Nonlinear stochastic dynamics and chaos by numerical path integration

Eirik Mo

January 5, 2008

Acknowledgements

The main gratitude goes to Professor Arvid Naess, who has been the advisor for this work, and given many valuable comments. The problem description was written by Naess, the development of path integration for stochastic differential equations is largely devoted to him, and he also applied for and received funding for the project.

For four years, I have worked together with Dr. Hans Christian Karlsen, who also was a PhD student advised by Naess. Karlsen's work on running path integration for some specific problems was of great importance for the development of more stable and efficient code. His validation of accuracy and general comments have been very helpful for this work.

From August 2004 until February 2005, I visited Professor Mircea Grigoriu at the Department of Structural Engineering at Cornell University, beautifully situated in Ithaca by the Finger Lakes, upstate New York. This was a wonderful stay, where I were warmly welcomed by PhD students at the department, and a lot of people in the area. I would especially like to mention Ms. (Kathleen) Misti Wilcox, who rented out rooms in her house, cooked wonderful meals, proudly presented Ithaca and the surrounding area, and motivated me for working. This visit was a big inspiration, and a turning point in the research.

The Research Council of Norway funded the project and the visit to Cornell University, of which I'm very grateful.

Motivating words from people around me have been important from time to time. I cannot mention everyone by name, but they basically consist of family, my colleagues at the Department of Mathematical Sciences, my best friends from the master programme who meet at the "cantina", the "ISFiT 2001 contact group", the interest organization for doctoral candidates at NTNU (DION), my housing cooperative, friends from charity organizations, and all my friends in the US. I would particularly like to thank my mother Linn Mo for reading through the thesis and correcting some basic language errors.

Preface

The main purpose of this thesis is to study Stochastic Differential Equations (SDEs) using Path Integration (PI). Nonlinear dynamical systems is a main focus of this work, both because such systems present important challenges for the PI method, and because these systems are generally hard to analyze analytically and by means of other numerical methods. More specifically, three kinds of systems will be discussed; simpler test cases where analytical solutions exist, systems that exhibit chaotic response, and systems with instantaneous changes at fixed values for the variable. The last group consists of signum-kind nonlinearities, impact systems with completely elastic impact and contacting impacts, and absorbing boundaries in reliability studies. For some systems, especially the chaotic systems, the random noise is basically a way to model small perturbations of the system. The role of this noise is to make theory and numerical methods from stochastic processes applicable, to model imperfections or noise in a physical system, or to account for the uncertainty in parameter estimates in such models. In other systems, the noise could be the main driving force, without which the system would just converge to one stability point.

A lot of this work is based on research from three earlier PhD candidates, [20, 34, 50]. Some of the techniques developed there are continued, some observations are investigated more thoroughly, and some of the problems and challenges found in this process will be discussed and explained. This also means that some important issues like convergence, stability, choice of numerical methods etc. have already been answered, so only the conclusions are referred here.

In the beginning of this work, the main question was if PI could be used as a tool to study chaotic systems, and if so, in what sense. One of the main characteristics of chaotic systems are "strange attractors"; limit sets with a complicated structure and fractal dimension. When adding noise, these limit sets are in some ways smeared out in the probability density over the state space, but this density could still have a very rugged structure. This is one of the challenges for PI, which requires a good approximative representation of the probability density by some interpolating surface. Different ways to perform a stable interpolation is discussed, and it is shown that the choice of interpolation could be crucial for the accuracy and convergence of the method, even for much simpler problems. The interpolation problem is related to the problem of discretizing the state space variables. Interpolation error seems to be the main reason for unstable results in earlier work, so this is the first challenge discussed in this thesis, and requires close attention from those who plan to follow up this method further. All interpolation methods used here have been written by the author, and are modified to attain the required stability and accuracy for the problems discussed. So this work is new.

The second challenge is the nonlinearity of the system, which is extremely important to handle properly. Time stepping methods like the Euler method, Runge Kutta schemes, and Taylor will be discussed together with the problem of choosing the optimal time step length. The path integration itself also requires that the nonlinear effects are properly taken care of. Except for the Taylor method, the programming code for the time stepping method is written by the author, though using standard methods. A reason that library functions

are not used, is that the time stepping method is highly integrated in the rest of the code to allow for optimization when many points are propagated forwards simultaneously.

A third challenge is dimensionality. Solving a one-dimensional autonomous SDE is fairly trivial and can be done with high accuracy in short time. A differential equation with chaotic response will either have three or more dimensions, or contain terms that explicitely depend upon time, e.g. periodic forcing. It is also of interest to study systems of coupled oscillators, filtered noise processes etc., which gives systems with a higher number of dimensions. Not only will computational time and memory requirements grow exponentially with the number of dimensions; even writing the computer code for such systems soon becomes an extremely challenging task. How do you write efficient computer code for a given problem with any number of dimensions and any choice of approximation method - without typing errors? One solution is to let the computer code be generated by yet another program, and this is a new development. This required describing the method of Path Integration and all its numerical structures in a generalized way. Some of the mathematical observations in this generalization and methods used to create efficient code is presented in this book. A milestone for the PI method, was the implementation and running of a stable system with six dimensions (6D), which was finally accomplished in April 2007. Six dimensions may not seem to be a lot for researchers from other fields, but to the authors' knowledge a full 6D response PDF of an SDE has not been computed by any numerical method earlier.

It has not been a main focus to do detailed studies of a large number of differential equations with chaotic and non-chaotic response under the influence of various noise processes. This is a vast field with endless possibilities, and there are many articles discussing various topics related to this - often for some very specific differential equations. Partly because of the challenges described above, there is very little work performed earlier that allows for detailed comparizations to the results of the PI method.

Some theory on chaos, chaotic and non-chaotic strange attractors is included. In addition, some of the varieties of possible behaviour when such systems are affected by noise is described, and illustrated by the PI method. PI could also give more insight to the behaviour of a system through other statistics than the PDF. Since chaos was a main interest, a great deal of work has been done trying to obtain the maximal Lyapunov exponent (MLE) of a system. Also here, the main question is if the PI method can be used for this task. The MLE, and the Lyapunov spectrum, is not trivial to compute with the methods already proposed by other authors - for general systems.

