|Q^{\frac{1}{2}}\| \sum_{u=1}^{U} \|Q^{\frac{1}{2}} G^u Q^{-\frac{1}{2}}\| \mathcal{R}^u_G.
\end{aligned}
$$

Further

$$
\begin{aligned}
&\sum_{r,s,k,l,m} x_k x_l x_m \sum_{u=1}^{U} G^u_{rk} R^{us}_{lm}(\mathbf{x}) (p-2) \sum_{i,j} x_i x_j Q_{ir} Q_{js} \\
&= (p-2) \sum_{u=1}^{U} \sum_{j,s} \left(\sum_{i,k,r} x_i Q_{ir} G^u_{rk} x_k\right) \\
&\quad \times \left(\sum_{l,m} x_l R^{us}_{lm}(\mathbf{x}) x_m\right) Q_{sj} x_j \\
&= (p-2) \sum_{u=1}^{U} \sum_{j,s} \left(\mathbf{x}^\top Q G^u \mathbf{x}\right) \left(\mathbf{x}^\top R^{us}\mathbf{x}\right) Q_{sj} x_j \\
&= (p-2) \sum_{u=1}^{U} \sum_{j,s} \left(\mathbf{x}^\top Q G^u Q^{-\frac{1}{2}} Q^{\frac{1}{2}} \mathbf{x}\right) \left(\mathbf{x}^\top R^{us}\mathbf{x}\right) Q_{sj} x_j \\
&\le |p-2| \sum_{u=1}^{U} \|\mathbf{x}\|_Q^2 \|Q^{\frac{1}{2}} G^u Q^{-\frac{1}{2}}\| \left|\sum_{j,s} z^u_s Q_{sj} x_j\right| \\
&\le |p-2| \|\mathbf{x}\|_Q^2 \sum_{u=1}^{U} \|Q^{\frac{1}{2}} G^u Q^{-\frac{1}{2}}\| \left|(\mathbf{z}^u)^\top Q \mathbf{x}\right| \\
&\le |p-2| \|\mathbf{x}\|_Q^2 \sum_{u=1}^{U} \|Q^{\frac{1}{2}} G^u Q^{-\frac{1}{2}}\| \|\mathbf{z}^u\| \|Q^{\frac{1}{2}}\| \|\mathbf{x}\|_Q \\
&\le |p-2| \|\mathbf{x}\|_Q^3 \|\mathbf{x}\|^2 \|Q^{\frac{1}{2}}\| \sum_{u=1}^{U} \|Q^{\frac{1}{2}} G^u Q^{-\frac{1}{2}}\| \mathcal{R}^u_G
\end{aligned}
$$

and similarly

$$
\begin{aligned}
&\sum_{r,s,k,l,m} x_k x_l x_m \sum_{u=1}^{U} G^u_{sk} R^{ur}_{lm}(\mathbf{x})(p-2)\sum_{i,j} x_i x_j Q_{ir} Q_{js} \\
&= (p-2)\sum_{u=1}^{U}\sum_{i,r}\left(\sum_{j,k,s} x_i Q_{js} G^u_{sk} x_k\right) \\
&\quad \times \left(\sum_{l,m} x_l R^{ur}_{lm}(\mathbf{x}) x_m\right) Q_{ri} x_i \\
&= (p-2)\sum_{u=1}^{U}\sum_{i,r}\left(\mathbf{x}^\top Q G^u \mathbf{x}\right)\left(\mathbf{x}^\top R^{ur}\mathbf{x}\right) Q_{ri} x_i \\
&\le |p-2| \sum_{u=1}^{U} \|\mathbf{x}\|_Q^2 \|Q^{\frac{1}{2}} G^u Q^{-\frac{1}{2}}\| \left|\sum_{i,r} z^u_r Q_{ri} x_i\right| \\
&\le |p-2| \|\mathbf{x}\|_Q^3 \|\mathbf{x}\|^2 \|Q^{\frac{1}{2}}\| \sum_{u=1}^{U} \|Q^{\frac{1}{2}} G^u Q^{-\frac{1}{2}}\| \mathcal{R}^u_G.
\end{aligned}
$$

Further

$$
\begin{aligned}
&\sum_{r,s,k,l,m,n} x_k x_l x_m x_n Q_{rs} \|\mathbf{x}\|_Q^2 \sum_{u=1}^{U} R^{ur}_{kl}(\mathbf{x}) R^{us}_{mn}(\mathbf{x}) \\
&= \|\mathbf{x}\|_Q^2 \sum_{u=1}^{U}\sum_{r,s}\left(\sum_{k,l} x_k R^{ur}_{kl}(\mathbf{x}) x_l\right) Q_{rs} \\
&\quad \times \left(\sum_{m,n} x_m R^{us}_{mn}(\mathbf{x}) x_n\right) \\
&= \|\mathbf{x}\|_Q^2 \sum_{u=1}^{U}\sum_{r,s} z^u_r Q_{rs} z^u_s \\
&= \|\mathbf{x}\|_Q^2 \sum_{u=1}^{U}\left((\mathbf{z}^u)^\top Q \mathbf{z}^u\right) \\
&\le \|\mathbf{x}\|_Q^2 \sum_{u=1}^{U} \|Q\| \|\mathbf{z}^u\|^2 \\
&\le \|\mathbf{x}\|_Q^2 \|\mathbf{x}\|^4 \|Q\| \sum_{u=1}^{U} (\mathcal{R}^u_G)^2 \\
&= \|\mathbf{x}\|_Q^2 \|\mathbf{x}\|^4 \|Q\| \mathcal{R}_G \\
&\le \|\mathbf{x}\|_Q^4 \|\mathbf{x}\|^2 \|Q^{-1}\| \|Q\| \mathcal{R}_G \\
&= \|\mathbf{x}\|_Q^4 \|\mathbf{x}\|^2 \kappa(Q) \mathcal{R}_G.
\end{aligned}
$$

Finally

$$
\begin{aligned}
&\sum_{r,s,k,l,m,n} x_k x_l x_m x_n (p-2)\sum_{i,j} x_i x_j Q_{ir} Q_{js} \sum_{u=1}^{U} R^{ur}_{kl}(\mathbf{x}) R^{us}_{mn}(\mathbf{x}) \\
&= (p-2)\sum_{u=1}^{U}\sum_{i,j,r,s} x_i Q_{ir}\left(\sum_{k,l} x_k R^{ur}_{kl}(\mathbf{x}) x_l\right) x_j Q_{js} \\
&\qquad \times \left(\sum_{m,n} x_m R^{us}_{mn}(\mathbf{x}) x_n\right) \\
&= (p-2)\sum_{u=1}^{U}\sum_{i,j,r,s} x_i Q_{ir} z^u_r x_j Q_{js} z^u_s \\
&= (p-2)\sum_{u=1}^{U}\left(\sum_{i,r} x_i Q_{ir} z^u_r\right)\left(\sum_{i,r} x_j Q_{js} z^u_s\right) \\
&= (p-2)\sum_{u=1}^{U}\left(\mathbf{x}^\top Q \mathbf{z}^u\right)^2 \\
&\le |p-2| \sum_{u=1}^{U} \|\mathbf{x}\|_Q^2 \|Q^{\frac{1}{2}}\|^2 \|\mathbf{z}^u\|^2 \\
&\le |p-2| \|\mathbf{x}\|_Q^2 \|\mathbf{x}\|^4 \|Q\| \mathcal{R}_G \\
&\le |p-2| \|\mathbf{x}\|_Q^4 \|\mathbf{x}\|^2 \|Q^{-1}\| \|Q\| \mathcal{R}_G \\
&= |p-2| \|\mathbf{x}\|_Q^4 \|\mathbf{x}\|^2 \kappa(Q) \mathcal{R}_G.
\end{aligned}
$$

By combining the results from these estimates we get

$$
\begin{aligned}
&|E_G(\mathbf{x})| \le \\
&\frac{1}{4} p \|\mathbf{x}\|_Q^{p-4} \Bigg| 2\|\mathbf{x}\|_Q^3 \|\mathbf{x}\|^2 \|Q^{\frac{1}{2}}\| \sum_{u=1}^{U} \|Q^{\frac{1}{2}} G^u Q^{-\frac{1}{2}}\| \mathcal{R}^u_G \\
&\qquad + 2|p-2| \|\mathbf{x}\|_Q^3 \|\mathbf{x}\|^2 \|Q^{\frac{1}{2}}\| \sum_{u=1}^{U} \|Q^{\frac{1}{2}} G^u Q^{-\frac{1}{2}}\| \mathcal{R}^u_G \Bigg| \\
&\quad + \frac{1}{8} p \|\mathbf{x}\|_Q^{p-4} \Bigg| \|\mathbf{x}\|_Q^4 \|\mathbf{x}\|^2 \kappa(Q) \mathcal{R}_G \\
&\qquad + |p-2| \|\mathbf{x}\|_Q^4 \|\mathbf{x}\|^2 \kappa(Q) \mathcal{R}_G \Bigg| \\
&= \frac{1}{2} p \|\mathbf{x}\|_Q^{p-1} \|\mathbf{x}\|^2 (1+|p-2|) \\
&\times \left(\|Q^{\frac{1}{2}}\| \sum_{u=1}^{U} \mathcal{R}^u_G \|Q^{\frac{1}{2}} G^u Q^{-\frac{1}{2}}\| + \frac{1}{4}\kappa(Q)\mathcal{R}_G \|\mathbf{x}\|_Q \right)
\end{aligned}
$$

and we can estimate

$$
\begin{aligned}
|E(\mathbf{x})| &\le |E_F(\mathbf{x})| + |E_G(\mathbf{x})| \\
&\le \frac{1}{2} p \|\mathbf{x}\|_Q^{p-2} \|\mathbf{x}\|^2 \cdot \|\mathbf{x}\|_Q \left(\widetilde{E}_G + \widetilde{E}^*_G \|\mathbf{x}\|_Q\right),
\end{aligned}
$$

which proves the first stated inequality.

Since

$$\begin{aligned} LV(\mathbf{x}) &= L_0V(\mathbf{x}) + E(\mathbf{x}) \\ &\leq -\frac{1}{2}pC\|\mathbf{x}\|_Q^{p-2}\|\mathbf{x}\|^2 + E(\mathbf{x}) \\ &\leq -\frac{1}{2}p\|\mathbf{x}\|_Q^{p-2}\|\mathbf{x}\|^2\left[C - \|\mathbf{x}\|_Q\left(\widetilde{E}_G + \widetilde{E}_G^*\|\mathbf{x}\|_Q\right)\right] \end{aligned}$$

we have $LV(\mathbf{x}) < 0$ if

$$\|\mathbf{x}\|_Q\left(\widetilde{E}_G + \widetilde{E}_G^*\|\mathbf{x}\|_Q\right) < C,$$

i.e.

$$\|\mathbf{x}\|_Q < \frac{-\widetilde{E}_G + \sqrt{(\widetilde{E}_G)^2 + 4C\widetilde{E}_G^*}}{2\widetilde{E}_G^*}.$$

Thus for

$$\mathbf{x} \in \mathcal{D} = \{\mathbf{x} \in \mathbb{R}^d \,:\, \|\mathbf{x}\|_Q \leq \rho\}$$

with

$$\rho < \min\left\{\rho^*, \frac{1}{2\widetilde{E}_G^*}\left(\sqrt{(\widetilde{E}_G)^2 + 4C\widetilde{E}_G^*} - \widetilde{E}_G\right)\right\},$$

we have $LV(\mathbf{x}) < 0$, which concludes the proof. □

4 CONCLUSIONS

We derived rigid bounds on a domain, on which a Lyapunov function for a linearized stochastic differential equation is also a Lyapunov function for the original nonlinear system. This allows for the derivation of a lower bound on the equilibrium's γ-basin of attraction, i.e. the area in which all started solutions converge to the equilibrium with probability no less than γ. Another application is the facilitation of a numerical method to compute Lyapunov functions for nonlinear stochastic differential equations on a larger domain as discussed in (Gudmundsson and Hafstein, 2018), because one first needs a local Lyapunov function at the equilibrium.

ACKNOWLEDGEMENT

The research done for this paper was supported by the Icelandic Research Fund (Rannís) in the project 'Lyapunov Methods and Stochastic Stability' (152429-051), which is gratefully acknowledged.

REFERENCES

Gudmundsson, S. and Hafstein, S. (2018). Probabilistic basin of attraction and its estimation using two Lyapunov functions. *Complexity*, Article ID:2895658.

Hafstein, S. (2004). A constructive converse Lyapunov theorem on exponential stability. *Discrete Contin. Dyn. Syst. Ser. A*, 10(3):657–678.

Hafstein, S., Gudmundsson, S., Giesl, P., and Scalas, E. (2018). Lyapunov function computation for autonomous linear stochastic differential equations using sum-of-squares programming. *Discrete Contin. Dyn. Syst. Ser. B*, 2(23):939–956.

Kallenberg, O. (2002). *Foundations of Modern Probability*. Springer, 2 edition.

Khasminskii, R. (2012). *Stochastic stability of differential equations*. Springer, 2nd edition.

Mao, X. (2008). *Stochastic Differential Equations and Applications*. Woodhead Publishing, 2nd edition.

Paper IV

Advanced algorithm for interpolation with Wendland functions

Bjornsson, Hjortur and Hafstein, Sigurdur

Informatics in Control, Automation and Robotics, Lecture Notes in Electrical Engineering 720

Advanced algorithm for interpolation with Wendland functions*

Hjortur Bjornsson and Sigurdur Hafstein[0000−0003−0073−2765]

Science Institute, University of Iceland, Dunhagi 3, 107 Reykjavík, Iceland
hjb6@hi.is, shafstein@hi.is

Abstract. We develop and study algorithms for computing Lyapunov functions using meshless collocation and Wendland functions. We present a software tool that generates a C/C++ library that implements Wendland functions of arbitrary order in a specified factorized form with advantageous numerical properties. Additionally, we describe the algorithm used by the tool to generate these Wendland functions. Our factorized form is more efficient and has higher numerical accuracy than previous implementations. We develop and implement optimal grid generation for the interpolation problem using the Wendland functions. Finally, we present software that calculates Lyapunov functions using these Wendland functions and the optimally generated grid. The software tool and library are available for download with examples of usage.

Keywords: Wendland function, Lyapunov functions, radial basis functions, code generation.

1 Introduction

Interpolation and collocation using Radial Basis Functions (RBF), in particular compactly supported RBFs, have been the subject of numerous research activities in the past decades [27, 9, 10, 24, 7, 6, 25, 26]. They are well suited as kernels of Reproducing Kernel Hilbert Spaces and their mathematical theory is mature. The authors and their collaborators have applied Wendland's compactly supported RBFs for computing Lyapunov functions for nonlinear systems, both deterministic [11, 12, 14] and stochastic [5], where Lyapunov functions are a useful tool to analyse stability of these systems, cf. e.g. [18, 22, 23, 19, 20]. Various numerical methods have been used to find Lyapunov functions for the systems at hand [14, 16]. Meshless collocation using RBFs is one such method and many different families of RBFs have been studied [25].

In the papers [12–14, 5] and the book [11] meshless collocation is used with Wendland functions, where the Wendland function family is defined in a recursive way and in order to determine the actual functions to use in a software implementation many calculations had to be done by hand. In [2] an algorithm

* This research was supported by the Icelandic Research Fund (Rannis), grant number 152429-051, Lyapunov Methods and Stochastic Stability.

is proposed that determines the Wendland polynomials in expanded form, that is: for each pair of integers $l, k \geq 0$, it finds a list of numbers $a_0, a_1, \dots a_d$ such that the Wendland function $\psi_{l,k}(r) = \sum_{i=0}^{d} a_i r^i$. However, it was shown in [3] that the evaluations of these polynomials in this form using typical schemes, such as Horner's scheme, can lead to significant numerical errors.

Having the Wendland functions in factorized form [3] is more efficient and numerically accurate, so we propose an alternate method to determine the functions in factorized form. For that purpose, we have created a software tool that generates a reusable software library in C/C++, which implements these Wendland polynomials in factorized form. A first version of this software library was presented in [4]. We have now extended it considerably and added more functionality, most notably efficient grid generation and algorithms to solve interpolation problems for generating Lyapunov functions for stochastic and deterministic dynamical systems.

2 Background

Meshless collocation with RBFs is a method that can be used to calculate Lyapunov functions for either stochastic or deterministic dynamical systems. In paper [5] meshless collocation was used to calculate Lyapunov functions for Stochastic Differential Equations (SDE); see e.g. [11, 14] for a similar approach for deterministic systems.

The method revolves around solving a linear Partial Differential Equation (PDE). Let $\Omega \subset \mathbb{R}^n$ be a given domain and $\Gamma \subset \mathbb{R}^n$ its boundary. Then we want to solve the (PDE)

$$\begin{cases} LV(\mathbf{x}) = h(\mathbf{x}) & \mathbf{x} \in \Omega \\ V(\mathbf{x}) = c(\mathbf{x}) & \mathbf{x} \in \Gamma, \end{cases}$$

where L is a certain differential operator, and h and c are some appropriately chosen functions.