There are many different kinds of questions that could be asked about adding noise to a differential equation. In some cases, the problem is stability of the attractor, and the noise is just there to create a small perturbation kicking the system off from the deterministic path. For other applications, the qualities of the noise process could be of vital importance, and different noise levels or noise processes could give very different behaviour of the system. The latter case requires knowing that the numerical method can handle the stochastic terms with sufficient accuracy. In cooperation with professor Mircea Grigoriu at Cornell University, this question developed into some studies of the compound Poisson process, and the possibilities of implementing compound Poisson noise in PI. A master student at

NTNU, Tore Selland Kleppe, studied the implementation of Lévy processes in PI, partly in correspondance with the author [26]. The results from these attempts to use the PI method in other fields will however not be discussed in this work, due to space limitations. Many methods in the field of stochastic dynamical systems are limited to processes with Gaussian increments. In the implementation of PI, a main focus is to avoid this restraint. Comparing PI to other numerical algorithms is also beyond the scope of this study, but a brief description of some other methods can be found in sections 2.2 and 4.5.

The structure of this thesis is as follows. Chapter 2 describes some basic theory of probability, dynamics, and stochastic dynamics. This is partly to set some notation for the proceeding chapters, and partly to introduce the subject to those not familiar with this field. Chapters 3 through 6 are devoted to numerical algorithms. First, in chapter 3, the discretization in time of a continuous system and a few time stepping methods are presented and discussed. This also includes some methods of solving SDEs numerically, such as moment equations and Monte Carlo simulations. This chapter is basically a review of earlier work, except that the implementation and combination of methods could be new. Then, in chapter 4, space is discretized, which means an interpolation method is needed. One would think that a review of well-known methods like linear interpolation and cubic B-spline interpolation would be sufficient, but a significant part of the PhD work has been devoted to stabilize interpolation methods, and also some new methods are proposed. In chapter 5 the PI method is presented, and some of its properties are investigated. Then chapter 6 discusses implementation issues of PI, especially with respect to interpolation. For all the numerical chapters, choices between different (seemingly equivalent) methods can be made. For some problems, this choice could be crucial for the stability – and even for the possibility of running the program, especially for the high dimensional case. This discussion has not been laid out in detail in previous work, and even still there are unanswered questions here.

In chapter 7, chaos is defined, and results for some chaotic systems are presented. One specific Duffing equation with harmonic forcing has been used as a model example, and is studied under the influence of different kinds of noise. This also serves as an example on how to interpret the resulting response PDF obtained from PI. This chapter includes some discussion on the possibility of computing the maximal Lyapunov exponent, or the Lyapunov spectrum for these systems using PI, and the comparization of various results obtained for the test equations. Also two constructed examples show how added noise can cause transitions from globally chaotic to globally non-chaotic, and opposite.

Finally, chapter 8 summarizes the main questions posed initially in this work, and the answers found so far.

The thesis also consists of some articles that were written during the work on this thesis. They give examples of how one can study specific systems with PI, and highlite specific observations, and present a broader view on stochastic differential equations. In these papers, the author has performed all the numerical work, with one exception. In appendix E, Finn-Erik Kolnes modified code from the author – with some assistance – to generate the results. Because some results were not completely satisfactory, the author adapted the original PI code again in a more stable manner, to verify and improve the

results. For all the articles, all the authors have taken part in the writing process, though the first author of each article wrote the major part of the text.

Notation list

To some extent, the text aims to reduce the number of different meanings for different variables. At the same time, some notation is well established in the field, so deviating from it will easily lead to confusion. Where a character is used with two very different denotations, like being both a scalar and a function, it is listed with a full stop. When constants, variables or functions have different interpretations depending on the setting without changing type, this is indicated by comma-separated list.

Latin lowercase

a	Vector field for the deterministic part of an SDE, often written $a(x)$.
$\widetilde{a}$	Approximated vector field for a based on some numerical method.
b	Vector field for the stochastic part of an SDE, often written $b(x)$ or $b_j(x)$ for integers j.
$\widetilde{b}$	Approximated vector field for b based on some numerical method.
c	Spline coefficient.
d	Real value, typically a diagonal value in a matrix.
d	Infinitesimal sign in integrals and proper differentials.
$\mathbf{e}$	Unit vector along some coordinate axis.
f	Probability density function.
$\widehat{f}$	Approximated function by interpolation.
$\widetilde{f}$	Approximated non-interpolating function.
g	General function, usually time stepping function.
g_0	One-sided spectral density constant.
h	General function, usually time stepping function.
i	$\sqrt{-1}$
j	Counting variable, order of interpolation function.
k	Counting variable.
l	Radius of ellipsoid along a principal axis.
m	Eigenvalues of a (flow) matrix. Integer (for dimension).
n	Number of dimensions.
p	Point in the state-space or real number.
q	Real number, temporary storage value in algorithms.
r	Number of dimensions where noise is not directly entering.
s	Calculated or estimated standard deviation.
t	Time.
u	Velocity.
v	Point value of function (to be interpolated).
w	Real number used for weighting or tension. Free variable.
x	Free variable.
y	Free variable.
z	Free variable.

Latin uppercase

A	Surface inside a geometric object. Path integration matrix.
B	Ball in the state space with uniform radius. Basis function for B-spline.
C	Compound Poisson process.
$\mathcal{C}_0$	Space of bounded functions that go to zero at infinity.
D	Differential operator, $D : \mathbb{R}^n \to \mathbb{R}^{n \times n}$.
$\mathcal{D}$	Space of all probability densities on some manifold.
E	Expectation operator.
F	Forcing function. Cumulative probability density function.
$\mathcal{F}$	σ-algebra, see section 1.2.1.
G	Linear operator.
H	Kernel of integral function or energy function.
I	Identity (matrix). Interval.
I_0	Modified Bessel function of the first kind.
J	Determinant of of the differential matrix, $J_h(\mathbf{x}) = \lvert Dh(\mathbf{x})\rvert$.
K	Positive constant, number of (knot) points.
L	Lower-triangular matrix. L^1, L^2 etc are vector spaces.
M	Number of gridpoints. Unspecified rectangular matrix. Martingale.
N	Formal time-derivative of W_t or poissonian stochastic variable.
$\mathbb{N}$	The set of all positive integers; $1, 2, \ldots$.
O	Order of numerical method.
P	Probability, probability density operator.
P_t	Unspecified stochastic process.
Q	Operator to compute the largest lyapunov exponents. Unitary matrix in QZ-factorization.
R	Radius.
$\mathbb{R}$	The real line.
S	Circle; $\mathbb{R} \mod 2\pi$, S^n is a hyper-sphere. Matrix evaluating spline coefficients. Spectrum.
$\mathbb{S}$	Circle, $\mathbb{S}^n$ is a cartesian product of circles (torus.)
T	Period or upcrossing time. Triangular matrix.
U	An (open) set. Upper-triangular matrix.
V	An open set.
W	Wiener process, white noise process with independent Gaussian increments.
X	Stochastic variable.
$\mathcal{X}$	Sample space for the stochastic variable.
Y	Stochastic variable, typically $Y = \dot{X}$, $\mathbf{Y} = \dot{\mathbf{X}}$.
Z	Stochastic variable. Unitary matrix in QZ-factorization.

Greek lowercase

α Real constant.
β Real constant.
γ Real constant, usually used for scaling of noise process in SDE.
δ Dirac delta function.
∂ Boundary, ∂A is the boundary of A. Infinitesimal sign in partial derivative.
ϵ Small number, radius of ϵ-ball.
ε Error in interpolant.
ζ Weighting function.
η Fixed point value for y. Perturbation of y.
θ Free variable for angle, $\boldsymbol{\theta} \in S^{n-1}$ or for the stochastic variable Θ.
ϑ Angle to the third axis in 3D space.
κ Real constant.
λ Lyapunov exponent, eigenvalue of matrix.
μ Expectation value in Gaussian (normal) PDF.
ν Probability measure.
ξ Fixed point value for x. Integration variable for x. Perturbation of x.
ρ Free variable for radial distance from a point to the origin in the state space.
ϱ Radius of circle or (hyper)sphere.
σ Standard deviation in Gaussian (normal) PDF.
ς Curve inside a geometric object.
τ Auxillary time variable or spline knot.
υ Free variable for the stochastic variable Υ.
ϕ Flow function of a dynamical system. Free variable for the stochastic variable Φ.
φ Flow matrix of a dynamical system, $\varphi(\mathbf{x}, t, \omega)\mathbf{x} = \phi(\mathbf{x}, t, \omega)$.
ω Stochastic outcome process. Real constant, typically angular velocity.

Greek uppercase

Γ Gamma function, constant matrix.
Δ Discretization length for the following variable.
Θ Stochastic variable for angle.
Ξ All collections of points from $\mathbb{R}^n$, compact volumes, line segments etc.
Π^{n-1} The $(n-1)$-dimensional projective space.
Σ Constant matrix.
Υ Stochastic variable, $\Upsilon = \dot{Z}$
Φ Stochastic variable for angle.
Ω A set.
$'$ denotes a variable at a previous time step $t' = t - \Delta t$.

Sub-indeces could indicate numbering or time. For simplicity, time is often skipped. The numbering of elements from a vector, unit vectors etc will correspond to the dimension

number, while constants and functions could be arbitrarily numbered. For specific systems, subscripts could be typeset as plain text to distinguish constants with the same symbol.

Boldface math denotes vector variables or constants, but is not used for functions, operators or matrices. All vectors are assumed to be column-vectors, written vertically. The symbol $^\top$ denotes vector or matrix transpose, typically switching from a row vector to a column vector or opposite. A vector can also be written out as $[\ldots]^\top$, e.g. $\mathbf{x} = [x, y, z]^\top$. This vector represents point in the state space. When the state space is discretized, a specific grid point could be referenced by a set of sub-indeces k_1, k_2, k_3, that is $\mathbf{x}_{k_1,k_2,k_3} = [x_{k_1}, y_{k_2}, z_{k_3}]^\top$.

Finally, complex conjugation is denoted by *. So if $z = x + iy$, then $z^* = x - iy$.

Contents

Chapter 1

Preliminaries

The purpose of this chapter is to introduce some basic notation and definitions for readers not familiar with the field of stochastic differential equations. More thorough introductions can be found in e.g. [1], [28], or [44].

1.1 The inverse and implicit function theorems

The following two theorems are presented without proof. More detailed discussions, examples and proof can be found in introductiory books to algebra.The theorems will be used in the development of the PI operator in section 4.1.

Theorem 1. ***The inverse function theorem.*** *Let $h : U \to V$, $U, V \in \mathbb{R}^n$ be a continuously differentiable function. If $Dh(\boldsymbol{\xi})$ is invertible for a fixed point $\boldsymbol{\xi} \in U$, then an open set W around $\boldsymbol{\xi}$ exists where h is an invertible function. The inverse function $h^{-1} : h(W) \to W$ is continuously differentiable.*

Theorem 2. ***The implicit function theorem.*** *Let $g : \mathbb{R}^{n+m} \to \mathbb{R}$ be a continuously differentiable function, and denote points in $\mathbb{R}^{n+m}$ by $(x'_1, x'_2, \ldots, x'_n, x_1, x_2, \ldots, x_m) = (\mathbf{x}', \mathbf{x})$. If $Dg(\boldsymbol{\xi}', \boldsymbol{\xi})$ is invertible for a fixed point $(\boldsymbol{\xi}', \boldsymbol{\xi})$ that fullfills $g(\boldsymbol{\xi}', \boldsymbol{\xi}) = 0$, an open set U with $\boldsymbol{\xi}' \in U$, an open set V with $\boldsymbol{\xi} \in V$, and a differentiable function $h : U \to V$ exist such that $\{(\mathbf{x}', h(\mathbf{x}')\} = (U \times V) \cap \{(\mathbf{x}', \mathbf{x}) | g(\mathbf{x}', \mathbf{x}) = 0\}$.*

1.2 Probability and statistics

1.2.1 Measure theory and the probability space

Let Ω be a set. A *σ-algebra* $\mathcal{F}$ on Ω is then a nonempty collection of subsets of Ω such that these three properties hold:

- $\Omega \in \mathcal{F}$, i.e. the largest subset in the σ-algebra is the whole set,
- $U \in \mathcal{F} \Rightarrow \Omega \setminus U \in \mathcal{F}$, so for every subset, its complement is also part of the σ-algebra,

- If $U_i \in \mathcal{F}, \quad i = 1, 2, \ldots$ then $\bigcup_{i=1}^{\infty} \in \mathcal{F}$, which means that taking unions of subsets in $\mathcal{F}$ will never throw you out of $\mathcal{F}$, even in the limit to infinity.

Roughly speaking, the σ-algebra is hence a collection of possible ways of partitioning your set. Every subset of Ω belonging to $\mathcal{F}$ is said to be $\mathcal{F}$-measurable. The items 1 and 2 above implies that the complement of Ω, which is called the empty set and denoted $\emptyset$ is also a member of $\mathcal{F}$.

The pair $(\Omega, \mathcal{F})$ is called a *measurable space*.

A *probability measure* P on a measurable space $(\Omega, \mathcal{F})$ is a function $P : \mathcal{F} \to [0, 1]$ such that

- $P(\emptyset) = 0$,
- $P(\Omega) = 1$,
- If $A_i \in \mathcal{F}, \quad i = 1, 2, \ldots$ and $A_i \cap A_j = \emptyset$ for all $i \neq j$, then $P(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$.