Using meshless collocation to solve the PDE above we choose points $X_1 = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \Omega$ and $X_2 = \{\xi_1, \dots, \xi_M\} \subset \Gamma$, and solve the interpolation problem

$$\begin{cases} LV(\mathbf{x}_i) = h(\mathbf{x}_i) & \text{for all } i = 1, \dots, N \\ V(\xi_i) = c(\xi_i) & \text{for all } i = 1, \dots, M. \end{cases}$$

The solution is then given in terms of a radial basis function ψ,

$$V(\mathbf{x}) = \sum_{k=1}^{N} \alpha_k (\delta_{\mathbf{x}_k} \circ L)^{\mathbf{y}} \psi(\|\mathbf{x} - \mathbf{y}\|) + \sum_{k=1}^{M} \alpha_{N+k} (\delta_{\xi_k} \circ L^0)^{\mathbf{y}} \psi(\|\mathbf{x} - \mathbf{y}\|), \quad (1)$$

where L^0 is the identity operator, $\delta_{\mathbf{y}} V(\cdot) = V(\mathbf{y})$ and superscript $\mathbf{y}$ denotes that the operator is applied with respect to the variable $\mathbf{y}$.

The constants α_i are determined as a solution to the linear system

$$A\alpha = \gamma, \tag{2}$$

where A, called the interpolation matrix, is the symmetric matrix

$$A = \begin{bmatrix} B & C \\ C^T & D \end{bmatrix} \tag{3}$$

and the matrices $B = (b_{jk})_{j,k=1,\ldots,N}$, $C = (c_{jk})_{j=1,\ldots,N,k=1,\ldots,M}$ and $D = (d_{jk})_{j,k=1,\ldots,M}$ have elements:

$$\begin{aligned} b_{jk} &= (\delta_{\mathbf{x}_j} \circ L)^{\mathbf{x}} (\delta_{\mathbf{x}_k} \circ L)^{\mathbf{y}} \psi(\mathbf{x} - \mathbf{y}) \\ c_{jk} &= (\delta_{\mathbf{x}_j} \circ L)^{\mathbf{x}} (\delta_{\xi_k} \circ L^0)^{\mathbf{y}} \psi(\mathbf{x} - \mathbf{y}) \\ d_{jk} &= (\delta_{\xi_j} \circ L^0)^{\mathbf{x}} (\delta_{\xi_k} \circ L^0)^{\mathbf{y}} \psi(\mathbf{x} - \mathbf{y}). \end{aligned}$$

The vector γ has components given by

$$\begin{aligned} \gamma_j &= r(\mathbf{x}_j), \quad 1 \le j \le N \\ \gamma_{j+N} &= c(\xi_j), \quad 1 \le j \le M \end{aligned}$$

There are different choices for the radial basis function ψ. We want the interpolation matrix A to be symmetric and positive definite and choosing ψ to have compact support can make A sparse. Under a few mild conditions the choice of ψ as a Wendland function, i.e. a compactly supported radial basis function, ensures this [26].

This method works the same way for determining Lyapunov functions for both deterministic systems and SDEs, the difference is the choice of the differential operator L. For deterministic systems it is the orbital derivative, a first order differential operator, and for stochastic systems it is a second order differential operator.

To compute such a Lyapunov function a large number of evaluations of the function ψ and its derivatives is necessary, see e.g. the examples given in equations (16) and (19). To verify the properties of a Lyapunov function for the function computed, even more evaluations are necessary. Therefore, it turned out to be essential that these evaluations could be carried out in an efficient and accurate way.

3 Wendland functions

The Wendland functions are compactly supported radial basis functions that are polynomials on their support, which makes computations with them simple. They are a family of functions depending on two parameters $l, k \in \mathbb{N}_0$ defined by the recursive relations:

$$\psi_{l,0}(r) = [(1 - r)_+]^l \tag{4}$$

and

$$\psi_{l,k+1}(r) = C_{l,k+1} \int_r^1 t\psi_{l,k}(t)\mathrm{d}t, \tag{5}$$

where $(1-r)_+ := \max\{1-r, 0\}$ and $C_{l,k+1} \neq 0$ is a constant.

Therefore these functions also satisfy the relation

$$-C_{l,k+1}\psi_{l,k}(r) = \frac{\frac{d}{dr}\psi_{l,k+1}(r)}{r}. \tag{6}$$

For interpolation using a particular Wendland function as the base function, the value of the constant $C_{l,k+1} \neq 0$ is not of importance because the Wendland function appears linearly on both sides of a linear equation. Therefore, one can just fix values that are convenient for the problem at hand and we will do this in the following section. However, when solving collocation problems, we apply a differential operator, see equations (1) and (3), so we get terms involving both the original Wendland function and its derivatives. The derivative of a Wendland function can be written in terms of a lower order Wendland function, using equation (6), and when doing this it is necessary to keep track of the constants $C_{l,k+1}$ for the derivatives. It is only the constant for the base function that can be chosen arbitrarily. After the choice has been made, we must keep track of it through all calculations.

First, we choose a particular function $\psi_{l,k}$ and by abuse of notation we denote it by $\psi_0 = \psi_{l,k}$. Then we define

$$\psi_i(r) = \frac{\frac{d}{dr}\psi_{i-1}(r)}{r} \quad \text{for } i = 1, 2, \dots, k. \tag{7}$$

The function ψ_i is then a specific Wendland function of order $l, k-i$.
Now the functions

$$\Phi_{l,0}(r) = [(1-r)_+]^l \quad \text{and} \tag{8}$$

$$\Phi_{l,k}(r) = \int_r^1 \Phi_{l,0}(t)t(t^2-r^2)^{k-1}\mathrm{d}t \quad \text{for } k > 0 \tag{9}$$

also satisfy a relation of the form

$$-2(k-1)\Phi_{l,k}(r) = \frac{\frac{d}{dr}\Phi_{l,k+1}(r)}{r},$$

for all integers $k, l \geq 0$, i.e. a relation identical to equation (6) with $C_{l,k+1} = 2(k-1)$. Just note that

$$\frac{d}{dr}\int_r^1 \Phi_{l,0}(t)t(t^2-r^2)^{k-1}\mathrm{d}t = -2r(k-1)\int_r^1 \Phi_{l,0}(t)t(t^2-r^2)^{k-2}\mathrm{d}t.$$

Therefore equation (9) delivers an alternative way to define the Wendland functions, see [26]. Note that [26] uses a different numbering scheme of the functions than we do in this paper.

The Wendland functions have several important properties, cf. e.g. [11, Prop. 3.10]:

1) $\psi_{l,k}(r)$ is a polynomial of degree $l+2k$ for $r \in [0,1]$ and $\operatorname{supp}(\psi_{l,k}) = [0,1]$.
2) The radial function $\Psi(\mathbf{x}) := \psi_{l,k}(\|\mathbf{x}\|)$ is C^{2k} at 0.
3) $\psi_{l,k}$ is C^{k+l-1} at 1.

Frequently we fix the parameter $l := \lfloor \frac{n}{2} \rfloor + k + 1$, where n is the spacial dimension we are working in, and a constant $c > 0$ to fix the support. By the properties stated above, the radial function $\Psi(\mathbf{x}) := \psi_{l,k}(c\|\mathbf{x}\|)$ is then a C^{2k} function with $\operatorname{supp}(\Psi) = \mathcal{B}^d(0, c^{-1}) \subset \mathbb{R}^n$, where $\mathcal{B}^n(0, c^{-1})$ is the closed n-dimensional ball around the origin with radius c^{-1}.

4 Computing formulas for Wendland functions

In this section we introduce a method to generate Wendland functions of arbitrary degree. As a first step we discuss polynomial representations in software.

4.1 Polynomials Representation

We represent d-degree polynomials $\sum_{i=0}^{d} a_i t^i$ as a list of coefficients $(a_0, a_1, \ldots, a_d)$. Our implementation uses *Python* with *List* objects. Addition and multiplication of polynomials of this form are easily implemented as:

$$\sum_{i=0}^{d_1} a_i t^i + \sum_{j=0}^{d_2} b_j t^j = \sum_{i=0}^{\max\{d_1,d_2\}} (a_i + b_i) t^i,$$

where $a_i = 0$ for $i > d_1$ and $b_j = 0$ for $j > d_2$. Multiplication is given by

$$\left(\sum_{i=0}^{d_1} a_i t^i \right) \left(\sum_{j=0}^{d_2} b_j t^j \right) = \sum_{i=0}^{d_1+d_2} c_i t^i,$$

where

$$c_i = \sum_{k+j=i} a_k b_j.$$

An anti derivative of a polynomial is given by

$$(a_0, a_1, a_2, \ldots, a_d) \mapsto (0, a_0, \frac{a_1}{2}, \frac{a_2}{3}, \ldots, \frac{a_d}{d+1}),$$

corresponding to

$$\int \sum_{i=0}^{d} a_i t^i dt = \sum_{i=0}^{d} \frac{a_i}{i+1} t^{i+1},$$

and differentiation by

$$(a_0, a_1, \ldots, a_d) \mapsto (a_1, 2a_2, 3a_3, \ldots, d a_d).$$

In order to maximise exact calculations up to computer limitations, we store the coefficients as tuples of Integers, numerator and denominator, avoiding the floating point approximation. Specifically, we used the *Rational* class provided in *Python*. Polynomials in two variables can be represented as a polynomial in one of the variables, where each coefficient is a polynomial in the second variable, and each of those coefficients is a rational number. This gives us then a list of lists.

4.2 The Method

To calculate a polynomial representing the Wendland function $\psi_{l,k}$ on the interval $[0,1]$ we start by fixing the derivative

$$p'(t) = (1-t)^l t(t^2-r^2)^{k-1},$$

see (9). This function is a polynomial in two variables, which we represent as a polynomial in t where each coefficient is a polynomial in r. Following equation (9), we integrate this function with respect to t, and we obtain a new polynomial $p(t)$ in t, again with coefficients that are polynomials in r. We evaluate the polynomial p at $t=1$ and at $t=r$, which in both cases result in a polynomial in r, and we obtain the polynomial $\psi(r) = p(1) - p(r)$. Note that $\psi(r)$ is a representative of a Wendland function of order l,k, that is $\psi(r) = C_1\psi_{l,k}(r)$ for some constant $C_1 \neq 0$.

We factor the polynomial ψ, using long division, into the form

$$\psi(r) = C_2(1-r)^{l+k}p_{l,k}(r) \tag{10}$$

such that $p_{l,k}(r)$ is a polynomial with co-prime integer coefficients. This is possible since $\psi_{l,0}$ has a zero of order l at 1, and by using the recursive relation in equation (5), we see that $\psi_{l,k}$ has a zero of order $l+k$ at 1. The Wendland function $\psi_{l,k}$ is only defined up to a multiplication by a non-zero constant, therefore we are free to ignore the constant C_2 and use $\psi(r) = (1-r)^{l+k}p_{l,k}(r)$, a polynomial with integer coefficients, as a starting point for our recursion.

Using the relation in (6) and discarding the constant $C_{l,k}$, we see that

$$\begin{aligned}\psi_{l,k-1}(r) &= \frac{\frac{d}{dr}\left[(1-r)^{l+k}p_{l,k}(r)\right]}{r} \qquad (11)\\ &= \frac{1}{r}(1-r)^{l+k-1}((1-r)p'_{l,k}(r) - (l+k)p_{l,k}(r)).\end{aligned}$$

Writing the function $\psi_{l,k-1}$, as $\psi_{l,k-1}(r) = (1-r)^{l+k-1}p_{l,k-1}(r)$, then we see

$$\begin{aligned}p_{l,k-1}(r) &:= \frac{\psi_{l,k-1}(r)}{(1-r)^{l+k-1}} \qquad (12)\\ &= \frac{1}{r}\left[(1-r)p'_{l,k}(r) - (l+k)p_{l,k}(r)\right].\end{aligned}$$

We know that $\psi_{l,k-1}$ is a polynomial, since it is a Wendland function of order l, k, therefore $\frac{d}{dr}\left[(1-r)^{l+k}p_{l,k}(r)\right]$ must by divisible by the monomial r. Since $(1-r)^{l+k-1}$ is not divisible by r, the right-hand side of (12) must be a polynomial in r. Therefore $p_{l,k-1}$ is a well defined polynomial.

By pulling out the common factor $b_{k-1} \in \mathbb{Z}$ of the coefficients in $p_{l,k-1}$ we obtain a new polynomial $\hat{p}_{l,k-1}$ and a constant b_{k-1} such that

$$p_{l,k-1} = b_{k-1}\hat{p}_{l,k-1}.$$

Repeating this step, until we arrive at $p_{l,0}$, we get a collection of polynomials in the form

$$\psi_i(r) = b_1 \cdots b_i (1-r)^{l+k-i}\hat{p}_{l,k-i}(r), \; i = 1, 2, \ldots, k. \tag{13}$$

where each of the polynomials $\hat{p}_{l,k-i}(r)$ has co-prime integer coefficients and each of the constants b_i is a negative integer.

The above list follows the notation in [11], where ψ_0 is the polynomial given in (10) and is equal to the Wendland function $\psi_{l,k}$, and $\psi_1, \ldots, \psi_i$ are the Wendland functions given by $\psi_{l,k-1}, \ldots, \psi_{l,k-i}$ respectively, see equation (5). It is important to keep track of the constants $b_1, \ldots, b_i$ in (13) as they are necessary for correct evaluation of formula (1).

4.3 Example

We will now demonstrate how the above method determines the Wendland function $\psi_{l,k}$ for $l = 6$ and $k = 4$. Here we start with the function $p'(t) = (1-t)^6 t(t^2 - r^2)^3$ and we obtain

$$\begin{aligned}
\psi(r) &= \int_r^1 (1-t)^6 t(t^2-r^2)^3 dt \\
&= \frac{1}{280}r^{14} - \frac{32}{1001}r^{13} + \frac{1}{8}r^{12} - \frac{64}{231}r^{11} \\
&\quad + \frac{3}{8}r^{10} - \frac{32}{105}r^9 + \frac{1}{8}r^8 - \frac{1}{56}r^6 \\
&\quad + \frac{1}{280}r^4 - \frac{1}{1848}r^2 + \frac{1}{24024} \\
&= \frac{1}{120120}(1-r)^{10}(429r^4 + 450r^3 + 210r^2 + 50r + 1).
\end{aligned}$$

We set $\psi_0(r) = (1-r)^{10}(429r^4 + 450r^3 + 210r^2 + 50r + 1)$. For $r \in [0,1]$ we have the formulas (recall that $\psi_{l,k}(r) = 0$ if $r \notin [0,1]$):

$$\begin{aligned}
\psi_{6,4}(r) &= \psi_0(r) \\
&= (1-r)^{10}(429r^4 + 450r^3 + 210r^2 + 50r + 1); \\
\psi_{6,3}(r) &= \psi_1(r) = \frac{\frac{d}{dr}\psi_0(r)}{r} \\
&= -26(1-r)^9(231r^3 + 159r^2 + 45r + 5); \\
\psi_{6,2}(r) &= \psi_2(r) = \frac{\frac{d}{dr}\psi_1(r)}{r} \\
&= 3,432(1-r)^8(21r^2 + 8r + 1); \\
\psi_{5,1}(r) &= \psi_3(r) = \frac{\frac{d}{dr}\psi_2(r)}{r} \\
&= -102,960(1-r)^7(7r + 1); \\
\psi_{5,0}(r) &= \psi_4(r) = \frac{\frac{d}{dr}\psi_3(r)}{r} \\
&= 5,765,770(1-r)^6.
\end{aligned}$$

Note that we have actually computed a lot more useful information than just a family of Wendland functions $\psi_{5,i}$, $i = 0,1,2,3,4$. In our algorithm, for a fixed l, k, we have

$$\begin{aligned}
\psi_{l,k-j} = \psi_j(r) &= \frac{\frac{d}{dr}\psi_{j-1}(r)}{r} \\
&= \frac{\frac{d}{dr}\psi_{l,k-j+1}(r)}{r}, \quad \text{for } j = 1,\dots,k,
\end{aligned}$$

and we have thus delivered all the radial basis functions needed for a collocation problem. This corresponds to computing a whole table as in [11, Table 3.1], but for a collocation problem with arbitrary high derivatives. The software tool, discussed in Section 6, also includes the constant $c > 0$ in the computations, which is used to fix the support of the Wendland functions.

5 Meshless collocation using Wendland functions

The method of meshless collocation can be used to calculate Lyapunov functions for both deterministic dynamical systems and stochastic dynamical systems. We just choose the operator L and the boundary values appropriately. The next two sections show some of the explicit formulas involved. We also talk about the optimal grid for the interpolation problem, and some aspects of solving the resulting linear systems using software.