The triplet $(\Omega, \mathcal{F}, P)$ is called a *probability space*. In such a space, the members of $\mathcal{F}$ are called *events*. A single point in the probability space is called an *outcome*, so an event is a collection of outcomes. $P(A)$ is then the probability that the event A occurs. The three items in the list above can then be read: The probability of the no outcome event is zero, the probability of the event containing all outcomes is one, and probabilities can be added for disjoint (non-overlapping) events.

Two events U and V are *independent* if $P(U \cap V) = P(U)P(V)$.

A *conditional probability* is a function $P(U|V) = P(U \cap V)/P(V)$, defined for all $\mathcal{F}$-measurable V except $\emptyset$. The conditional probability also fulfills the three criteria for a probability measure, and is hence also a probability measure. Independence could also be defined from the conditional probability by the formula $P(U|V) = P(U)$.

1.2.2 Random variables

A *random variable*, sometimes denoted a *stochastic* variable, is a function $X : \Omega \to \mathbb{R}$ such that $U \in \Omega : X(U) \leq x \in \mathcal{F}$ for every $x \in \mathbb{R}$. The random variable is $\mathcal{F}$-measurable if $X^{-1}(U) \in \mathcal{F}$ for every Borel set U. The collection of Borel sets are all possible unions and intersections of open and closed sets. The outcome U will usually not be written as an argument in this text.

1.2.3 Probability densities

From here on and throughout the thesis, the set $\mathcal{X}$ will be the *sample space* of X, i.e. all values that X could take are members of $\mathcal{X}$. Further, $\mathcal{X}$ will here always be some compact subset of $\mathbb{R}^m \times \mathbb{S}^n$ for some integers m and n where one of them could be zero. Here $\mathbb{R}$ is the real line and $\mathbb{S}$ the circle, usually represented by a real number $[0, 2\pi\rangle$ modulo 2π. This space is spanned by $m + n$ orthogonal eigenvectors, and one random variable from is assigned to each direction. These random variables are indexed as $X_1, X_2, \ldots, X_{m+n}$,

where X_j is simply the function extracting the jth component of the coordinates of the point $\mathbf{X}$. The allows the standard definition of a sigma-algebra over the state space as the collection of all tensor products of possible connected intervals in these variables.

If $\mathcal{X}$ is only a subset of $\mathbb{R}$ or $\mathbb{S}$ in some variable, it can be trivially extended to $\mathbb{R}$ or $\mathbb{S}$. By a suitable extension of $\mathcal{F}$, and defining $P(U) = 0$ for every U disjoint with the original $\mathcal{X}$, the probability space is retained. So in the teoretical work, we will always assume $\mathcal{X} = \mathbb{R}^m \times \mathbb{S}^n$ so that the range for all the random variables is from $-\infty$ to ∞, or 0 to 2π for angles. Stochastic differential equations could be defined on other manifolds too, but except for a brief discussion on systems on the projective plane Π^{n-1} in section 6.13, and spheres and hyper-spheres in section 6.15, this will be beyond the scope of this work.

If all the random variables live in $\mathbb{R}$, i.e. $\mathcal{X} = \mathbb{R}^n$, the *cumulative probability function* can be defined by $F(x_1, x_2, \ldots, x_n) = P(X_1 \in \langle -\infty, x_1] \cap X_2 \in \langle -\infty, x_2] \cap \ldots)$. This fulfills $F(-\infty, -\infty, \ldots) = 0$ and $F(\infty, \infty, \ldots) = 1$. If F is differentiable in all the state space variables, the *probability density function* (PDF) is the function

$$f(\mathbf{x}) = f(x_1, x_2, \ldots, x_n) = \frac{\partial^n F(x_1, x_2, \ldots, x_n)}{\partial x_1 \partial x_2 \cdots \partial x_n}. \tag{1.1}$$

Some dynamical systems, e.g. systems with dry friction, could result in a cumulative probability function that is not differentiable everywhere. The corresponding PDF will contain delta-shaped peaks or structures. This construction is beyond the scope of this thesis. Generally, the probability density is a function f such that for every event $U \in \mathcal{F}$ the probability of U equals $\int_U f(\mathbf{x})\mu(\,\mathrm{d}\mathbf{x})$. This gives $\int_{\mathcal{X}} f(\mathbf{x})\mu(\,\mathrm{d}\mathbf{x}) = 1$.

A probability density that will be used extensively in this book is the normal or Gaussian PDF. In n dimensions, a multivariate case, with independent variables it reads

$$f(\mathbf{x}) = \frac{(2\pi)^{-n/2}}{\prod_{j=1}^n \sigma_n} \exp\left(-\sum_{j=1}^n \frac{(x_j - \mu_j)^2}{2\sigma_j^2} \right). \tag{1.2}$$

The expectation operator is an integral with respect to the probability measure over the whole space;

$$\mathrm{E}(g(\mathbf{X})) = \int_{\mathcal{X}_t} g(\mathbf{x}) f(\mathbf{x}) \mu(\,\mathrm{d}\mathbf{x}). \tag{1.3}$$

1.3 Stochastic processes

A *stochastic process* is a collection of random variables parameterized in time, e.g. $X_t, t \in \mathbb{R}^+$. The time parameter will be written as an argument $X_t = X(t)$ where the subscript is already used to denote dimension, and where time explicitly appears in an equation or formula. The time parameter might also be skipped where it is regarded unneccessary.

1.3.1 Markov processes and Markov operators

All the systems studied in this text will follow the *Markov property*, namely that the future behaviour is independent of the past - given knowledge about the present. Mathematically,

$$f(x(t+\Delta t)|x(s)=\xi(s), s\leq t) = f(x(t+\Delta t)|x(t)=\xi(t)), \quad \text{for all } t, \Delta t > 0 \text{ and fixed } \xi(t). \tag{1.4}$$

In some cases this is obtained by including a sufficiently large number of state-space variables to the system. These variables then carries the necessary information about the past in their present value.

Let $(X, \mathcal{F}, \mu)$ be a measure space, and let $\mathcal{L}^1$ denote the integrable functions with respect to this space, that is, every $f \in \mathcal{L}^1$ satisfies $\|f\| = \int_X |f| \, \mathrm{d}\mu < \infty$. Any linear operator $P : \mathcal{L}^1 \to \mathcal{L}^1$, satisfying the following properties

(1) $Pf \geq 0$ for $f \geq 0$, $f \in \mathcal{L}^1$,

(2) $\|Pf\| = \|f\|$ for $f \geq 0$, $f \in \mathcal{L}^1$,

is called a *Markov operator* [29]. Hence, any linear operator $P : \mathcal{L}^1 \to \mathcal{L}^1$ that preserves densities, that is, $P : \mathcal{D} \to \mathcal{D}$, is a Markov operator. Here $\mathcal{D} = \{f \in \mathcal{L}^1 : f \geq 0, \|f\| = 1\}$.

1.3.2 The Wiener process and white noise

The *Wiener process*, or Brownian motion, W_t is characterised by the following properties;

- $W_0 = 0$.
- If $0 < \tau < t$, then $W_t - W_\tau \sim N(0, t-\tau)$.
- If $0 < t_1 < t_2 \leq t_3 < t_4$, then $W_{t_2} - W_{t_1}$ and $W_{t_4} - W_{t_3}$ are independent variables. That is, for any pair of non-overlapping intervals, the behaviour in one is independent of the other.
- Further, $E(W_t) = 0$ and $E(W_\tau W_t) = \min(\tau, t)$.

This process is continuous, as $W_\tau \to W_t$ almost surely (a.s.) when $\tau \to t$. However, it is almost surely not differentiable - in any point. To avoid the technical definition of a formal derivative, the noise process $N_t = N(t)$ can be introduced as follows. For every t, N_t is a Gaussian zero-mean random variable with unit variance, and N_τ is independent of N_t when $\tau \neq t$. Hence, these variables are uncorrelated, which results in a uniformly distributed (flat) spectrum, which motivates the notation "white" noise. So the process N_t is denoted *Gaussian white noise*. Such a process cannot exist in a real physical system, but is in some cases a good approximation, and the random noise could be made more physical through filtering of the white noise process. Many theoretical noise processes represent white noise. Therefore, one should always specify the underlying distribution of the process. For this thesis, the process is always Gaussian white noise.

Now use this process as input to a simple differential equation;

$$\mathrm{d}X_t = N_t\,\mathrm{d}t = N(t)\,\mathrm{d}t. \tag{1.5}$$

When the response X_t is calculated for equation (1.5), e.g. by a numerical method, the result is indistinguishable from a (time discretized) Wiener process. Therefore, the notation $N_t\,\mathrm{d}t = \mathrm{d}W_t$ is used. The numerical computation of equations like (1.5) is discussed in the following section.

1.3.3 Stratonovich and Itô calculus

The next problem is to define stochastic integrals. This is here discussed for one dimension only, however, these results are trivially generalized to more dimensions. The Riemann method of integrating a function X_t with respect to a process P_t is

$$\int_{t_1}^{t_2} X_t\,\mathrm{d}P_t \approx \sum_{j=1}^{M-1} X_{\tau_j^*}(P_{\tau_{j+1}} - P_{\tau_j}) \tag{1.6}$$

with a partitioning $t_1 = \tau_1 < \tau_2 < \ldots < \tau_M = t_2$ and $\tau_j^* \in [\tau_j, \tau_{j+1}]$. When taking the limit $m \to \infty$, the right hand side should converge to the integral.

Unfortunately, the Riemann method does not converge uniquely for stochastic processes, but depends on how the evalutation points τ_j^* are chosen. Two different definitions of the integral are commonly used, namely the Itô integral $\tau_j^* = \tau_i$ and the Stratonovich integral $\tau_j^* = (\tau_{j+1} + \tau_j)/2$. Excluding the outcome process ω_t and returning to the Wiener process, the Itô and Stratonovich integrals are then formally written

$$\int_{t_1}^{t_2} X_t\,\mathrm{d}W_t = \lim_{M\to\infty,\Delta\to 0} \sum_{j=1}^{M-1} X(\tau_j)(W_{\tau_{j+1}} - W_{\tau_j}), \tag{1.7}$$

$$\int_{\alpha}^{t_2} X_t \circ \mathrm{d}W_t = \lim_{M\to\infty,\Delta\to 0} \sum_{j=1}^{M-1} X(\frac{\tau_j + \tau_{j+1}}{2})(W_{\tau_{j+1}} - W_{\tau_j}), \tag{1.8}$$

respectively. The sequence $\{\tau_j\}_{j=1}^{M}$ is the same as for the Riemann integral in equation (1.6). Equations (1.7) and (1.8) leads to the corresponding *stochastic differential equations* (SDEs)

$$\mathrm{d}X_t = a(X_t, t)\,\mathrm{d}t + b(X_t, t)\,\mathrm{d}W_t, \tag{1.9}$$

$$\mathrm{d}X_t = a(X_t, t)\,\mathrm{d}t + b(X_t, t) \circ \mathrm{d}W_t. \tag{1.10}$$

It can be shown that an SDE of one kind with a modified drift term has the same solution as the other, where solution can be interpreted many ways, as will be shown in the next section. E.g. solving the Stratonovich SDE from equation (1.10), is equivalent to solving the Itô SDE

$$\mathrm{d}X_t = \left(a(X_t, t) + \frac{1}{2}b(X_t, t)\frac{\partial b}{\partial x}(X_t, t)\right)\,\mathrm{d}t + b(X_t, t)\,\mathrm{d}W_t. \tag{1.11}$$

So an algorithm for solving Itô SDEs is directly adapted for solving Stratonovich SDEs. In the following, every SDE presented should be interpreted in Itô sense. Note that the Itô and Stratonovich representations are equal in the case of additive noise, i.e. when $\frac{\partial b}{\partial x} = 0$ everywhere. The extra term in equation (1.11) is called the Wong-Sakai correction term. For a noise process that is not additive, this directly influences the expectation value of the system.

1.3.4 Dynamical systems and flows

Given a set of $r+1$ smooth vector fields on $\mathbb{R}^n$, $a, b_1, \ldots, b_r : \mathbb{R}^n \to \mathbb{R}^n$ and a standard $\mathbb{R}^r$-valued Brownian motion, given as $W_t^1, W_t^2, \ldots, W_t^r$ for $t \geq 0$, on some probability space $(\Omega, \mathcal{F}, P)$. From this an SDE can be written

$$\mathrm{d}\mathbf{X}_t = a(\mathbf{X}_t, t)\,\mathrm{d}t + \sum_{j=1}^{r} b_j(\mathbf{X}_t, t)\,\mathrm{d}W_t^j, \quad X_0 = x_0. \tag{1.12}$$

Very often, an initial condition will be specified, $\mathbf{X}_0 = \mathbf{X}_{\text{init}}$, where $\mathbf{X}_{\text{init}}$ could be a random variable with some given probability density, or a deterministic point in the state space. A simple additive noise process is obtained for $b_1(\mathbf{x}, t) \equiv (0, 0, \ldots, 0, \gamma)^\top = \text{const}$, and $r = 1$. This is the most typical example in this thesis. Note that this simple process does not have a stable fix point, and this is generally assumed through the rest of the work. Other studies of SDEs restrict the analysis to very different cases, i.e. Baxendale [5] where $b_1(\mathbf{0}, t) = b_2(\mathbf{0}, t) = \ldots = b_r(\mathbf{0}, t) = 0$, and $b_j(\mathbf{x}, t)$ is independent of t. In this work, whenever a or b_j has only one index, it is assumed that they are independent of t.