5.1 Deterministic systems

Consider an autonomous deterministic system, that is a dynamical system of the form

$$\mathbf{x}'(t) = f(\mathbf{x})$$

with $f : \mathbb{R}^n \to \mathbb{R}^n$, for which the origin is an asymptotically stable equilibrium. We can generate a Lyapunov function, $V : \mathbb{R}^n \to \mathbb{R}$, for this system by solving the interpolation problem (2) with the differential operator L being given by

$$LV(\mathbf{x}) = \langle \nabla V(\mathbf{x}), f(\mathbf{x}) \rangle,$$

setting the boundary $\Gamma = \emptyset$ and choosing the function h appropriately. Setting the radial basis function ψ to be the Wendland function $\psi_0(r) = \psi_{l,k}(r)$ for some constants l, k, and then fixing ψ_1 and ψ_2 according to equation (7), the matrix obtained in equation (3) is given by elements of the form (see [11]):

$$\begin{aligned} b_{kl} = {} & \psi_2(\|\mathbf{x}_k - \mathbf{x}_l\|)\langle \mathbf{x}_k - \mathbf{x}_l, f(\mathbf{x}_k)\rangle \langle \mathbf{x}_l - \mathbf{x}_k, f(\mathbf{x}_l)\rangle \\ & - \psi_1(\|\mathbf{x}_k - \mathbf{x}_l\|)\langle f(\mathbf{x}_k), f(\mathbf{x}_l)\rangle. \end{aligned} \tag{14}$$

The components of the vector γ are given by

$$\gamma_j = r(\mathbf{x}_j). \tag{15}$$

Then the solution to the interpolation problem has the formula, see equation (1),

$$V(\mathbf{x}) = \sum_{k=1}^{N} \alpha_k \psi_1(\|\mathbf{x} - \mathbf{x}_k\|)\langle \mathbf{x}_k - \mathbf{x}, f(\mathbf{x}_k)\rangle, \tag{16}$$

where α is the solution of

$$A\alpha = \gamma$$

and ψ_0 and ψ_1 are given by equation (7).

5.2 Stochastic Systems

For SDEs of the form

$$d\mathbf{x}(t) = f(\mathbf{x}(t))dt + g(\mathbf{x}(t))dW(t), \tag{17}$$

$f : \mathbb{R}^d \to \mathbb{R}^d$, $g : \mathbb{R}^d \to \mathbb{R}^{d\times Q}$, we consider the operator L given by the associated generator of the SDE:

$$\begin{aligned} LV(\mathbf{x}) &:= \nabla V(\mathbf{x}) \cdot f(\mathbf{x}) + \frac{1}{2}\sum_{i,j=1}^{d} [g(\mathbf{x})g(\mathbf{x})^\top] \frac{\partial^2 V}{\partial x_i \partial x_j}(\mathbf{x}) \\ &= \nabla V(\mathbf{x}) \cdot f(\mathbf{x}) + \frac{1}{2}\sum_{i,j}^{d} m_{ij}(\mathbf{x}) \frac{\partial^2}{\partial x_i \partial x_j} V(\mathbf{x}). \end{aligned} \tag{18}$$

Here $(m_{ij}(\mathbf{x}))_{i,j=1,\dots,d} = g(\mathbf{x})g(\mathbf{x})^\top$, that is $m_{ij}(\mathbf{x}) = \sum_{q=1}^{Q} g_{iq}(\mathbf{x})g_{jq}(\mathbf{x})$. We choose a Wendland function $\psi_0 = \psi_{l,k}$, for some constants l, k, and set $\psi(\mathbf{x}) = \psi_0(\|\mathbf{x}\|)$. We define ψ_i according to equation (7) for $i = \{1, 2, 3, 4\}$ and we get that the solution to the interpolation problem is given by, see equation (1),

$$\begin{aligned} V(\mathbf{x}) = \sum_{k=1}^{N} \alpha_k \Bigg[&- \psi_1(\|\mathbf{x}-\mathbf{x}_k\|)\langle \mathbf{x}-\mathbf{x}_k, f(\mathbf{x}_k)\rangle \\ &+ \frac{1}{2}\sum_{i,j}^{d} m_{ij}(\mathbf{x}_k)[\psi_2(\|\mathbf{x}-\mathbf{x}_k\|)(\mathbf{x}-\mathbf{x}_k)_i(\mathbf{x}-\mathbf{x}_k)_j + \delta_{i,j}\psi_1(\|\mathbf{x}-\mathbf{x}_k\|)]\Bigg] \\ &+ \sum_{k=1}^{M} \alpha_{N+k}\psi_0(\|\mathbf{x}-\xi_k\|). \end{aligned} \tag{19}$$

In this formula the vector α is the solution to the linear system in equation (2). The formulas for the matrix elements are

$$\begin{aligned} d_{kl} &= \psi_0(\|\xi_k - \xi_l\|), \\ c_{kl} &= -\psi_1(\|\xi_l - \mathbf{x}_k\|)\langle \xi_l - \mathbf{x}_k, f(\mathbf{x}_k)\rangle \\ &\qquad + \frac{1}{2}\sum_{i,j=1}^{d} m_{ij}(\mathbf{x}_k)[\psi_2(\|\xi_l-\mathbf{x}_k\|)(\xi_l-\mathbf{x}_k)_i(\xi_l-\mathbf{x}_k)_j \\ &\qquad + \delta_{ij}\psi_1(\|\xi_l-\mathbf{x}_k\|)], \end{aligned} \tag{20}$$

and, abbreviating $\beta = \mathbf{x} - \mathbf{x}_k$,

$$\begin{aligned} b_{kl} = &-\psi_2(\|\beta\|)\langle\beta, f(\mathbf{x}_k)\rangle\langle\beta, f(\mathbf{x}_l)\rangle - \psi_1(\|\beta\|)\langle f(\mathbf{x}_k), f(\mathbf{x}_l)\rangle \\ &+ \frac{1}{2}\sum_{i,j=1}^{d} m_{ij}(\mathbf{x}_l)\Bigg[\psi_3(\|\beta\|)\langle\beta, f(\mathbf{x}_k)\rangle\beta_i\beta_j + \psi_2(\|\beta\|)f_j(\mathbf{x}_k)\beta_i \\ &\qquad + \psi_2(\|\beta\|)f_i(\mathbf{x}_k)\beta_j + \delta_{ij}\psi_2(\|\beta\|)\langle\beta, f(\mathbf{x}_k)\rangle\Bigg] \\ &+ \frac{1}{2}\sum_{i,j=1}^{d} m_{ij}(\mathbf{x}_k)\Bigg[-\psi_3(\|\beta\|)\langle\beta, f(\mathbf{x}_l)\rangle\beta_i\beta_j - \psi_2(\|\beta\|)f_j(\mathbf{x}_l)\beta_i \\ &\qquad - \psi_2(\|\beta\|)f_i(\mathbf{x}_l)\beta_j - \delta_{ij}\psi_2(\|\beta\|)\langle\beta, f(\mathbf{x}_l)\rangle\Bigg] \\ &+ \frac{1}{4}\sum_{r,s=1}^{d}\sum_{i,j=1}^{d} m_{rs}(\mathbf{x}_k)m_{ij}(\mathbf{x}_l)\Bigg[\psi_4(\|\beta\|)\beta_i\beta_j\beta_r\beta_s \\ &\qquad + \psi_3(\|\beta\|)[\delta_{ij}\beta_r\beta_s + \delta_{ir}\beta_j\beta_s + \delta_{is}\beta_j\beta_r \\ &\qquad\qquad + \delta_{jr}\beta_i\beta_s + \delta_{js}\beta_i\beta_r + \delta_{rs}\beta_i\beta_j] \\ &\qquad + \psi_2(\|\beta\|)[\delta_{ij}\delta_{rs} + \delta_{ir}\delta_{js} + \delta_{is}\delta_{jr}]\Bigg]. \end{aligned} \tag{21}$$

5.3 Generating the grid

The optimal grid in $\Omega \subset \mathbb{R}^n$ for the interpolation problem (2) was studied in [17]. The grid that delivers the smallest fill-distance, the parameter which determines the accuracy of the solution, is defined using the basis vectors $w_1, w_2, \ldots, w_n \in \mathbb{R}^n$, where (the e_is denote the usual orthonormal basis in $\mathbb{R}^n$)

$$w_k = \sum_{j=1}^{k-1} \epsilon_j e_j + (k+1)\epsilon_k e_k \quad \text{and} \quad \epsilon_k = \frac{1}{\sqrt{2k(k+1)}}.$$

The grid-points $G_{\alpha,z} = \{g_{\mathbf{i}} : \mathbf{i} \in \mathbb{Z}^n\} \subset \mathbb{R}^n$ with fill=distance parameter $\alpha > 0$ and offset $z \in \mathbb{R}^n$ are then given by

$$g_{\mathbf{i}} := z + \alpha \sum_{k=1}^{n} i_k w_k, \quad \mathbf{i} = (i_1, i_2, \ldots, i_n) \in \mathbb{Z}^n.$$

Given two vectors $a, b \in \mathbb{R}^n$ such that $a_i < b_i$ for $i = 1, 2, \ldots, n$, we want to compute the coordinates of the grid-points $g_{\mathbf{i}} \in G_{\alpha,z}$ that are in the cube

$$C_{a,b} := [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_n, b_n].$$

By observing that w_n is the only basis vector with a nonzero entry in its last component, w_{n-1} and w_n are the only basis vectors with nonzero entries in their second to last component, etc. , these can be computed efficiently in a recursive manner. Let us illustrate this with $n = 3$, the general strategy can be read from the code below.

Given the offset vector $z = (z_1, z_2, z_3) \in \mathbb{R}^3$ and $a_3 < b_3$, we see that only those $\mathbf{i} = (i_1, i_2, i_3) \in \mathbb{Z}^3$ with i_3 fulfilling

$$a_3 \leq z_3 + i_3 \cdot \alpha \cdot (3+1)\frac{1}{\sqrt{2 \cdot 3 \cdot (3+1)}} = z_3 + i_3 \cdot \alpha \sqrt{\frac{3+1}{2 \cdot 3}} \leq b_3,$$

i.e.

$$\left\lceil \frac{a_3 - z_3}{\alpha}\sqrt{\frac{2 \cdot 3}{3+1}} \right\rceil \leq i_3 \leq \left\lfloor \frac{b_3 - z_3}{\alpha}\sqrt{\frac{2 \cdot 3}{3+1}} \right\rfloor, \tag{22}$$

have to be considered, because all other choices of i_3 deliver an entry in the third component that is not in the interval $[a_3, b_3]$. For each i_3 fulfilling this inequality, let us denote it i_3^*, we can generate appropriate i_2 components by observing that $g_{\mathbf{i}}$ with $\mathbf{i} = (0, 0, i_3^*)$ has the entry

$$z_2^* := z_2 + \alpha \cdot i_3^* \cdot \epsilon_2 = z_2 + \alpha \cdot i_3^* \cdot \frac{1}{\sqrt{2 \cdot 2 \cdot (2+1)}}$$

in its second component. The appropriate i_2s for i_3^* are thus given by considering the formula for w_2 and are easily seen to fulfill

$$a_2 \leq z_2^* + \alpha \cdot i_2 \cdot (2+1)\epsilon_2 = z_2 + \alpha \cdot i_3^* \cdot \epsilon_2 + \alpha \cdot i_2 \cdot (2+1)\epsilon_2 \leq b_2,$$

which can be written similarly to before as

$$\left\lceil \frac{a_2 - z_2^*}{\alpha}\sqrt{\frac{2\cdot 2}{2+1}} \right\rceil \leq i_2 \leq \left\lfloor \frac{b_2 - z_2^*}{\alpha}\sqrt{\frac{2\cdot 2}{2+1}} \right\rfloor . \tag{23}$$

Now, having fixed an i_3^* fulfilling (22) and subsequently an i_2^* for this i_3^* fulfilling (23), we can in a similar manner compute appropriate i_1s. The vector $g_{\mathbf{i}}$ with $\mathbf{i} = (0, i_2^*, i_3^*)$ has the entry

$$z_1^* := z_1 + \alpha \cdot (i_2^* \epsilon_1 + i_3^* \epsilon_1) = z_1 + \alpha \cdot (i_2^* + i_3^*) \cdot \frac{1}{\sqrt{2\cdot 1 \cdot (1+1)}}$$

in its first component. Similarly to before the appropriate i_1s in (i_1, i_2^*, i_3^*) are read from an inequality:

$$a_1 \leq z_1^* + \alpha \cdot i_1 \cdot (1+1)\epsilon_1 = z_1 + \alpha \cdot (i_2^* + i_3^*) \cdot \frac{1}{\sqrt{2\cdot 1 \cdot (1+1)}} + \alpha \cdot i_1 \cdot (1+1)\epsilon_1 \leq b_1$$

or

$$\left\lceil \frac{a_1 - z_1^*}{\alpha}\sqrt{\frac{2\cdot 1}{1+1}} \right\rceil \leq i_1 \leq \left\lfloor \frac{b_1 - z_1^*}{\alpha}\sqrt{\frac{2\cdot 1}{1+1}} \right\rfloor .$$

This recursive procedure computes all grid vectors $g_{\mathbf{i}} \in G_{\alpha,z}$ in the cube $C_{a,b}$ and is implemented in C++ using the Armadillo library [21] is given in Listing 1.1.

Listing 1.1. Code in C++ that generates the optimal grid

```
list<arma::vec> HexaGridnew(arma::vec a, arma::vec b, ...
    double c, int N) {
    a = a(span(0, N - 1));
    b = b(span(0, N - 1));
    double tol = 1e-10; // add a small tolerance to the cube
    a -= tol*ones<vec>(N);
    b += tol*ones<vec>(N);
    list<vec> Ret;
    unsigned int i, k;
    vec e(N, fill::zeros);

    for (k = 1; k <= N; k++) {
        e(k - 1) = sqrt(1.0 / (2.0*k*(k + 1)));
    }
    vector<vec> w(N);
    for (i = 1; i <= N; i++) {
        vec v(N, fill::zeros);
        for (int k = 1; k < i; k++) {
            v(k - 1) = e(k - 1);
        }
        v(i - 1) = (i + 1)*e(i - 1);
        w[i - 1] = v;
    }

    function<void(int, vec)> ML = [&](int r, vec x) {
        for (int i = int(ceil((a(r) - x(r)) / (c*(r + ...
            2)*e(r)))); i <= int(floor((b(r) - x(r)) / ...
            (c*(r + 2)*e(r)))); i++) {
            if (r == 0) {
                Ret.push_back(x+i*c*w[r]);
            }
            else {
                ML(r - 1, x + i*c*w[r]);
            }
        }
    };
    ML(N - 1, 0.5*c*w[N-1]);
    return Ret;
}
```

5.4 Solving the linear system

The linear system of equations that we obtain when solving the interpolation problem is defined by a symmetric and positive definite matrix A, see equation (3). LAPACK [1] has specific methods for solving these types of equations, that use the Cholesky decomposition of the matrix $A = U^\top U$, where U is upper triangular with positive diagonal entries. The function DPOSV overwrites the contents of the matrix A with the Cholesky decomposition U and solves the system $A\alpha = \gamma$. The acronym is understood in the following way, D stands for Double, PO stands for Symmetric or Hermitian positive definite and SV stands for solve. This has much better numerical properties than solving the system with e.g. LU-decomposition.

It is also possible to store the matrix A in packed format, that is, since A is a symmetric matrix, we can store just the upper triangular part of it. This saves a considerable amount of memory. LAPACK has functions for computing the Cholesky decomposition of the matrix A in packed format. The function DPPTRF calculates the Cholesky decomposition of A in packed format, overwriting the contents of matrix A, and the function DPPTRS solves then the system $A\alpha = \gamma$ using the Cholesky factor computed by DPPTRF. Here the letters PP stand for Symmetric or Hermitian positive definite in packed storage, TRF means factorize to a product of triangular matrices, and TRS stands for solving the factorized system using forward or backwards substitution.

For ease of usage we have implemented functions that calculate the interpolation matrices described before, i.e. equations (3),(14),(20) and (21), for both stochastic and deterministic dynamical systems. These are available in the software repository.

6 Software library

We have implemented the algorithm described in Section 4.2 in a software tool[1] that generates C/C++ code versions of the Wendland functions in factorized form. In a previous work [3], we determined that the most efficient and accurate way to evaluate these Wendland functions was to use this factorized form. Evaluating these polynomials in fully expanded format using Horner's scheme [8], can lead to very large numerical errors as shown in [3]. We give a brief summary of these results in a later section and in Table 1. Below in Listing 1.2 is a part of the library generated by our tool, which shows the family of Wendland functions obtained when starting with $\varPsi_0(\mathbf{x}) = \psi_{5,4}(c\|\mathbf{x}\|)$, where $c > 0$ is the constant that controls the support of the radial function $\varPsi$.

Listing 1.2. Generated code for the $\psi_{6,4}$ family

```
double __wendlandpsi_6_4_0(double x, double c){
    double t=__ipow((1.0-x),10);
    t=1.0*t*(((((429)*x + 450)*x + 210)*x + 50)*x + 5);
    return t;
}
double __wendlandpsi_6_4_1(double x, double c){
    double t=__ipow((1.0-x),9);
    t=-26.0*t*__ipow(c,2)*((((231)*x + 159)*x + 45)*x + 5);
    return t;
}
double __wendlandpsi_6_4_2(double x, double c){
    double t=__ipow((1.0-x),8);
    t=3432.0*t*__ipow(c,4)*(((21)*x + 8)*x + 1);
    return t;
}
double __wendlandpsi_6_4_3(double x, double c){
    double t=__ipow((1.0-x),7);
    t=-102960.0*t*__ipow(c,6)*((7)*x + 1);
    return t;
}
double __wendlandpsi_6_4_4(double x, double c){
    double t=__ipow((1.0-x),6);
    t=5765760.0*t*__ipow(c,8)*(1);
    return t;
}
```

Note that `__wendlandpsi_6_4_j` corresponds to ψ_j in the example, but with $x = cr$ as argument.