The *solution* of an SDE is ambiguous, so this term will mostly be avoided from here on for SDEs. The SDE induces a *random flow*, or *random dynamical system* (RDS) over the state space that will be represented by the function $\phi : \mathbb{R}^n \times \mathbb{R} \times \Omega \to \mathbb{R}^{n \times n}$. For equation (1.12) the resulting flow function will be written $\mathbf{X}_t = \phi(\mathbf{X}_{\text{init}}, t, \omega_0)$. The criterion $\mathbf{X}_{\text{init}} = \mathbf{X}_0$ implies that $\phi(\cdot, 0, \omega)$ is the identity. The order of the arguments of ϕ is reversed compared to standard litterature in this field. The reason is that for many of the systems here, ϕ depends nonlinearly on the $\mathbf{X}$ argument, the ω argument will for simplicity be skipped many places.The time argument t refers to the length of time the random variable $\mathbf{X}$ is propagated. So for autonomous systems, the fact that the system could also depend on the value of t is not included.

In some cases, it will be convinient to write the flow as a matrix operator $\mathbf{X}_t = \varphi(\mathbf{X}_0, t, \omega_0)\mathbf{X}_0$. Here, the rhs is equivalent to $\phi(\mathbf{X}_0, t, \omega_0)$. Much work has been devoted to study linear SDEs, where the first argument can be skipped, which might explain the typical order of arguments in other texts. In e.g. [1], φ is a nonlinear operator and not a matrix, to avoid using $\mathbf{X}$ as an argument.

Within a time step in a discretized system and for deterministic systems, ϕ will not be considered a random variable. In these situations, the first input argument is also deterministic. However, generally ϕ will depend on the outcome process ω. If this is assumed known, e.g. by simulation with a random number generator, the resulting flow

will be referred to as a *realization* of the SDE. In many papers, individual realizations, or the collection of all possible realizations, is referred to as the solution of the SDE. Some authors would require the realization of the pair (X_t, W_t) to be given to call it a solution [19]. For a deeper discussion on the connection between SDEs and RDSs, the reader is recommended to study Arnold's well-known book [1].

Another way of solving the SDE, is to find the *response probability density* for $\mathbf{X}_t$, $f_t(\mathbf{x}|\mathbf{X}_0 = \mathbf{X}_{\text{init}})$ for some t or all t, which is a PDF in the sense of equation (1.1). For many of the systems studied here, f_t will converge by increasing t to a stationary PDF, which will not depend on $\mathbf{X}_0$. This is the *stationary PDF* for the SDE, which simply will be written $f(\mathbf{x})$. In such cases, the initial condition will be skipped. A stationary distribution for a system means that the Markov operator P from section has a fixed point. In section 4.1 it will be shown how this operator can be implemented numerically, which is the main goal of this research project. The sequence of response PDFs for all t does not carry the same information as the realizations, just as a stationary PDF does not say anything about the underlying process after reaching stationary behaviour. Therefore, it could be argued that solving an SDE in the sense of probability densities must require finding the joint PDF of the state space variables at all times, i.e. $f(\mathbf{x}_t, t, \mathbf{x}_\tau, \tau)$ for every possible combination of t and τ, such that all conditional densities of the form $f(\mathbf{x}_t, t|x_\tau = \mathbf{x}_\tau)$ can be calculated. This kind of joint PDF will not be found by the method described here.

Solving an SDE could also involve determining the stochastic asymptotic stability for all possible initial conditions, or mapping other properties of stability or asymptotic behaviour. A detailed theoretical discussion of stability, uniqueness, boundedness etc for SDEs can be found in Gards introduction to stochastic differential equations [13].

1.3.5 Dynamics and differential equations

Nonlinearity can cause a variety of effects. The behaviour of the system can be chaotic, periodic, oscilliating without a fixed period, stable or unstable. The *attractor* of a dynamical system is a the set of points that the system converges to after a sufficiently long time. This could one or more single points, or a curve or another compact subset of the state space. The latter includes complicated stuctures without an integer dimension, the so-called *strange attractors*. In some systems there are critical values, in terms of a change of parameter or a change in initial values, for which the system changes abruptly. Understanding such dynamics requires qualitative descriptions. Analytical tools can be used to examin local behaviour and stability, but in the general case numerical methods must be used to answer questions of global behaviour. However, one must recognize that a numerical method is by itself a new dynamical system, possibly with a quite different solution, at least in some cases. Together with sensitivity or unstability of the system studied, we are faced with computational difficulties. Sometimes it is better to avoid the problem through selection of other methods, a change of parameters, or choosing better starting conditions than trying to overcome the difficulties with a general algorithm.

A common tool for studying the dynamics of a system is state space plots. Based on realizations Poincaré maps. For some chosen sampling time T, usually equal to the period

of the differential equation, we register $\mathbf{X}_t, \mathbf{X}_{t+T}, \mathbf{X}_{t+2T}, \ldots$, and plot these values together.

1.3.6 Examples of differential equations

A very simple system is a linear filter, or an oscillator, with no damping.

$$\begin{aligned} \mathrm{d}x &= y\,\mathrm{d}t, \\ \mathrm{d}y &= -\alpha_1 x\,\mathrm{d}t. \end{aligned} \tag{1.13}$$

Without a forcing term, the system results in a pure harmonic wave, which corresponds to an ellipse in the state space, or a delta-shaped spectral frequency. The frequency spectrum for a process is found from a temporal Fourier transformation of the response process $x(t)$, this will be described in section 5.3. The ellipse drawn by $(x(t), y(t))$ in the state space from equation (1.13) is uniquely determined from the initial value (x_0, y_0), and there is no transient behaviour. From one single initial value (x_0, y_0), every Poincaré map will be just one point in the state space, and the location of this point is determined by where on the period the map is located.

Adding noise to equation (1.13) results in a stochastic system that almost surely diverges from every initial point. This illustrates a typical effect of added noise called *diffusion*. Also the white noise process, or random walk, W_t, has diffusion. The standard deviation in the probability density for $X_t = W_t$, say $f(x,t)$, given $W_0 = 0$ is proportional to the square root of t.