When starting with $\varPsi_0(x) = \psi_{6,3}(c\|x\|)$ instead, the relevant definitions are given in Listing 1.3.

[1] The tool is available at https://gitlab.com/hjortur/wendland-function-generator/ with example outputs.

Listing 1.3. Generated code for the $\psi_{6,3}$ family

```
double __wendlandpsi_6_3_0(double x, double c){
    double t=__ipow((1.0-x),9);
    t=1.0*t*((((231)*x + 159)*x + 45)*x + 5);
    return t;
}
double __wendlandpsi_6_3_1(double x, double c){
    double t=__ipow((1.0-x),8);
    t=-132.0*t*__ipow(c,2)*(((21)*x + 8)*x + 1);
    return t;
}
double __wendlandpsi_6_3_2(double x, double c){
    double t=__ipow((1.0-x),7);
    t=3960.0*t*__ipow(c,4)*((7)*x + 1);
    return t;
}
double __wendlandpsi_6_3_3(double x, double c){
    double t=__ipow((1.0-x),6);
    t=-221760.0*t*__ipow(c,6)*(1);
    return t;
}
```

Note that the polynomials `__wendlandpsi_6_3_1` and `__wendlandpsi_6_4_2` differ only by a multiplication of a constant and a power of c, and both polynomials are a representative of the Wendland function $\psi_{6,2}$.

The function `__ipow(x,i)` evaluates x^i where x is a double and i is a positive integer. We have "flattened" the functions `__wendlandpsi_x_y_z` in the sense that their domain is $[0, 1]$. They require the user to pre-multiply the x value with the chosen RBF-constant $c > 0$, that is for $\Psi(\mathbf{x}) = \psi_{l,k}(c\|\mathbf{x}\|)$, the user needs to pass in the value $c\|\mathbf{x}\|$ and c after ensuring that $c\|\mathbf{x}\| \in [0, 1]$. A possible implementation using the Armadillo library [21] can be see in listing 1.4.

Listing 1.4. Example usage

```
double psi3(const arma::vec &x, double c){
    double cx=c*arma::norm(x,2);
    return ( cx < 1.0 ) ? __wendlandpsi_6_4_3(cx,c) : 0.0;
}
```

The tool is a simple Python script named *wendlandfunctions.py*. When the script is run it outputs text for code- and header-files, which contain the Wendland function definitions. The user can supply the script with a parameter `--l` and an integer value $m \geq 2$, in order to output code for Wendland functions from $\psi_{2,1}$ up to $\psi_{m,i}$ for all $0 \leq i < m$.

6.1 Example Lyapunov functions

Included in the repository are example outputs and example programs that calculate Lyapunov functions for deterministic and stochastic systems, using the software library that our tool generates and functions that generate the optimal interpolation grid and the interpolation matrices. Figures 1, 2, 3, and 4 show graphs of Lyapunov functions obtained from these example programs, where the systems considered are:

$$\begin{bmatrix} x'(t) \\ y'(t) \end{bmatrix} = \begin{bmatrix} y(t) \\ -x(t) - (1 - x(t)^2)y(t) \end{bmatrix}, \tag{24}$$

$$\begin{bmatrix} x'(t) \\ y'(t) \end{bmatrix} = \begin{bmatrix} y(t) \\ -x(t) + \frac{1}{3}x(t)^3 - y(t) \end{bmatrix}, \tag{25}$$

and the stochastic systems

$$dx = \sin(x)dt + \frac{3x}{1+x^2}dW, \tag{26}$$

$$d\mathbf{x} = \begin{bmatrix} \|\mathbf{x}\| - 1.0 & 1.0 \\ -1.0 & \|\mathbf{x}\| - 1.0 \end{bmatrix} \mathbf{x}dt + \|\mathbf{x}\|(\|\mathbf{x}\| - 0.5)(\|\mathbf{x}\| - 1.5)\,\mathbf{x}\,dW. \tag{27}$$

For the systems in the above equations we have used the optimal grid as described in Section 5.3 as the collocation/interpolation points. Denoting by $\mathcal{B}^2(\mathbf{x}, r)$ the 2 dimensional open ball around $\mathbf{x}$ with radius r and

$$\Gamma_r(N) = \left\{ r\left(\cos\left(\frac{j2\pi}{N}\right), \sin\left(\frac{j2\pi}{N}\right)\right) \quad | \quad j \in \{1, \dots, N\} \right\} \subset \mathbb{R}^2,$$

we used the following data to calculate the Lyapunov functions:

- Collocation grid on $[-2,2] \times [-2,2] \setminus \mathcal{B}^2(0, 0.1)$ with fill-distance parameter $\alpha = \frac{4}{15}$, $LV(\mathbf{x}) = -\|\mathbf{x}\|$, and $\psi_0(\mathbf{x}) = \psi_{6,4}(\|\mathbf{x}\|)$ for the system in equation (24);
- Collocation grid on $[-1.4, 1.4] \times [-1.4, 1.4] \setminus \mathcal{B}^2(0, 0.1)$ with fill distance parameter $\alpha = \frac{2.8}{20}$, $LV(\mathbf{x}) = -\|\mathbf{x}\|$ and $\psi_0(\mathbf{x}) = \psi_{5,3}(\|\mathbf{x}\|)$ for the system in equation (25);
- Collocation grid on $[0.1, 8.0]$ with fill distance parameter $\alpha = \frac{1}{400}$, $LV(x) = 10^{-4}$, $V(0.1) = 0$, $V(8.0) = 1.0$ and $\psi_0(\mathbf{x}) = \psi_{7,6}(2\|\mathbf{x}\|)$ for the system in equation (26);
- Collocation grid on $[-2,2] \times [-2,2] \setminus \mathcal{B}^2(0, 0.4)$ with fill distance parameter $\alpha = \frac{1}{25}$, $LV(\mathbf{x}) = 10^{-2}$, $V(\xi_j) = 0$ and $V(\beta_j) = 1$ for all $\xi_j \in \Gamma_{0.4}(4)$ and all $\beta_j \in \Gamma_{1.9}(80)$. Furthermore we set $\psi_0(\mathbf{x}) = \psi_{6,4}(\|\mathbf{x}\|)$, for the system in equation (27).

Note that the resulting Lyapunov functions for the systems in equations (26) and (27) in figures 3 and 4 are comparable to the results obtained in [5] and [15].

6.2 Comparison of evaluation methods

In the paper [3] we compared different methods of evaluation for Wendland functions at a point. The methods used where:

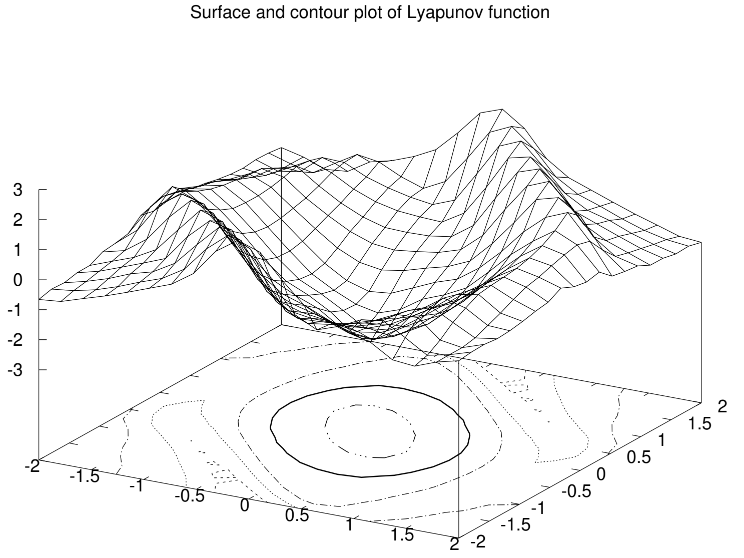


Fig. 1. Lyapunov function for the system in equation (24)

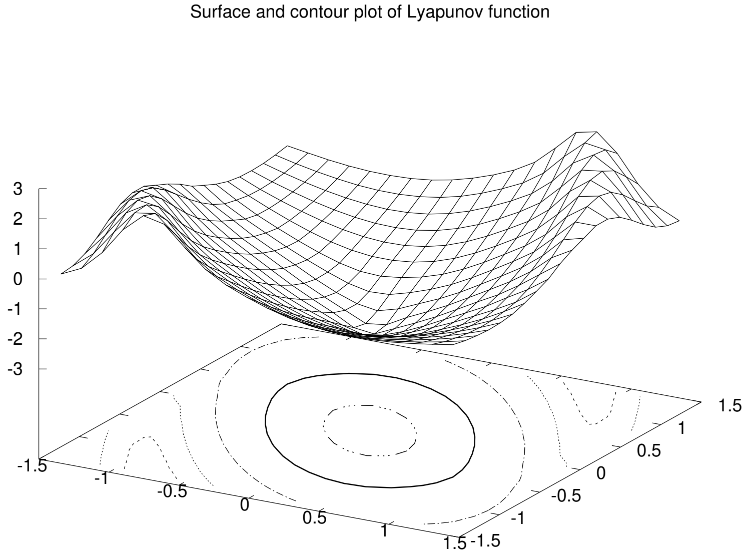


Fig. 2. Lyapunov function for the system in equation (25)

- Having them in factorized form, as our software tool provides, see Listing 1.2;
- Fully expanded polynomials and evaluated using Horner's Scheme, as in [2];
- Pre-computing the function in high precision (see below) at 10^7 evenly spaced points on the interval $[0, 1]$ and using them as a lookup table. That is, round to the closest value;

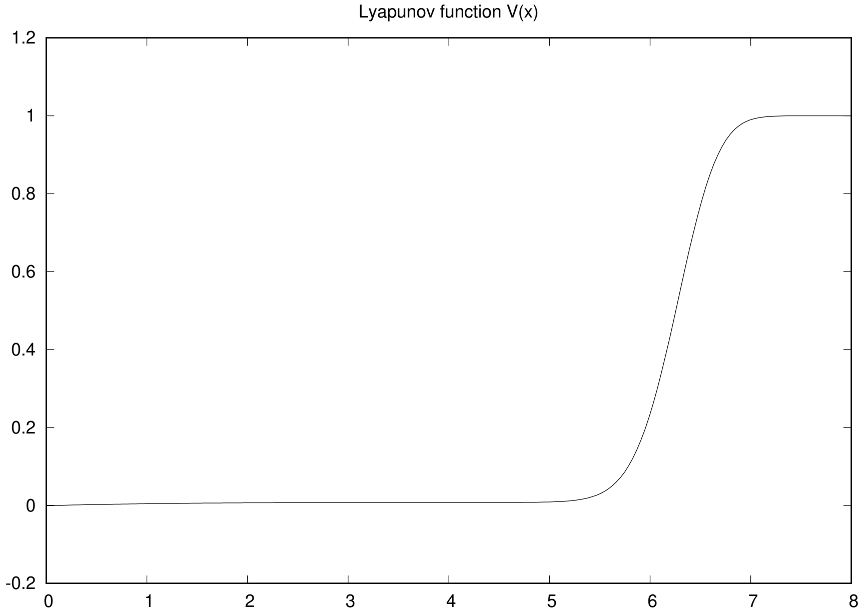


Fig. 3. Lyapunov function for the stochastic system in equation (26)

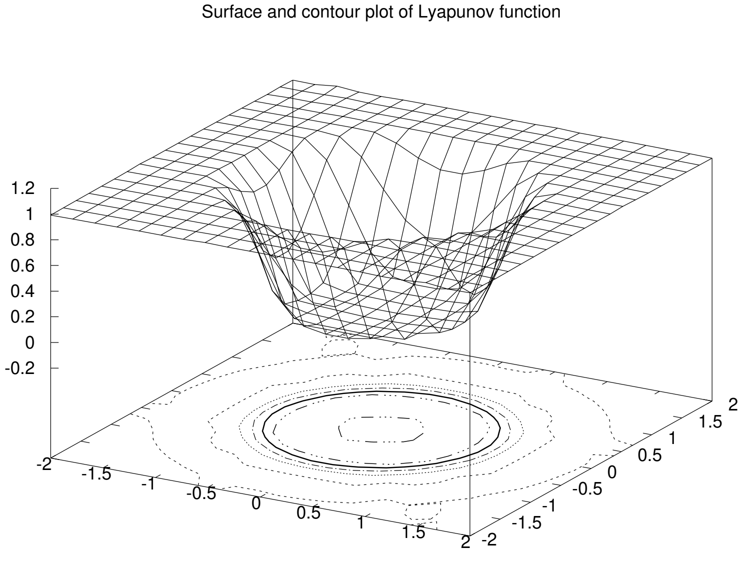


Fig. 4. Lyapunov function for the stochastic system in equation (27)

– Using the same lookup table but additionally linearly interpolate between two nearest neighbours to improve accuracy.

Table 1 shows time elapsed to evaluate the Wendland function $\psi_{7,2}$ at 10^7 different points on the interval $[0, 1]$, and the scale of the relative error obtained on this interval. For further analysis see [3].

Method / Processor	i5-8250U	i7-4790K	Rel.error
Factorized form	171.5ms	107ms	10^{-13}
Horner's scheme	548.1ms	395ms	1
Lookup table	125.8ms	105ms	10^{-5}
Lookup table with interpolation	165.5ms	128ms	10^{-9}

Table 1. Evaluation of $\psi_{7,2}$ at 10^7 different points, for different CPUs

7 Conclusion

In this paper we have presented a software tool for generating Wendland's compactly supported Radial Basis Functions in an optimal form. This tool generates a C/C++ library for Wendland functions of arbitrary degree in factorized form. Furthermore, this tool generates an entire family of these functions, used for solving collocation problesm, for each initial Wendland function $\psi_{l,k}$. We have also presented an algorithm that this software tool uses for generating Wendland functions in this factorized form, for accurate and efficient evaluations. Finally, we have created a software library for calculating Lyapunov functions for both stochastic and deterministic dynamical systems, using these factorized Wendland functions that our tool generates. All the software, with example usage, is available for download at https://gitlab.com/hjortur/wendland-function-generator/.

Acknowledgement: This research was supported by the Icelandic Research Fund (Rannis), grant number 152429-051, Lyapunov Methods and Stochastic Stability.

References

1. Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D.: LAPACK Users' Guide. Society for Industrial and Applied Mathematics, 3 edn. (1999)
2. Argaez, C., Hafstein, S., Giesl, P.: Wendland functions - a C++ code to compute them. In: Proceedings of the 7th International Conference on Simulation and Modeling Methodologies, Technologies and Applications - Volume 1: SIMULTECH,. pp. 323–330. INSTICC, SciTePress (2017)
3. Bjornsson, H., Hafstein, S.: Verification of a numerical solution to a collocation problem. In: Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics - Volume 1: CTDE,. pp. 587–594. INSTICC, SciTePress (2018)
4. Bjornsson, H., Hafstein, S.: Algorithm and software to generate code for Wendland functions in factorized form. In: Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics. pp. 156–162. INSTICC, SciTePress (2019)
5. Bjornsson, H., Hafstein, S., Giesl, P., Gudmundsson, S., Scalas, E.: Computation of the stochastic basin of attraction by rigorous construction of a Lyapunov function. Discrete and Continuous Dynamical Systems-Series B **24**(8), 4247–4269 (2019)

6. Buhmann, M.: Radial basis functions. In: Acta numerica, 2000, Acta Numer., vol. 9, pp. 1–38. Cambridge Univ. Press, Cambridge (2000)
7. Buhmann, M.: Radial basis functions: theory and implementations, Cambridge Monographs on Applied and Computational Mathematics, vol. 12. Cambridge University Press, Cambridge (2003)
8. Burrus, C.S., Fox, J.W., Sitton, G.A., Treitel, S.: Horner's method for evaluating and deflating polynomials. DSP Software Notes, Rice University, Nov **26** (2003)
9. Floater, M., Iske, A.: Multistep scattered data interpolation using compactly supported Radial Basis Functions. J. Comput. Appl. Math. **73**(1-2), 65–78 (1996)
10. Franke, C., Schaback, R.: Solving partial differential equations by collocation using radial basis functions. Appl. Math. Comput. **93**(1), 73–82 (1998)
11. Giesl, P.: Construction of Global Lyapunov Functions Using Radial Basis Functions, Lecture Notes in Mathematics, vol. 1904. Springer-Verlag, Berlin (2007)
12. Giesl, P.: Construction of a local and global Lyapunov function using radial basis functions. IMA J. Appl. Math. **73**(5), 782–802 (2008)
13. Giesl, P.: Construction of a finite-time Lyapunov function by meshless collocation. Discrete Contin. Dyn. Syst. Ser. B **17**(7), 2387–2412 (2012)
14. Giesl, P., Hafstein: Computation and verification of Lyapunov functions. SIAM Journal on Applied Dynamical Systems **14**(4), 1663–1698 (2015)
15. Grüne, L., Camilli, F.: Characterizing attraction probabilities via the stochastic Zubov equation. Discrete Contin. Dyn. Syst. Ser. B **3**(3), 457–468 (2003)
16. Hafstein, S., Gudmundsson, S., Giesl, P., Scalas, E.: Lyapunov function computation for autonomous linear stochastic differential equations using sum-of-squares programming. Discrete and Continuous Dynamical Systems - Series B **23**(2), 939–950 (2018)
17. Iske, A.: Perfect centre placement for radial basis function methods. Tech. Rep. TUM-M9809, TU Munich, Germany (1998)
18. Khalil, H.: Nonlinear systems. Pear, 3. edn. (2002)
19. Khasminskii, R.: Stochastic stability of differential equations. Springer, 2nd edn. (2012)
20. Mao, X.: Stochastic Differential Equations and Applications. Woodhead Publishing, 2nd edn. (2008)
21. Sanderson, C.: Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments. Tech. rep., NICTA. (2010)
22. Sastry, S.: Nonlinear Systems: Analysis, Stability, and Control. Springer (1999)
23. Vidyasagar, M.: Nonlinear System Analysis. Classics in applied mathematics, SIAM, 2. edn. (2002)
24. Wendland, H.: Error estimates for interpolation by compactly supported Radial Basis Functions of minimal degree. J. Approx. Theory **93**, 258–272 (1998)
25. Wendland, H.: Scattered data approximation, Cambridge Monographs on Applied and Computational Mathematics, vol. 17. Cambridge University Press, Cambridge (2005)
26. Wendland, H.: Multiscale radial basis functions. In: Frames and other bases in abstract and function spaces, pp. 265–299. Appl. Numer. Harmon. Anal., Birkhäuser/Springer, Cham (2017)
27. Wu, Z.: Hermite-Birkhoff interpolation of scattered data by radial basis functions. Approx. Theory Appl. **8**(2), 1–10 (1992)

Paper V

Algorithm and Software to Generate Code for Wendland Functions in Factorized Form

Bjornsson, H. and Hafstein, S.

Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO)

Algorithm and Software to Generate Code for Wendland Functions in Factorized Form

Hjortur Bjornsson and Sigurdur Hafstein[a]
Science Institute, University of Iceland, Dunhagi 3, 107 Reykjavík, Iceland

Keywords: Wendland Function, Lyapunov Functions, Radial Basis Functions, Code Generation.

Abstract: In this paper we describe an algorithm to determine Wendland's Radial Basis Functions in a specific factorized form. Additionally, we present a software tool that uses this algorithm to generate a C/C++ library that implements the Wendland functions with arbitrary parameters in factorized form. This library is more efficient and has higher numerical accuracy than previous implementations. The software tool is written in Python and is available for download.

1 INTRODUCTION

Interpolation and collocation using Radial Basis Functions (RBF), in particular compactly supported RBFs, has been the subject of numerous research activities in the past decades (Wu, 1992; Floater and Iske, 1996; Franke and Schaback, 1998; Wendland, 1998; Buhmann, 2003; Buhmann, 2000; Wendland, 2005; Wendland, 2017). They are well suited as kernels of Reproducing Kernel Hilbert Spaces and their mathematical theory is mature. The authors and their collaborators have applied Wendland's compactly supported RBFs for computing Lyapunov functions for nonlinear systems, both deterministic (Giesl, 2007; Giesl, 2008; Giesl and Hafstein, 2015) and stochastic (Bjornsson et al., 2019). Lyapunov functions are a useful tool to analyse stability of various dynamical systems, either deterministic or stochastic, cf. e.g. (Khalil, 2002; Sastry, 1999; Vidyasagar, 2002; Khasminskii, 2012; Mao, 2008). Various numerical methods have been used to find Lyapunov functions for the systems at hand (Giesl and Hafstein, 2015; Hafstein et al., 2018). Meshless collocation using RBFs is one such method and many different families of RBFs have been studied (Wendland, 2005).

In the papers (Giesl and Hafstein, 2015; Bjornsson et al., 2019) meshless collocation with so called Wendland functions is used. The Wendland functions are compactly supported radial functions, that are polynomials on their support. The Wendland function family is defined in such a way that many tedious and error prone calculations have to be done by hand in order to obtain formulas for the functions to be used in the software implementation in (Giesl and Hafstein, 2015; Bjornsson et al., 2019). In (Argaez et al., 2017) an algorithm is proposed that determines the Wendland polynomials in expanded form, that is: for each integer $l,k \geq 0$ finds a list of numbers $a_0, a_1, \ldots a_d$ such that the Wendland function $\psi_{l,k}(r) = \sum_{i=0}^{d} a_i r^i$ on its compact support. However, it was shown in (Bjornsson and Hafstein, 2018) that the evaluations of these polynomials in this form using typical schemes, such as Horner's scheme, can lead to significant numerical errors.

Evaluating the Wendland functions in factorized form (Bjornsson and Hafstein, 2018) is more efficient and numerically accurate, so we propose an alternate algorithm to determine the functions in factorized form. In addition to developing the algorithm, we implemented it and created a software tool that generates a reusable software library, which implements these Wendland polynomials in factorized form in C/C++.

2 BACKGROUND

In the paper (Bjornsson et al., 2019), meshless collocation using RBFs was used to calculate Lyapunov functions for various Stochastic Differential Equations (SDE). This included computing, for a given domain $\Omega \subset \mathbb{R}^n$ and a boundary $\Gamma \subset \mathbb{R}^d$, a solution to the Partial Differential Equation (PDE)

[a] https://orcid.org/0000-0003-0073-2765

Bjornsson, H. and Hafstein, S.
Algorithm and Software to Generate Code for Wendland Functions in Factorized Form.
DOI: 10.5220/0008191901560162
In *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2019)*, pages 156-162
ISBN: 978-989-758-380-3

$$\begin{cases} LV(x) = h(x) & x \in \Omega \\ V(x) = c(x) & x \in \Gamma, \end{cases} \tag{1}$$

where L is a certain second-order differential operator, and h and c are appropriately chosen functions. A numerical solution to the above problem was determined by choosing points $X_1 = \{x_1, \ldots, x_N\} \subset \Omega$ and $X_2 = \{\xi_1, \ldots, \xi_M\} \subset \Gamma$ and solving the interpolation problem

$$\begin{cases} LV(x_i) = h(x_i) & \text{for all } i = 1, \ldots, N \\ V(\xi_i) = c(\xi_i) & \text{for all } i = 1, \ldots, M. \end{cases}$$

The solution to this interpolation problem is given by

$$\begin{aligned} V(x) &= \sum_{k=1}^{N} \alpha_k (\delta_{x_k} \circ L)^y \psi(\|x-y\|) \\ &\quad + \sum_{k=1}^{M} \alpha_{N+k} (\delta_{\xi_k} \circ L^0)^y \psi(\|x-y\|), \end{aligned} \tag{2}$$

where $\delta_y V(x) = V(y)$, the superscript y denotes that the operator is applied with respect to the variably y, and the operator L^0 is the identity operator. Here the function ψ is a compactly supported RBF (Wendland, 2017). The constants α_i are determined as a solution to the linear system $A\alpha = \gamma$, where A is the symmetric and positive definite matrix

$$A = \begin{bmatrix} B & C \\ C^T & D \end{bmatrix}$$

and the matrices $B = (b_{jk})_{j,k=1,\ldots,N}$, $C = (c_{jk})_{j=1,\ldots,N,k=1,\ldots,M}$ and $D = (d_{jk})_{j,k=1,\ldots,M}$ have elements:

$$\begin{aligned} b_{jk} &= (\delta_{x_j} \circ L)^x (\delta_{x_k} \circ L)^y \psi(x-y) \\ c_{jk} &= (\delta_{x_j} \circ L)^x (\delta_{\xi_k} \circ L^0)^y \psi(x-y) \\ d_{jk} &= (\delta_{\xi_j} \circ L^0)^x (\delta_{\xi_k} \circ L^0)^y \psi(x-y). \end{aligned}$$

To compute such a Lyapunov function a large number of evaluations of the function ψ and its derivatives is necessary. To verify the properties of a Lyapunov function for the function computed, even more evaluations are necessary. Therefore, it turned out to be essential that these evaluations could be carried out in an efficient and accurate way.

3 WENDLAND FUNCTIONS

The Wendland functions are a family of functions, depending on two parameters $l, k \in \mathbb{N}_0$ defined by

$$\psi_{l,0}(r) = [(1-r)_+]^l \tag{3}$$

and

$$\psi_{l,k+1}(r) = C_{l,k+1} \int_r^1 t \psi_{l,k}(t) \mathrm{d}t, \tag{4}$$

where $(1-r)_+ := \max\{1-r, 0\}$ and $C_{l,k+1} \neq 0$ is a constant. For interpolation and collocation using a particular Wendland function, the value of the constant $C_{l,k+1} \neq 0$ is not of importance because the Wendland function appears linearly on both sides of a linear equation. Therefore, one can just fix values that are convenient for the problem at hand and we will do this in the following section. These functions satisfy the relation

$$-C_{l,k+1} \psi_{l,k}(r) = \frac{\frac{d}{dr} \psi_{l,k+1}(r)}{r}. \tag{5}$$

It is not difficult to verify that the functions

$$\Phi_{l,0}(r) = [(1-r)_+]^l \quad \text{and} \tag{6}$$

$$\Phi_{l,k}(r) = \int_r^1 \Phi_{l,0}(t) t (t^2 - r^2)^{k-1} \mathrm{d}t \quad \text{for } k > 0 \tag{7}$$

also satisfy a relation of the form

$$-2(k-1)\Phi_{l,k}(r) = \frac{\frac{d}{dr} \Phi_{l,k+1}(r)}{r},$$

for all integers $k, l \geq 0$, i.e. a relation identical to equation (5) with $C_{l,k+1} = 2(k-1)$. Just note that

$$\begin{aligned} &\frac{d}{dr} \int_r^1 \Phi_{l,0}(t) t (t^2 - r^2)^{k-1} \mathrm{d}t \\ &\quad = -2r(k-1) \int_r^1 \Phi_{l,0}(t) t (t^2 - r^2)^{k-2} \mathrm{d}t. \end{aligned}$$

Therefore (7) delivers an alternative way to define the Wendland functions, see (Wendland, 2017). Note that (Wendland, 2017) uses a different numbering scheme of the functions than we do.

The Wendland functions have several important properties, cf. e.g. (Giesl, 2007, Prop. 3.10):

1. $\psi_{l,k}(r)$ is a polynomial of degree $l + 2k$ for $r \in [0,1]$ and $\mathrm{supp}(\psi_{l,k}) = [0,1]$.
2. The radial function $\Psi(x) := \psi_{l,k}(\|x\|)$ is C^{2k} at 0.
3. $\psi_{l,k}$ is C^{k+l-1} at 1.

Frequently we fix the parameter $l := \lfloor \frac{n}{2} \rfloor + k + 1$, where n is the space dimension we are working in, and a constant $c > 0$ to fix the support. By the properties stated above, the radial function $\Psi(x) := \psi_{l,k}(c\|x\|)$ is then a C^{2k} function with $\mathrm{supp}(\Psi) = \mathcal{B}^d(0, c^{-1}) \subset \mathbb{R}^n$, where $\mathcal{B}^n(0, c^{-1})$ is the closed n-dimensional ball around the origin with radius c^{-1}.

4 ALGORITHM

We represent d-degree polynomials $\sum_{i=0}^{d} a_i t^i$ as a list of coefficients $(a_0, a_1, \ldots, a_d)$. Our implementation uses *Python* with *List* objects. Addition and multiplication of polynomials of this form are easily implemented as:

$$\sum_{i=0}^{d_1} a_i t^i + \sum_{j=0}^{d_2} b_j t^j = \sum_{i=0}^{\max\{d_1,d_2\}} (a_i + b_i) t^i,$$

where $a_i = 0$ for $i > d_1$ and $b_j = 0$ for $j > d_2$. Multiplication is given by

$$\left(\sum_{i=0}^{d_1} a_i t^i\right)\left(\sum_{j=0}^{d_2} b_j t^j\right) = \sum_{i=0}^{d_1+d_2} c_i t^i,$$

where

$$c_i = \sum_{k+j=i} a_k b_j.$$

An antiderivative of a polynomial is given by

$$(a_0, a_1, a_2, \ldots, a_d) \mapsto (0, a_0, \frac{a_1}{2}, \frac{a_2}{3}, \ldots, \frac{a_d}{d+1}),$$

corresponding to

$$\int \sum_{i=0}^{d} a_i t^i dt = \sum_{i=0}^{d} \frac{a_i}{i+1} t^{i+1},$$

and differentiation by

$$(a_0, a_1, \ldots, a_d) \mapsto (a_1, 2a_2, 3a_3, \ldots, da_d).$$

In order to maximize exact calculations up to computer limitations, we store the coefficients as tuples of Integers, numerator and denominator, avoiding the floating point approximation. Specifically, we used the *Rational* class provided in Python. We can represent polynomials in two variables as a polynomial in the first variable where each coefficient is a polynomial in the second variable (of which each coefficient is a rational number).

4.1 Construction

To calculate a polynomial representing the Wendland function $\psi_{l,k}$ on the interval $[0,1]$ we start by fixing the derivative

$$p'(t) = (1-t)^l t (t^2 - r^2)^{k-1},$$

see (7), which we represent as a polynomial in t with each coefficient a polynomial in r. We integrate with respect to t and obtain a new polynomial $p(t)$ in t, again with coefficients that are polynomials in r. We evaluate the polynomial p at $t = 1$ and at $t = r$, which in both cases result in a polynomial in r, and we obtain the polynomial $\psi(r) = p(1) - p(r)$. Note that $\psi(r) = C_1 \psi_{l,k}(r)$ for some constant $C_1 \neq 0$.

By using a long division algorithm we factor ψ into the form

$$\psi(r) = C_2 (1-r)^{l+k} p_{l,k}(r) \tag{8}$$

such that $p_{l,k}(r)$ is a polynomial with integer coefficients with no common factors. This is possible since $\psi_{l,0}$ has a zero of order l at 1, and by using the recursive relation in equation (4), we see that $\psi_{l,k}$ has a zero of order $l+k$ at 1. Since $\psi_{l,k}$ is essentially only defined up to a multiplicative non-zero constant, we discard the constant C_2 and use $\psi(r) = (1-r)^{l+k} p_{l,k}(r)$, a polynomial with integer coefficients, as a starting point for our recursion.

Using the relation in (5), ignoring the constant $C_{l,k}$, we see that

$$\begin{aligned}\psi_{l,k-1}(r) &= \frac{\frac{d}{dr}\left[(1-r)^{l+k} p_{l,k}(r)\right]}{r} \qquad (9)\\ &= \frac{1}{r}(1-r)^{l+k-1}((1-r)p'_{l,k}(r) - (l+k)p_{l,k}(r)).\end{aligned}$$

Therefore we have

$$\begin{aligned}p_{l,k-1}(r) &:= \frac{\psi_{l,k-1}(r)}{(1-r)^{l+k-1}} \qquad (10)\\ &= \frac{1}{r}\left[(1-r)p'_{l,k}(r) - (l+k)p_{l,k}(r)\right].\end{aligned}$$

We know that $\psi_{l,k-1}$ is a polynomial, therefore $\frac{d}{dr}\left[(1-r)^{l+k} p_{l,k}(r)\right]$ must by divisible by the monomial r. Since $(1-r)^{l+k-1}$ is not divisible by r, the right-hand side of (10) must be a polynomial in r. Therefore $p_{l,k-1}$ is a well defined polynomial.

By pulling out the common factor $b_{k-1} \in \mathbb{Z}$ of the coefficients in $p_{l,k-1}$ we obtain a new polynomial $\hat{p}_{l,k-1}$ and a constant b_{k-1} such that

$$p_{l,k-1} = b_{k-1} \hat{p}_{l,k-1}.$$

Repeating this step, until we arrive at $p_{l,0}$, we get a collection of polynomials in the form

$$\psi_i(r) = b_1 \cdots b_i (1-r)^{l+k-i} \hat{p}_{l,k-i}(r),\ i = 1,2,\ldots,k \tag{11}$$

where each of the polynomials $\hat{p}_{l,k-i}(r)$ has integer coefficients and each of the constants b_i is a negative integer.

The above list follows the notation in (Giesl, 2007) were ψ_0 is the polynomial given in (8) and is equal to the Wendland function $\psi_{l,k}$, and $\psi_1, \ldots, \psi_i$ are the Wendland functions given by $\psi_{l,k-1}, \ldots, \psi_{l,k-i}$ respectively, see equation (4). It is important to keep track of the constants $b_1, \ldots, b_i$ in (11) as they are necessary for correct evaluation of formula (2).