For both these stochastic systems, the probability density would gradually become wider, and $\lim_{t\to\infty} \max(f(\mathbf{x},t)) \to 0$ for every $\mathbf{x}$. To balance out the diffusion from an additive noise process, some kind of damping term is needed. In a variable X_t, the damping term should make $E(\dot{X}_t)$ negative when $X_t \gg 0$, and positive when $X_t \ll 0$. The only possible linear damping term is a constant times $-X_t$, but a number of different nonlinear terms will do, e.g. any $-K|X_t|^\alpha$ for every real $\alpha \neq 0$ and $K > 0$. A damping term typically alters the spectrum, but most systems will still be dominated by one frequency, an *eigenfrequency*.

1.3.7 Some SDEs with analytical solution

Some SDEs have analytically known solutions. Knowing a few such examples is a big help for testing convergence and accuracy of numerical methods, and a discussion of these is also valuable to point out important aspects of SDE behaviour in general. This list starts with one-dimensional SDEs, then some two-dimensional systems are presented, and finally a few general solutions are found by matrix equations. In all the displayed equations below, $f(x)$ (or $f(x,y)$ or $f(\mathbf{x})$) is the stationary PDF of the system, with some requirements on parameters of the SDE. K is a constant needed for f to integrate to one, not the same constant for the different equations.

The most simple SDE with a stationary distribution is linear with additive noise.

$$\mathrm{d}X_t = -\alpha X_t \,\mathrm{d}t + \gamma \,\mathrm{d}W_t, \tag{1.14}$$

$$f(x) = K \exp\left(-\frac{\alpha x^2}{\gamma^2}\right), \quad \alpha > 0. \tag{1.15}$$

This illustrates that a linear filter preserves the Gaussian distribution. Also note that for every $\alpha > 0$ there is a stationary solution, i.e. the process will with probability one not diverge to infinity.

To study nonlinear effects, a cubic term can be added.

$$\mathrm{d}X_t = -(\alpha_1 X_t + \alpha_3 X_t^3) \,\mathrm{d}t + \gamma \,\mathrm{d}W_t, \tag{1.16}$$

$$f(x) = K \exp\left(-\frac{1}{2\gamma^2}(2\alpha_1 x^2 + \alpha_3 x^4)\right), \quad \alpha_3 > 0. \tag{1.17}$$

If $\alpha_1 < 0$, the density is bimodal with a local minimum in $x = 0$, and with $\alpha_1 = 0$, there is zero curvature at the maximum at $x = 0$. This system is very suitable to study interpolation of a function that goes smoothly to zero at infinity.

The analytical solution can be generalized to other drift functions and multiplicative noise.

$$\mathrm{d}X_t = a(X_t) \,\mathrm{d}t + b(X_t) \,\mathrm{d}W_t, \tag{1.18}$$

$$f(x) = \frac{K}{b(x)^2} \exp\left(2 \int^x \frac{a(\xi)}{b(\xi)^2} \,\mathrm{d}\xi\right). \tag{1.19}$$

So one just needs to be able to integrate the fraction in the exponent. The integral is indeterminate, as the lower limit only affects the normalizing constant K. The formula for the stationary density can be verified by inserting it into e.g. the forward Kolmogorov equation, which for equation (1.18) reads

$$\frac{\partial(a(x)f(x))}{\partial x} - \frac{1}{2}\frac{\partial^2(b(x)^2 f(x))}{\partial x^2} = 0. \tag{1.20}$$

This general formula can be used for a variety of formulas. One less trivial example is

$$\mathrm{d}X = (\alpha - \beta X) \,\mathrm{d}t + \sqrt{\gamma X(1-X)} \,\mathrm{d}W, \quad 0 \le X \le 1, \quad beta > \alpha > 0, \quad \gamma > 0,$$

$$f(x) = \frac{\Gamma(p+q)}{\Gamma(p)\Gamma(q)} x^{p-1}(1-x)^{q-1}, \quad p = 2\frac{\alpha}{\gamma}, \quad q = 2\frac{\beta - \alpha}{\gamma},$$

$$\mathrm{E}(X) = \frac{\alpha}{\beta}, \quad \mathrm{Var}(X) = \frac{\alpha\gamma}{\beta^2}\frac{\beta - \alpha}{\beta + \gamma}.$$

Using $\alpha = 1$, $\beta = 2$, and $\gamma = 1/2$, the PDF is nicely bell-shaped, quite similarly to the Gaussian Âdistribution although with $f(0) = f(1) = 0$, while a steep tail with nonzero derivative at the right hand side is obtained by letting $\beta \to 1^+$. This will clearly be of

use to reveal typical interpolation challenges. Increasing γ introduces this tail behaviour at both sides of the domain.

A very different one-dimensional SDE with additive noise is the Rayleigh equation,

$$\mathrm{d}X_t = -\frac{\gamma^2}{2}\left(X_t - \frac{\kappa}{X_t}\right)\mathrm{d}t + \gamma\,\mathrm{d}W_t, \quad X_t > 0, \kappa > 0 \tag{1.21}$$

$$f(x) = \frac{x}{\kappa}\exp\left(\frac{x^2}{2\kappa}\right). \tag{1.22}$$

This will be discussed in particular in section 5.10.

The stochastic linear oscillator has two state-space variables.

$$\begin{aligned}\mathrm{d}X_t &= Y_t\,\mathrm{d}t,\\ \mathrm{d}Y_t &= -(\omega^2 X_t + \alpha Y_t)\,\mathrm{d}t + \gamma\,\mathrm{d}W_t,\end{aligned} \tag{1.23}$$

$$f(x,y) = K\exp\left(-\frac{\alpha}{\gamma^2}(\omega^2 x^2 + y^2)\right). \tag{1.24}$$

Also here, the system can still be solved when adding a cubic term, e.g.

$$\begin{aligned}\mathrm{d}X_t &= Y_t\,\mathrm{d}t,\\ \mathrm{d}Y_t &= -(\alpha_1 X_t + \alpha_3 X_t^3 + \kappa Y)\,\mathrm{d}t + \gamma\,\mathrm{d}W_t,\end{aligned} \tag{1.25}$$

$$f(x,y) = K\exp\left(-\frac{\kappa(2\alpha_1 x^2 + \alpha_3 x^4 + 2y^2)}{2\gamma^2}\right). \tag{1.26}$$

The cases where a stationary PDF can be found analytically in two dimensions are more limited than in one dimension. If the system is an oscillator with known potential energy wrt position, $h(x)$, and one writes the derivative $h'(x) = \mathrm{d}h/\,\mathrm{d}x$, the following is obtained;

$$\begin{aligned}\mathrm{d}X_t &= Y_t\,\mathrm{d}t,\\ \mathrm{d}Y_t &= -\alpha Y_t - h'(X_t) + \gamma\,\mathrm{d}W(t),\end{aligned} \tag{1.27}$$

$$f(x,y) = K\exp\left(-\frac{2\alpha}{\gamma^2}H(x,y)\right), \tag{1.28}$$

where the energy function $H(x,\dot{x}) = h(x) + \dot{x}^2/2$, i.e. the potential and kinetic energy.