4.2 Example

To see how the algorithm works let us consider how it computes the Wendland function $\psi_{l,k}$ for $l=5$ and $k=4$. Here $p'(t)=(1-t)^5 t(t^2-r^2)^3$ and we obtain

$$\begin{aligned}\psi(r) &= \int_r^1 (1-t)^{5+4} t(t^2-r^2)^3 dt\\ &= -\frac{16}{3003}r^{13}+\frac{1}{24}r^{12}-\frac{32}{231}r^{11}+\frac{1}{4}r^{10}-\frac{16}{63}r^9\\ &+\frac{1}{8}r^8-\frac{1}{42}r^6+\frac{1}{168}r^4-\frac{1}{924}r^2+\frac{1}{10296}\\ &= \frac{1}{72072}(1-r)^9(384r^4+453r^3+237r^2+63r+7)\end{aligned}$$

and set $\psi_0(r)=(1-r)^9(384r^4+453r^3+237r^2+63r+7)$. For $r\in[0,1]$ we have the formulas (recall that $\psi_{l,k}(r)=0$ if $r\notin[0,1]$):

$$\begin{aligned}\psi_{5,4}(r) &= \psi_0(r)\\ &= (1-r)^9(384r^4+453r^3+237r^2+63r+7)\\ \psi_{5,3}(r) &= \psi_1(r) = \frac{\frac{d}{dr}\psi_0(r)}{r}\\ &= -156(1-r)^8(32r^3+25r^2+8r+1)\\ \psi_{5,2}(r) &= \psi_2(r) = \frac{\frac{d}{dr}\psi_1(r)}{r}\\ &= 3{,}432(1-r)^7(16r^2+7r+1)\\ \psi_{5,1}(r) &= \psi_3(r) = \frac{\frac{d}{dr}\psi_2(r)}{r}\\ &= -82{,}368(1-r)^6(6r+1)\\ \psi_{5,0}(r) &= \psi_4(r) = \frac{\frac{d}{dr}\psi_3(r)}{r}\\ &= 3{,}459{,}456(1-r)^5\end{aligned}$$

Note that we have, indeed, computed a lot more useful information than just a family of Wendland functions $\psi_{5,i}$, $i=0,1,\ldots,4$. In our algorithm, for a fixed l,k, we have

$$\begin{aligned}\psi_{l,k-j} = \psi_j(r) &= \frac{\frac{d}{dr}\psi_{j-1}(r)}{r}\\ &= \frac{\frac{d}{dr}\psi_{l,k-j+1}(r)}{r}, \quad \text{for } j=1,\ldots,k,\end{aligned}$$

and we have thus delivered all the radial basis functions needed for a collocation problem. This corresponds to computing a whole table as in (Giesl, 2007, Table 3.1), but for a collocation problem with arbitrary high derivatives. In the software tool, discussed in the next section, also the constant $c>0$ used to fix the support of the Wendland function, is included in these computations.

5 SOFTWARE LIBRARY

We have implemented the above algorithm in a software tool[1] that generates C/C++ code versions of the Wendland functions in factorized form. In a previous work (Bjornsson and Hafstein, 2018), we determined that the most efficient and accurate way to evaluate these Wendland functions was to use this factorized form. Evaluating these polynomials in fully expanded format using Horner's scheme (Burrus et al., 2003), can lead to very large numerical errors as shown in (Bjornsson and Hafstein, 2018). Below is a part of the library generated by our tool, which shows the family of Wendland functions obtained when starting with $\Psi_0(x)=\psi_{5,4}(c\|x\|)$, where $c>0$ is the constant that controls the support of the radial function Ψ.

Listing 1: Generated code for the $\psi_{5,4}$ family.

```
double __wendlandpsi_5_4_0(double x, ...
     double c){
    double t=__ipow((1.0-x),9);
    t=1.0*t*(((((384)*x + 453)*x + ...
         237)*x + 63)*x + 7);
    return t;
}
double __wendlandpsi_5_4_1(double x, ...
     double c){
    double t=__ipow((1.0-x),8);
    t=-156.0*t*__ipow(c,2)*((((32)*x ...
         + 25)*x + 8)*x + 1);
    return t;
}
double __wendlandpsi_5_4_2(double x, ...
     double c){
    double t=__ipow((1.0-x),7);
    t=3432.0*t*__ipow(c,4)*(((16)*x ...
         + 7)*x + 1);
    return t;
}
double __wendlandpsi_5_4_3(double x, ...
     double c){
    double t=__ipow((1.0-x),6);
    t=-82368.0*t*__ipow(c,6)*((6)*x ...
         + 1);
    return t;
}
double __wendlandpsi_5_4_4(double x, ...
     double c){
    double t=__ipow((1.0-x),5);
    t=3459456.0*t*__ipow(c,8)*(1);
    return t;
}
```

Note that `__wendlandpsi_5_4_j` corresponds to ψ_j in the example, but with $x=cr$ as argument.

When starting with $\Psi_0(x)=\psi_{5,3}(c\|x\|)$ instead, the relevant definitions are:

[1] The tool is available at https://gitlab.com/hjortur/wendland-function-generator/ with example outputs.

Listing 2: Generated code for the $\psi_{5,3}$ family.

```
double __wendlandpsi_5_3_0(double ...
    x, double c){
    double t=__ipow((1.0-x),8);
    t=1.0*t*((((32)*x + 25)*x + ...
        8)*x + 1);
    return t;
}
double __wendlandpsi_5_3_1(double ...
    x, double c){
    double t=__ipow((1.0-x),7);
    t=-22.0*t*__ipow(c,2)*(((16)*x ...
        + 7)*x + 1);
    return t;
}
double __wendlandpsi_5_3_2(double ...
    x, double c){
    double t=__ipow((1.0-x),6);
    t=528.0*t*__ipow(c,4)*((6)*x ...
        + 1);
    return t;
}
double __wendlandpsi_5_3_3(double ...
    x, double c){
    double t=__ipow((1.0-x),5);
    t=-22176.0*t*__ipow(c,6)*(1);
    return t;
}
```

Note that the polynomials `__wendlandpsi_5_3_1` and `__wendlandpsi_5_4_2` differ only by a multiplication of a constant and a power of c, and both polynomials are a representative of the Wendland function $\psi_{5,2}$.

The function `__ipow(x,i)` evaluates x^i where x is a double and i is a positive integer. A possible efficient implementation is given by:

Listing 3: Exponentiation routine.

```
//Function for fast squaring to ...
    integer power
//Posted by user Elias Yarrkov on ...
    Stackoverflow.
static double __ipow(double base, ...
    int exp){
    double result = 1.0;
    for (;;){
        if (exp & 1)
            result *= base;
        exp >>= 1;
        if (!exp)
            break;
        base *= base;
    }
    return result;
}
```

The functions `__wendlandpsi_x_y_z` have been “flattened” in the sense that their domain is $[0,1]$. They require the user to premultiply the x value with the chosen RBF-constant $c > 0$, that is for $\Psi(x) = \psi_{l,k}(c\|x\|)$, the user needs to pass in the value $c\|x\|$ and c after ensuring that $c\|x\| \in [0,1]$. A possible implementation using the Armadillo library (Sanderson, 2010) could, for example, be:

Listing 4: Example usage.

```
double psi3(const arma::vec &x, ...
    double c){
    double cx=c*arma::norm(x,2);
    return ( cx < 1.0 ) ? ...
        __wendlandpsi_5_4_3(cx,c) ...
        : 0.0;
}
```

The tool is a simple Python script named *wendlandfunctions.py*. When the script is run it outputs text for code- and header-files, which contain the Wendland function definitions. The user can supply the script with a parameter `--l` and an integer value $m \geq 2$, in order to output code for Wendland functions from $\psi_{2,1}$ up to $\psi_{m,i}$ for all $0 \leq i < m$.

5.1 Performance

In previous work (Bjornsson and Hafstein, 2018) we have compared different method to evalute these Wendland functions. The methods used for point evaluation were:

- Having them in factorized form, as our software tool provides, see Listing 1.
- Fully expanded polynomials and evaluated using Horners Scheme, as in (Argaez et al., 2017).
- Precomputing the function in high precision (see below) at 10^7 evenly spaced points on the interval $[0,1]$ and using them as a lookup table. That is, look up the closes value.
- Using the same lookup table but additionally linearly interpolate between two nearest neighbours to improve accuracy.

Figure 1 shows the performance of evaluating $\psi_{7,2}$ at 10^7 different points on the interval $[0,1]$, and Figure 2 shows the highest relative error at these points, with these four different methods. To estimate the relative error we calculated the value of the polynomial $\psi_{7,2}$ in *Matlab* using variable precision arithmetic (VPA), up to 32 significant digits, then rounded the value to the closest double precision floating point number. Note that the factorized form and Lookup-table are close in speed, but the factorized form gives much greater accuracy.

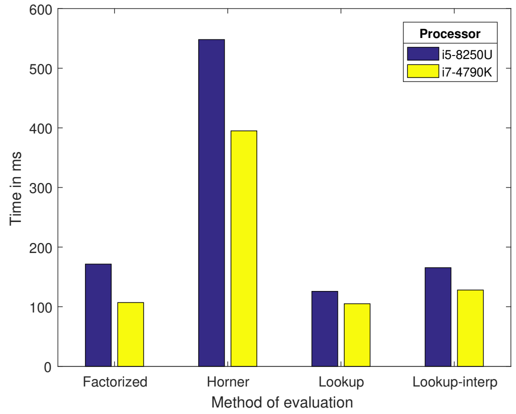


Figure 1: Running time of evaluating $\psi_{7,2}$ at 10^7 points.

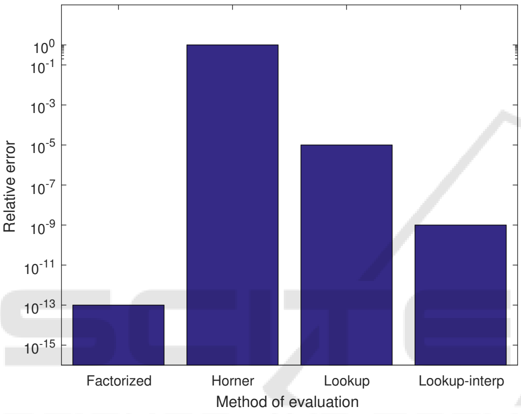


Figure 2: Relative error of evaluations of $\psi_{7,2}$. Note the logarithmic scale of the y-axis.

6 CONCLUSION

We have developed an algorithm and created a tool to generate a C/C++ library for Wendland's compactly supported Radial Basis Functions with arbitrary parameters in a factorized form. This allows for the efficient and numerically accurate evaluation of these functions. This is desirable since previously they were generated in a non-optimal form or had to be evaluated by hand, which is a tedious and error prone process. Additionally, the software generates a whole family of Wendland functions suitable for solving collocation problems for each initial Wendland function $\psi_{l,k}$ and its support radius c^{-1}. The tool takes less than a second to output the *.c* and *.h* files for all the Wendland function families up to and including order 8.

ACKNOWLEDGEMENT

This research was supported by the Icelandic Research Fund (Rannís) in grant number 152429-051, Lyapunov Methods and Stochastic Stability.

REFERENCES

Argaez, C., Hafstein, S., and Giesl, P. (2017). Wendland functions - a C++ code to compute them. In *Proceedings of the 7th International Conference on Simulation and Modeling Methodologies, Technologies and Applications - Volume 1: SIMULTECH,*, pages 323–330. INSTICC, SciTePress.

Bjornsson, H. and Hafstein, S. (2018). Verification of a numerical solution to a collocation problem. In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics - Volume 1: CTDE,*, pages 587–594. INSTICC, SciTePress.

Bjornsson, H., Hafstein, S., Giesl, P., Gudmundsson, S., and Scalas, E. (2019). Computation of the stochastic basin of attraction by rigorous construction of a Lyapunov function. *Discrete and Continuous Dynamical Systems-Series B (to appear)*.

Buhmann, M. (2000). Radial basis functions. In *Acta numerica, 2000*, volume 9 of *Acta Numer.*, pages 1–38. Cambridge Univ. Press, Cambridge.

Buhmann, M. (2003). *Radial basis functions: theory and implementations*, volume 12 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge.

Burrus, C. S., Fox, J. W., Sitton, G. A., and Treitel, S. (2003). Horner's method for evaluating and deflating polynomials. *DSP Software Notes, Rice University, Nov*, 26.

Floater, M. and Iske, A. (1996). Multistep scattered data interpolation using compactly supported Radial Basis Functions. *J. Comput. Appl. Math.*, 73(1-2):65–78.

Franke, C. and Schaback, R. (1998). Solving partial differential equations by collocation using radial basis functions. *Appl. Math. Comput.*, 93(1):73–82.

Giesl, P. (2007). *Construction of Global Lyapunov Functions Using Radial Basis Functions*, volume 1904 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin.

Giesl, P. (2008). Construction of a local and global Lyapunov function using radial basis functions. *IMA J. Appl. Math.*, 73(5):782–802.

Giesl, P. and Hafstein, S. (2015). Computation and verification of Lyapunov functions. *SIAM Journal on Applied Dynamical Systems*, 14(4):1663–1698.

Hafstein, S., Gudmundsson, S., Giesl, P., and Scalas, E. (2018). Lyapunov function computation for autonomous linear stochastic differential equations using sum-of-squares programming. *Discrete and Continuous Dynamical Systems - Series B*, 23(2):939–950.

Khalil, H. (2002). *Nonlinear systems*. Pear, 3. edition.

Khasminskii, R. (2012). *Stochastic stability of differential equations*. Springer, 2nd edition.

Mao, X. (2008). *Stochastic Differential Equations and Applications*. Woodhead Publishing, 2nd edition.

Sanderson, C. (2010). Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments. Technical report, NICTA.

Sastry, S. (1999). *Nonlinear Systems: Analysis, Stability, and Control*. Springer.

Vidyasagar, M. (2002). *Nonlinear System Analysis*. Classics in applied mathematics. SIAM, 2. edition.

Wendland, H. (1998). Error estimates for interpolation by compactly supported Radial Basis Functions of minimal degree. *J. Approx. Theory*, 93:258–272.

Wendland, H. (2005). *Scattered data approximation*, volume 17 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge.

Wendland, H. (2017). Multiscale radial basis functions. In *Frames and other bases in abstract and function spaces*, Appl. Numer. Harmon. Anal., pages 265–299. Birkhäuser/Springer, Cham.

Wu, Z. (1992). Hermite-Birkhoff interpolation of scattered data by radial basis functions. *Approx. Theory Appl.*, 8(2):1–10.

Paper VI

Verification of a Numerical Solution to a Collocation Problem.

Bjornsson, H. and Hafstein, S.

In Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO)

Verification of a Numerical Solution to a Collocation Problem

Hjortur Bjornsson and Sigurdur Hafstein
Science Institute and Faculty of Physical Sciences, University of Iceland, Dunhagi 5, 107 Reykjavík, Iceland

Keywords: Numerical Computation, Lyapunov Function, Radial Basis Functions.

Abstract: In a recent method to compute Lyapunov functions for nonlinear stochastic differential equations a subsequent verification of the results is needed. The theory has been developed but there are several practical difficulties in its implementation because of the huge amount of function evaluations needed during verification. We study several different methods and compare their accuracy and efficiency.

1 INTRODUCTION

We will discuss numerical solutions to Partial Differential Equations (PDE) that arise when computing Lyapunov functions for Stochastic Differential Equations (SDE) and, in particular, how the validity of the computed Lyapunov functions can be verified numerically. In a novel numerical method (Bjornsson et al., 2018) we obtain a numerical solution to a PDE, and that solution is supposed to be a Lyapunov function for a certain SDE. To guarantee that the numerical solution is in fact a Lyapunov function, we have an error estimate which states that if the value of the numerical solution on a certain grid of points is lower then some constant, then the numerical solution is indeed a Lyapunov function for the system. The theory supporting this novel method was developed in (Gudmundsson and Hafstein, 2018; Hafstein et al., 2018), and in (Bjornsson et al., 2018) the method is developed and it is shown that it converges to a true Lyapunov function if the collocation grid used for the numerical solution of the PDE is sufficiently dense. However, one must verify a posteriori on an evaluation grid that the collocation grid was indeed adequate. An issue with the method is that the evaluation grid is so dense that we need to evaluate the computed Lyapunov function at typically 10^9, and even up to 10^{16}, points. Note that the Lyapunov function is computed using Radial Basis Functions (RBF) and to evaluate it at a point, one must sum over all the RBFs used in the computation, i.e. the sum contains a number of terms that is equal to the number of the collocation points used. Here we will compare the numerical errors and performances of various methods used for this nontrivial and involved evaluation.

2 BACKGROUND

For completeness we give a quick background with many of the details omitted. For full details see (Gudmundsson and Hafstein, 2018; Hafstein et al., 2018; Bjornsson et al., 2018). We consider a d-dimensional SDE of the form

$$dX(t) = f(X(t))dt + g(X(t))dW(t), \qquad (1)$$

where $f : \mathbb{R}^d \to \mathbb{R}^d$, $g : \mathbb{R}^d \to \mathbb{R}^{d\times Q}$, $f(0) = g(0) = 0$, and $W(t)$ is a Q-dimensional Brownian motion. We are specifically interested in the stability of the trivial solution $X = 0$ of the system.

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with a smooth boundary $\Gamma = \partial\Omega$. We solve numerically the boundary problem of the PDE given by:

$$\begin{cases} LV(x) = r(x) & \text{for } x \in \Omega, \\ V(x) = c(x) & \text{for } x \in \Gamma, \end{cases} \qquad (2)$$

where L denotes the following differential operator associated with the system given in equation (1):

$$LV(x) = \frac{1}{2}\sum_{i,j=1}^{d} m_{ij}(x)\frac{\partial^2 v}{\partial x_i \partial x_j}(x) + \sum_{i=1}^{d} f_i(x)\frac{\partial v}{\partial x_i}(x), \quad (3)$$

where $(m_{ij}(x))_{i,j} = g(x)g(x)^\top$. For suitable functions $r(x)$ and $c(x)$ the solution to this PDE will be a Lyapunov function asserting the asymptotic stability in probability of the trivial solution and we can use it to estimate its probabilistic basin of attraction.

To solve this PDE numerically we use the RBF method similar to (Giesl, 2007; Giesl, 2008; Giesl and Wendland, 2007), where Lyapunov functions are computed for deterministic ordinary differential equations (ODEs), but adapted to SDEs. Given a set

Bjornsson, H. and Hafstein, S.
Verification of a Numerical Solution to a Collocation Problem.
DOI: 10.5220/0006945405870594
In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2018) - Volume 1*, pages 587-594
ISBN: 978-989-758-321-6

of points $X_1 = \{x_1, \ldots, x_N\} \subset \Omega \subset \mathbb{R}^d$ and $X_2 = \{\xi_1, \ldots, \xi_M\} \subset \Gamma$, we solve the interpolation problem

$$\begin{cases} LV(x_i) = r(x_i) & \text{for all } i = 1, \ldots, N, \\ V(\xi_i) = c(\xi_i) & \text{for all } i = 1, \ldots, M. \end{cases} \tag{4}$$

The solution to the interpolation problem is given by

$$\begin{aligned} V(x) = & \sum_{k=1}^{N} \alpha_k (\delta_{x_k} \circ L)^y \psi(\|x-y\|) \\ & + \sum_{k=1}^{M} \alpha_{N+k} (\delta_{\xi_k} \circ L^0)^y \psi(\|x-y\|) \end{aligned} \tag{5}$$

where $\delta_y V(x) = V(y)$ and the superscript y denotes that the operator is applied with respect to the variable y, $L^0 = id$ and $\psi = \psi_{l,k}$ is a so-called Wendland function, which are compactly supported RBFs (Wendland, 1998). The constants α_k in equation (5) can be determined as the solution to a certain linear system $A\alpha = \gamma$, where the matrix A is symmetric and positive definite; see (Bjornsson et al., 2018) for full details on the matrix A and the vector γ.

2.1 Lyapunov Function

The method described in the preceding section is used to compute a certain Lyapunov function for the system (1), whose domain does not include the equilibrium at the origin, hence the name *non-local Lyapunov function*. Again see (Gudmundsson and Hafstein, 2018; Hafstein et al., 2018; Bjornsson et al., 2018) for details.

Definition (Non-Local Lyapunov Function). Let $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^d$, $\mathcal{B} \subset \mathcal{A}^\circ$, be simply connected compact neighbourhoods of the origin with C^2 boundaries and set $\mathcal{U} := \mathcal{A} \setminus \mathcal{B}^\circ$. A function $V \in C^2(\mathcal{U})$ for the system (1) such that

(1) $0 \leq V(x) \leq 1$ for all $x \in \mathcal{U}$,

(2) $LV(x) < 0$ for all $x \in \mathcal{U}$, and

(3) $V^{-1}(0) = \partial\mathcal{B}$ and $V^{-1}(1) = \partial\mathcal{A}$,

is called a *non-local Lyapunov function* for the system (1), and we refer to $\partial\mathcal{B}$ and $\partial\mathcal{A}$ is the inner and outer boundary of $\mathcal{U}$, respectively.

To compute a non-local Lyapunov function for the system (1) we look for solutions of the PDE (2) with $r(x) = -h$ where $h > 0$ is a small constant, $c(x) = 1$ for $x \in \partial\mathcal{A}$ and $c(x) = 0$ for $x \in \partial\mathcal{B}$. Because we compute a numerical approximation to the solution of the system we can not expect $LV(x) = -h$ to hold for all $x \in \mathcal{U}$, but since the definition of a non-local Lyapunov function only requires $LV(x) < 0$, our numerical approximation will still be a true Lyapunov function as long as there is no point in $x \in \mathcal{U}$ with $LV(x) \geq 0$. In this paper we are concerned with the numerical verification of the condition that $LV(x) < 0$ holds for all $x \in \mathcal{U}$. The theory for this verification is developed in (Bjornsson et al., 2018). Here we are interested in the nontrivial technical details of its efficient implementation.

2.2 Explicit Formulas

For the explicit computations of our Lyapunov function, one must choose a specific Wendland function $\psi_{k,l}(x)$, where the indices k, l are non-negative integers. In the one dimensional case we have used $\psi_{7,6}$ and for two dimensional cases we used $\psi_{8,6}$. Note that there are certain restriction on the indices of the Wendland function for the problem at hand and in our computations we need at least these rather large indices.

For example, assuming we are using the Wendland function $\psi_{7,6}$ and the RBF constant $c > 0$, we define by some abuse of notation $\psi_0(r) := \psi_{7,6}(cr)$ for $r \geq 0$. Then we define $\psi_{i+1}(r) = \frac{1}{r}\frac{\partial}{\partial r}\psi_i(r)$ recursively for $i = 0, 1, 2, 3$. Finally, we define $W_i(x) = \psi_i(x/c)$. The formulas for the functions W_i starting with $\psi_0 := \psi_{7,6}$ are the following:

$$\begin{aligned} W_0(x) = & (1-x)^{13}(4096x^6 + 7059x^5 + 5751x^4 \\ & + 2782x^3 + 830x^2 + 143x + 11), \end{aligned} \tag{6}$$

$$\begin{aligned} W_1(x) = & -38c^2(1-x)^{12}(2048x^5 + 2697x^4 \\ & + 1644x^3 + 566x^2 + 108x + 9), \end{aligned} \tag{7}$$

$$\begin{aligned} W_2(x) = & \, 10336c^4(1-x)^{11}(128x^4 \\ & + 121x^3 + 51x^2 + 11x + 1), \end{aligned} \tag{8}$$

$$\begin{aligned} W_3(x) = & -62016c^6(1-x)^{10}(320x^3 \\ & + 197x^2 + 50x + 5), \text{ and} \end{aligned} \tag{9}$$

$$W_4(x) = 3224832c^8(1-x)^9(80x^2 + 27x + 3). \tag{10}$$

Note that these formulas are only valid for $0 \leq x \leq 1$ and for $x > 1$ we have $W_i(x) = 0$. The parameter c is referred to as the RBF constant and is used to control the size of the support of the functions $\mathbb{R}^d \to \mathbb{R}^d$, $x \mapsto \psi_i(\|x\|)$, i.e. the support is a ball of radius $1/c$ around the origin.

The explicit formulas for $V(x)$ and $LV(x)$ from our numerical solution to the interpolation problem (4), and by writing $\beta = x - x_k$ for brevity, are given by the following expressions:

$$\begin{aligned} V(x) = \sum_{k=1}^{N} \alpha_k \Big[& -\psi_1(\|\beta\|)\langle \beta, f(x_k)\rangle \\ & + \frac{1}{2}\sum_{i,j=1}^{d} m_{ij}[\psi_2(\|\beta\|)\beta_i\beta_j \\ & + \delta_{ij}\psi_1(\|\beta\|)]\Big] \\ & + \sum_{k=1}^{M} \alpha_{N+k}\psi_0(\|\beta\|) \end{aligned} \tag{11}$$

and

$$\begin{aligned} & LV(x) \\ & = \sum_{k=1}^{N} \alpha_k \Big[-\psi_2(\|\beta\|)\langle \beta, f(x)\rangle\langle \beta, f(x_k)\rangle \\ & \quad -\psi_1(\|\beta\|)\langle (f(x), f(x_k))\rangle \\ & \quad + \frac{1}{2}\sum_{i,j=1}^{d} m_{ij}(x_k)\Big[\psi_3(\|\beta\|)\langle \beta, f(x)\rangle\beta_i\beta_j \\ & \qquad + \psi_2(\|\beta\|)f_j(x)\beta_i \\ & \qquad + \psi_2(\|\beta\|)f_i(x)\beta_j + \delta_{ij}\psi_2(\|\beta\|)\langle \beta, f(x)\rangle\Big] \\ & \quad + \frac{1}{2}\sum_{i,j=1}^{d} m_{ij}(x)\Big[-\psi_3(\|\beta\|)\langle \beta, f(x_k)\rangle\beta_i\beta_j \\ & \qquad - \psi_2(\|\beta\|)f_j(x_k)\beta_i \\ & \qquad - \psi_2(\|\beta\|)f_i(x_k)\beta_j - \delta_{ij}\psi_2(\|\beta\|)\langle \beta, f(x_k)\rangle\Big] \\ & \quad + \frac{1}{4}\sum_{r,s=1}^{d}\sum_{i,j=1}^{d} m_{rs}(x)m_{ij}(x_k)\Big[\psi_4(\|\beta\|)\beta_i\beta_j\beta_r\beta_s \\ & \qquad + \psi_3(\|\beta\|)[\delta_{ij}\beta_r\beta_s + \delta_{ir}\beta_j\beta_s \\ & \qquad + \delta_{is}\beta_j\beta_r + \delta_{jr}\beta_i\beta_s + \delta_{js}\beta_i\beta_s + \delta_{rs}\beta_i\beta_j] \\ & \qquad + \psi_2(\|\beta\|)[\delta_{ij}\delta_{rs} + \delta_{ir}\delta_{js} + \delta_{is}\delta_{jr}]\Big]\Big] \\ & + \sum_{k=1}^{M} \alpha_{N+k}\Big[-\psi_1(\|\xi_k - x\|)\langle \xi_k - x, f(x)\rangle \\ & \quad + \frac{1}{2}\sum_{i,j=1}^{d} m_{ij}(x)[\psi_2(\|\xi_k - x\|)(\xi_k - x)_i(\xi_k - x)_j \\ & \qquad + \delta_{ij}\psi_1(\|\xi_k - x\|)]\Big]. \end{aligned} \tag{12}$$

In these formulas $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{N+M})^\top$ is the solution to $A\alpha = \gamma$ associated with the interpolation problem (4) and β_i is the i-th component of the vector $\beta = x - x_k$. Note the above formulas are independent of which Wendland function $\psi_{k,l}$ we start with in the beginning.

3 VERIFICATION

Let $\mathcal{A}, \mathcal{B}$, and $\mathcal{U}$ be as in the definition of non-local Lyapunov functions and let $V(x)$ be a numerical approximation like is described afterwards. Now by (Bjornsson et al., 2018, Theorem 4.3), if

$$\nu := \max_{y \in Y_{\mathcal{U}}} LV(y) + C_u \frac{d^2}{4} h^2 < 0, \tag{13}$$

then V is a non-local Lyapunov function for the system. Here d is the dimension of the system, $h > 0$ is a parameter controlling the density of the evaluation grid, and $Y_{\mathcal{U}}$ is the evaluation grid that covers $\mathcal{U}$. Finally the constant $C_{\mathcal{U}}$ is an upper estimate on the second derivatives of our function LV, for further details see (Bjornsson et al., 2018; Mohammed and Giesl, pted).

3.1 One Dimensional Example

For an explicit example let us consider the one dimensional SDE from (Bjornsson et al., 2018)

$$dx(t) = \sin(x(t))\,dt + \frac{3x(t)}{1 + x(t)^2}\,dW(t), \tag{14}$$

where W is a one dimensional Brownian motion. We determine an approximate solution to the PDE:

$$\begin{cases} LV(x) & = -10^{-3} \text{ for } 10^{-2} < x < 8, \\ V(x) & = 0 \text{ for } x = 10^{-1}, \\ V(x) & = 1 \text{ for } x = 8. \end{cases} \tag{15}$$

The approximate solution was determined using the Wendland function $\psi_{7,6}$, the RBF constant $c = 2$, and 700 evenly spaced collocation points on the interval $[1.1 \cdot 10^{-2}, 7.99]$. Figure 1 shows the numerical solution to PDE (15), for the system in equation (14). For this system and these values the estimate $C_{\mathcal{U}}$ is equal to $1.6846 \cdot 10^{12}$.

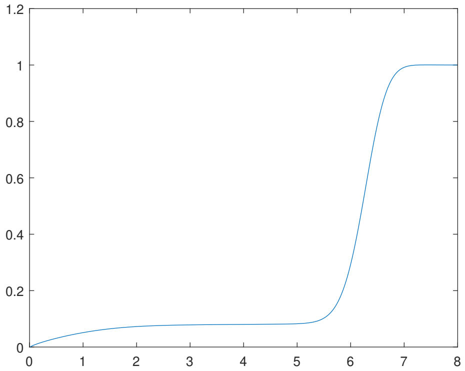


Figure 1: Numerical solution of $V(x)$ in equation (15).

By estimating through rough computations that the maximum value of our numerical approximation to $LV(x)$ is 0.3×10^{-3}, we get by setting $\nu = 0$ in equation (13)

$$h = \sqrt{\frac{4 \cdot 0.3 \cdot 10^{-3}}{1.6846}} = 2.6 \cdot \times 10^{-8}.$$

To give us some safety, which is needed since ν has to be strictly negative, we choose $h = 2.18 \cdot 10^{-8}$. This corresponds to evaluating the function LV at $7.5 \cdot 10^{8}$ evenly spaced points $Y_{\mathcal{U}}$ on the interval $[10^{-2} - h, 8 + h]$. This large number of points that needs to be evaluated takes about 10 seconds on a normal desktop computer (i7 4790k). The maximum value of LV found on this grid is $-0.281 \cdot 10^{-3}$ and thus

$$\nu = -0.281 \cdot 10^{-3} + 0.19119 \cdot 10^{-3} < 0,$$

so our approximation is a non-local Lyapunov function.

3.2 Two Dimensional Example

For a second explicit example consider the two dimensional system from (Grüne and Camilli, 2003) also studied in (Bjornsson et al., 2018), given by

$$dx(t) = (M + \rho(x(t))I)x(t)\,dt + g(x(t))\,dW(t), \quad (16)$$

where W is a one dimensional Brownian motion, I the 2×2-identity matrix,

$$M = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \ \rho(x) = \|x\| - 1,$$

and

$$g(x) = \|x\| \left(\|x\| - \frac{1}{2} \right) \left(\|x\| - \frac{3}{2} \right) x.$$

We find an approximate solution to the PDE, $LV(x) = -10^{-2}$, on an annulus around the origin with inner radius 0.4 and outer radius 1.9 with $V(x) = 0$ for $\|x\| = 0.4$ and $V(x) = 1$ for $\|x\| = 1.9$. To calculate the approximation we used the Wendland function $\psi_{8,6}$ and a grid of 80×80 points evenly spaced on the annulus. Figure 2 shows the resulting numerical approximation for the system in equation (16). For this system, and using $\psi_{8,6}$, the constant $C_{\mathcal{U}}$ is determined to be $4.3220 \cdot 10^{12}$, and following similar calculations as in the preceding section we estimate the maximum value of LV to be ≈ -0.005, this gives us $h = 3.4013 \cdot 10^{-8}$, so we need to evaluate LV on a grid with $(1.1760 \cdot 10^{8})^2 \approx 10^{16}$ points.

3.3 Comparison of Methods

In the inner-most loop of our program we have to evaluate W_i, $i = 1,2,3,4$. For simplicity we consider only the polynomials resulting from the Wendland function $\psi_{7,6}$ given by equations (7)-(10) and we

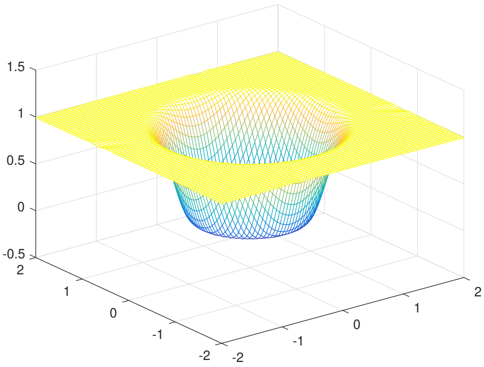


Figure 2: Numerical solution of $V(x)$ for the two dimensional system (16).

fixed $c = 2$; the function (6) is unused as LV does not depend on it. We tried different Wendland functions but the results were comparable. Fast evaluation of these functions is critical for the performance of our verification, therefore we tested 5 different evaluation methods: having these functions hardcoded in factorized form as in (7)-(10), expanding the polynomials and using Horner's method for the evaluation, using Lookup-tables, using a Lookup-table and additionally applying linear interpolation, and combining the two previous approaches on different subintervals. All tests were written in C and compiled using *gcc* with optimization flag *-O2*. The figures in the following section only show function (10) since it has the largest numerical errors. The errors for the other functions, i.e. (7)-(9), are qualitatively identical but of lower magnitude.

3.3.1 Evaluation with Hardcoded Functions

We hardcoded the polynomials as they are written in equations (7)-(10).

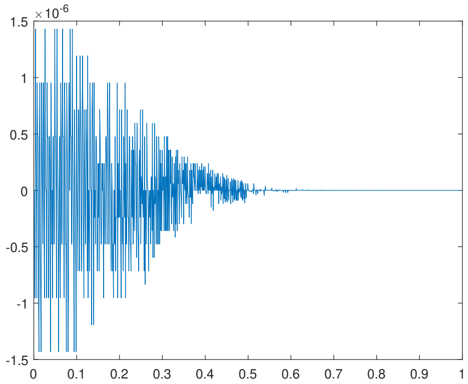


Figure 3: Absolute error as a function of x of W_4 using Hardcoded Functions. Note the scale on the y-axis is $\times 10^{-6}$.

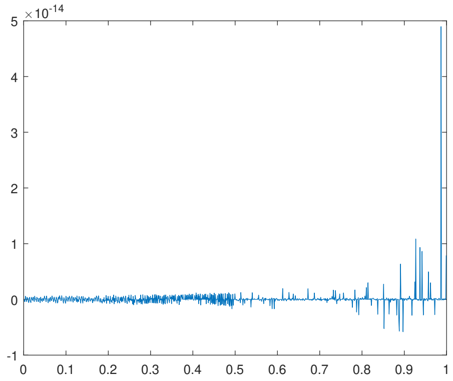


Figure 4: Relative error as a function of x of W_4 using Hardcoded Functions. Note the scale on the y-axis is $\times 10^{-14}$.

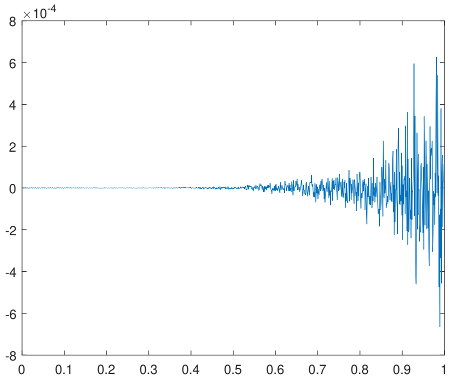


Figure 5: Absolute error as a function of x of W_4 using Horner's method. Note the scale on the y-axis is $\times 10^{-4}$.

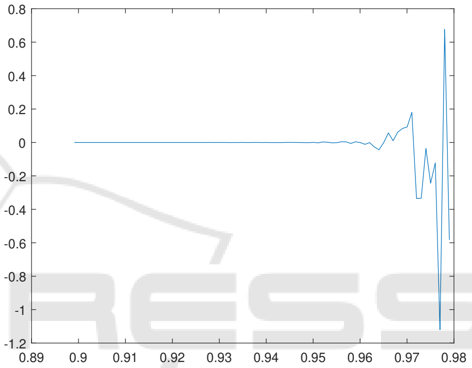


Figure 6: Relative error as a function of x of W_4 using Horner's method. Note the scale on the y-axis is $\times 10^{0}$.

Figures 3 and 4 show the absolute and relative error, respectively, of the function W_4 compared to the values obtained from infinite precision arithmetic truncated to 64-bit floating point values. These figures show us that this method, i.e. having hardcoded functions, gives us the best accuracy out of all of the methods tested.

3.3.2 Evaluation using Horner's Method

We expanded the polynomials, i.e. obtained coefficients $a_n, a_{n-1}, \ldots, a_0$ such that

$$W_i(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0.$$

Obviously the coefficients a_i depend on which Wendland function $\psi_{k,l}$ we started with. Having the polynomials in expanded form allows us to evaluate them at any point x using the following scheme (Horner's method):

```
Horner(x, [a_n])
    acc:=0;
    for(i=n;i>=0;i--)
        acc=acc*x;
        acc=acc+a_i;
    return acc;
```

By taking advantage of SIMD-instructions (Single Instruction, Multiple Data) we can evaluate two of these polynomials at a time in double precision arithmetic, or even all four at the same time on machines that support 256-bit wide SIMD registers (AVX2 or later).

Since the polynomial functions have a high order zero at $x = 1$, and the coefficients are relatively large, we get significant absolute errors in the evaluation when x is close to 1, see figure 5. For the function W_4 the relative error close to $x = 1$ explodes, as the value of the function is close to 0 there. Using higher values for the Wendland RBF constant c exaggerates this behaviour, so it is virtually impossible to use Horner's method to evaluate the polynomials with sufficient accuracy with $c = 10$ or higher.

Figure 6 shows us how the relative error of the Horner's method explodes the closer we get to $x = 1$.

3.3.3 Evaluation using Lookup-tables

We pre-evaluate the polynomials at $K = 10^7$ evenly spaced points, $x_0, \ldots, x_K$ between 0 and 1, using infinite precision arithmetic and then truncate and store the results. At runtime we evaluate $W_j(x)$ by finding i such that x_i is the closest value to x and return $W_j(x) \approx W_j(x_i)$. Here is a pseudo-code of the full procedure:

```
//j selects W_j
Lookuptable(x, j)
    i=round(x*(K-1));
    return W_j[i];
```

The tables are constructed in such a way that we can use the same index to get the values of all of the W_j functions, and furthermore by weaving the tables together we can evaluate all four of them by read-

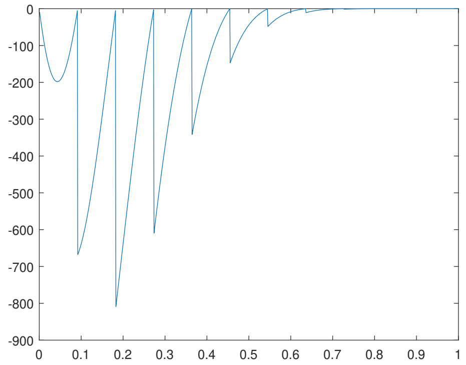


Figure 7: Absolute error as a function of x of W_4 using Lookup-table. Note the scale on the y-axis is $\times 10^0$.

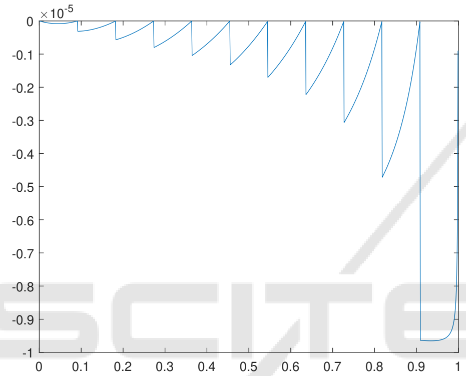


Figure 8: Relative error as a function of x of W_4 using Lookup-table. Note the scale on the y-axis is $\times 10^{-5}$.

ing twice from memory, or even once on AVX2 capable machines. This is a significant performance boost compared to the Horner's method, see table 1. As for the absolute error see figure 7. This kind of sawtooth shape is typical for Lookup-tables, however, note that the absolute error is high for x close to 0 but stabilizes when x gets closer 1. Since the true value of the function W_4 close to 0 is very high, this translates to a low relative error around 0, see figure 8.

3.3.4 Lookup-tables with Linear Interpolation

One improvement on the Lookup-table method described in the previous section is to use linear interpolation between the lookup values to get more accurate evaluations at the cost of some processing time. Here is a pseudo-code for the procedure:

```
Lookuptable_interpolate(x,j)
    i=floor(x*(K-1));
    interpolant:=(x-x_n[i])/  ...
       ...(x_n[i+1]-x_n[i]);
    return W_j[i] + ...
       ...interpolant*(W_j[i+1]-W_j[i]);
```

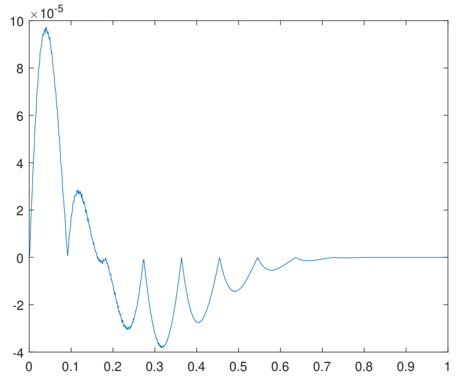


Figure 9: Absolute error as a function of x of W_4 using Lookup-table with interpolation. Note the scale on the y-axis is $\times 10^{-5}$.

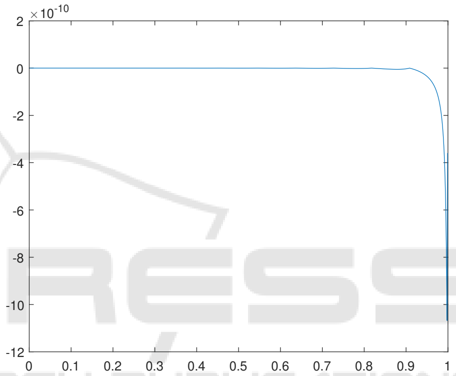


Figure 10: Relative error as a function of x of W_4 using Lookup-table with interpolation. Note the scale on the y-axis is $\times 10^{-10}$.

There is a significant increase in accuracy, see figure 9, but this method requires more memory access than the previous method.

3.3.5 Combination of Lookup-table Methods

As noted in the previous sections, the accuracy of the Lookup-table method is quite good when x is close to 1, therefore we propose the following method: we start by selecting a cutoff value b between 0 and 1, then when we evaluate $W_j(x)$ we use the simple lookup method if $x \geq b$, otherwise we use the Lookup-table with interpolation.

```
Lookuptable_combined(x,j)
    if(x>=b)
        return Lookuptable(x,j);
    else
        return Lookuptable_interpolate(x,j);
```

Note the large absolute error around the value $x = 0.8$ in figure 11 translates to a low relative error of

Table 1: Total execution time to evaluate the functions W_i, $i = 1,2,3,4$, at 10^7 different points, using a single thread.

Method / Processor	i5-8250U	i5-3210M	i5-2500k	i7-4790K
Horner's method	548.1ms	721ms	657.8ms	395ms
Simple Lookup-table	125.8ms	152.7ms	146.7ms	105ms
Lookup-table interp.	165.5ms	231.9ms	216.9ms	128ms
Lookup-table comb.	148.2ms	187.6ms	172.5ms	105ms
Hardcoded	171.5ms	214.8ms	199.8ms	107ms

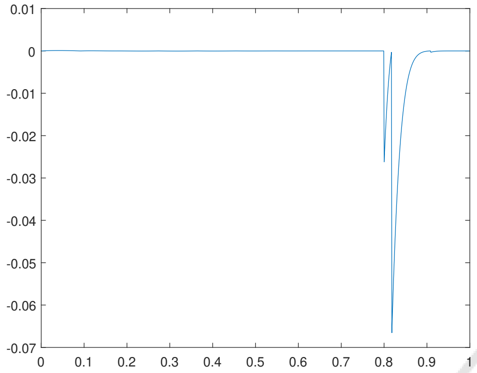


Figure 11: Absolute error as a function of x of W_4 using Lookup-table with $b = 0.8$. Note the scale on the y-axis is $\times 10^0$.

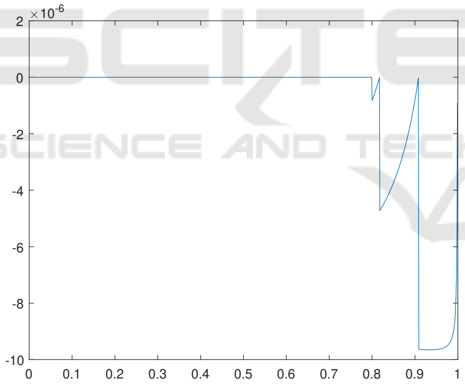


Figure 12: Relative error as a function of x of W_4 using Lookup-table with $b = 0.8$. Note the scale on the y-axis is $\times 10^{-6}$.

$8 \cdot 10^{-6}$, see figure 12. The value of $b = 0.8$ was empirically determined to give the best tradeoff between speed and accuracy.

4 CONCLUSIONS

We compared different methods to evaluate polynomials stemming from Wendland's compactly supported radial basis functions. In our application for rigidly verifying the negativity of Lyapunov functions computed for stochastic differential equations these polynomials have to be evaluated at numerous points and this has to be done with sufficient accuracy. Table 1 shows how long it takes to evaluate the polynomials W_1, W_2, W_3, and W_4 at 10^7 different points on different processors. The fastest method is to use a Lookup-table, but it is too inaccurate for practical use, at least in our application. Using linear interpolation between the lookup-values produced much more accurate results, but the method is considerably slower. A more efficient method that is sufficiently accurate is to use linear interpolation between the lookup-values in the most troublesome areas of the Lookup-table, and just use simple Lookup-table otherwise. Hardcoding the polynomials in factorized form as in equations (7)-(10) is both very fast, although not as fast as using a Lookup-table, and very accurate. Horner's method should be avoided since it produces inaccurate results and is slow. The inaccuracy is supposedly due to the multiple zero at 1 of the functions W_i.

ACKNOWLEDGEMENTS

The research for this paper was supported by the Icelandic Research Fund (Rannís) in the project 'Lyapunov Methods and Stochastic Stability' (152429-051), which is gratefully acknowledged.

REFERENCES

Bjornsson, H., Gudmundsson, S., Giesl, P., Hafstein, S., and Scalas, E. (2018). Computation of the stochastic basin of attraction by rigorous construction of a Lyapunov function. submitted.

Giesl, P. (2007). *Construction of Global Lyapunov Functions Using Radial Basis Functions*, volume 1904 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin.

Giesl, P. (2008). Construction of a local and global Lyapunov function using Radial Basis Functions. *IMA J. Appl. Math.*, 73(5):782–802.

Giesl, P. and Wendland, H. (2007). Meshless collocation: error estimates with application to dynamical systems. *SIAM J. Numer. Anal.*, 45(4):1723–1741.

Grüne, L. and Camilli, F. (2003). Characterizing attraction probabilities via the stochastic Zubov equation. *Discrete Contin. Dyn. Syst. Ser. B*, 3(3):457–468.

Gudmundsson, S. and Hafstein, S. (2018). Probabilistic basin of attraction and its estimation using two Lyapunov functions. *Complexity*, Article ID:2895658.

Hafstein, S., Gudmundsson, S., Giesl, P., and Scalas, E. (2018). Lyapunov function computation for autonomous linear stochastic differential equations using sum-of-squares programming. *Discrete Contin. Dyn. Syst. Ser. B*, 2(23):939–956.

Mohammed, N. and Giesl, P. (accepted). Grid refinement in the construction of Lyapunov functions using Radial Basis Functions. *Discrete Contin. Dyn. Syst. Ser. B*.

Wendland, H. (1998). Error estimates for interpolation by compactly supported Radial Basis Functions of minimal degree. *J. Approx. Theory*, 93:258–272.

References

Argáez, C., Hafstein, S., and Giesl, P. (2017). “Wendland Functions - A C++ Code to Compute Them.” *Proceedings of the 7th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, pp. 323–330.

Arnold, L. (2003). *Random Dynamical Systems*. 2. Springer Monographs in Mathematics. Springer.

Bjarkason, E. (2022). “Lyapunov Functions for Linear Stochastic Differential Equations by Bilinear Matrix Inequalities: Theory and Implementation.” MA thesis. University of Iceland.

Bjornsson, H., Giesl, P., et al. (2018). “Local Lyapunov Functions for Nonlinear Stochastic Differential Equations by Linearization.” *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pp. 579–586.

Bjornsson, H. and Hafstein, S. (2018a). “Dynamical Systems in Theoretical Perspective.” Ed. by J. Awrejcewicz. Vol. 248. Springer Proceedings in Mathematics and Statistics. Springer. Chap. Lyapunov functions for almost sure exponential stability, pp. 51–61.

– (2018b). “Verification of a Numerical Solution to a Collocation Problem.” *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pp. 587–594.

– (2019a). “Advanced algorithm for interpolation with Wendland functions.” *International Conference on Informatics in Control, Automation and Robotics*. Springer, pp. 99–117.

– (2019b). “Algorithm and Software to Generate Code for Wendland Functions in Factorized Form.” *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pp. 156–162.

Bjornsson, H., Hafstein, S., et al. (2019). “Computation of the stochastic basin of attraction by rigorous construction of a Lyapunov function.” *Discrete Contin. Dyn. Syst Ser. B* 24.8, pp. 4247–4269.

Burrus, C. S. et al. (2003). “Horner’s method for evaluating and deflating polynomials.” *DSP Software Notes, Rice University, Nov* 26.

Giesl, P. (2007a). *Construction of Global Lyapunov Functions Using Radial Basis Functions*. Lecture Notes in Math. 1904, Springer.

– (2007b). *Construction of Global Lyapunov Functions Using Radial Basis Functions*. Vol. 1904. Lecture Notes in Mathematics. Springer-Verlag, Berlin.

Gilbarg, D. and Trudinger, N. S. (2001). *Elliptic Partial Differential Equations of Second Order*. Springer- Classics in Mathematics.

Gudmundsson, S. and Hafstein, S. (2018). “Probabilistic basin of attraction and its estimation using two Lyapunov functions.” *Complexity* Article ID, p. 2895658.

Iske, A. (2018). *Approximation theory and algorithms for data analysis*. Springer.

Khasminskii, R. (2012). *Stochastic stability of differential equations*. 2nd. Springer.

Kloeden, Peter E et al. (1992). *Numerical Solution of Stochastic differential equations*. Springer.

Mao, X. (2008). *Stochastic Differential Equations and Applications*. 2nd. Woodhead Publishing.

Menshikov, Mikhail, Popov, Serguei, and Wade, Andrew (2016). *Non-homogeneous random walks: Lyapunov function methods for near-critical stochastic systems*. Vol. 209. Cambridge University Press.

Mohammed, N. and Giesl, P. (2015). "Grid refinement in the construction of Lyapunov functions using Radial Basis Functions." *Discrete Contin. Dyn. Syst. Ser. B*.

Øksendal, B. (2010). *Stochastic Differential Equations*. 6th. Springer.

Shaikhet, Leonid (2013). *Lyapunov functionals and stability of stochastic functional differential equations*. Springer Science & Business Media.

Wendland, H. (2005). *Scattered data approximation*. Vol. 17. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge, pp. x+336. ISBN: 978-0521-84335-5.

– (2017). "Multiscale radial basis functions." *Frames and other bases in abstract and function spaces*. Appl. Numer. Harmon. Anal. Birkhäuser/Springer, Cham, pp. 265–299.