In a the multi-degree-of-freedom system, some SDEs can be solved.

$$\begin{aligned}\mathrm{d}\mathbf{X}_t &= -\Gamma\mathbf{X}_t\,\mathrm{d}t + \mathbf{\Sigma}\,\mathrm{d}W_t,\\ f(\mathbf{x}) &= Ke^{-\frac{1}{2}\mathbf{x}^\top M\mathbf{x}},\end{aligned}$$

where $M, \Gamma \in \mathbb{R}^{n\times n}$, $\mathbf{\Sigma} \in \mathbb{R}^n$, and M fulfills $\Gamma^\top M + M\Gamma = M(\mathbf{\Sigma}\mathbf{\Sigma}^\top)M$.

Finally, it could be added that linear filtering of stochastic processes can be handled analytically, at least in some cases. Assume an SDE of the form

$$\mathrm{d}\mathbf{X}_t = (a_1(t)\mathbf{X}_t + a_2(t)\mathbf{Y}_t)\,\mathrm{d}t, t \geq 0.$$

The process $\mathbf{X}_t$ can then be expressed

$$\mathbf{X}_t = H(t,0)\mathbf{X}(0) + \int_0^t H(t,\tau)a_2(\tau)\mathbf{Y}(\tau)d\tau,$$

where H must fulfill $\frac{\partial H(t,\tau)}{\partial \tau} = a_1(t)H(t,\tau)$ for all $t \geq \tau \geq 0$ and $H(\tau,\tau)$ is the identity matrix.

1.3.8 Volume stretching and contraction

In addition to propagating single points with the dynamical system, it also gives meaning to propagate subsets of the full state space. The main effect then is to study the behaviour of the dynamical system on a measure like volume, area, or distance. This work will show that describing this process is necessary to classify systems as chaotic, and to perform the numerical algorithm presented in this paper correctly.

Consider the one-dimensional damped ODE

$$\mathrm{d}x_t = -\alpha x_t \,\mathrm{d}t. \tag{1.29}$$

The solution to this equation is

$$x_t = Ke^{-\alpha t} = x_{t-\Delta t}e^{-\alpha \Delta t} = x_0 e^{-\alpha t} = \phi(x_0, t). \tag{1.30}$$

Now propagate a small interval Δx between x_0 and $x_0 + \Delta x$ forwards in time a small time step Δt. The length of the interval then changes to $\phi(x_0 + \Delta x, \Delta t) - \phi(x_0, \Delta t) = \Delta x e^{-\alpha \Delta t}$, i.e. the the interval shrinks, for $\alpha > 0$, by a factor $e^{-\alpha \Delta t}$.

For nonlinear systems, the relative change of interval lengths will normally have to be computed numerically. In section 4.2 one such attempt will be presented for systems $\mathrm{d}\mathbf{x}_t = a(\mathbf{x}_t, t)\,\mathrm{d}t$ where $a(\mathbf{x}_t, t)$ is differentiable in both arguments. The derivative wrt the spatial variable is only needed to derive the numerical approximation, and the resulting expression also works well for piecewise smooth systems where a is continuous in space. Then, in section 4.4, the line of thought shown above is extended to simple piecewise systems that have a discontinuity, to complete the picture. In this setting, the flow function ϕ from section 1.3.4 will be extended to $\phi : \Xi \times \mathbb{R} \times \Omega \to \Xi$, where Ξ represents all collections of points, i.e. all subsets from $\mathbb{R}^n$. This extension of notation should not lead to confusion.

The long-term effect of the dynamics on line segments and measuring the lengths of the resulting lines lead to the maximum Lyapunov exponent, which is described in detail in equation (6.22) in section 6.7.

For a two-dimensional system, the change of infinite decimal areas will be studied. That is, $\phi(\,\mathrm{d}A, \Delta t)$ will be considered for correct stepwise integration of two-dimensional SDEs, typically with $\mathrm{d}A = \mathrm{d}x\,\mathrm{d}y$. Similarly, in three dimensional SDEs one should consider the propagation of infinite decimal volumes for the stepwise integration.

1.3.9 Stochastic perturbations of deterministic systems

Adding a small noise term to a deterministic ODE could in some cases give new and valuable insight to the dynamics of the system. One example is to study stability of the system. If the noise term changes the dynamics radically, the deterministic ODE will probably be unstable in the sense that it is very sensitive to round-off errors or small deviation in the initial values. The borderline between the stochastic and the deterministic world is not necessarily trivial. One would expect similar results from simulations when the noise level is small. However, this might not be the case. The small noise term could lead some of the probability mass into a potential well that is not seen by the deterministic solution, or completely eliminate a deterministically stable solution. One such example will be studied numerically in section 1.4, two similar constructed situations will be studied in sections 6.17 and 6.18. In some systems, an attractor can turn into a repellor when sufficient noise is added. This is denoted "noise-induced crisis" in Sommerer et.al. in [45].

For an autonomous system, the stationary response PDF of the stochastic system should be compared to the attractor, limit cycle or fix-point of the corresponding deterministic system, possibly with some probability distribution function as discussed in the introduction. For a time-dependent system, the averaged processes should be comparable in the same sense, and the Poincaré sections of the deterministic system with a properly chosen sampling interval will in some sense correspond to the instance response PDF of the stochastic system. In short, all the possible resulting outcomes of the deterministic system, from a single fixed point to the continuous set of Poincaré sections, will be referred to as the system's deterministic attractor.

Results obtained when studying systems with and without added noise, indicate that the typical behaviour is that the structure is preserved. A survey of such methods can be found in [25]. Some theory is actually based on this assumption. Due to round-off errors, a digital simulation of a system cannot exhibit a completely random pattern. There are only a finite number of possible states, which means that the simulation will eventually become periodic, hence any invariant probability measure of the original system will not be preserved. Ulam [53] proposed already in 1960 that one should add random perturbations to such simulations to avoid spurious periodicity, and this scheme is still used e.g. in [48]. However, the requirements for this kind of stability is not known, nor is a general theory for the behaviour of a system influenced by very high levels of noise.

Finally, in many practical experiments, a perfect deterministic behaviour cannot be guaranteed. It seems reasonable to assume that noise is present in input forces, there might be imperfectnesses in the mounting or physical couplings of the system, and there might be small errors in the model, measurement errors etc. In this sense, an SDE probably represent a more realistic model for applications than deterministic equations. Even if a correct modelling also of the noise processes is infeasable, a study of and SDE version of your equation could be valuable for stability analysis.

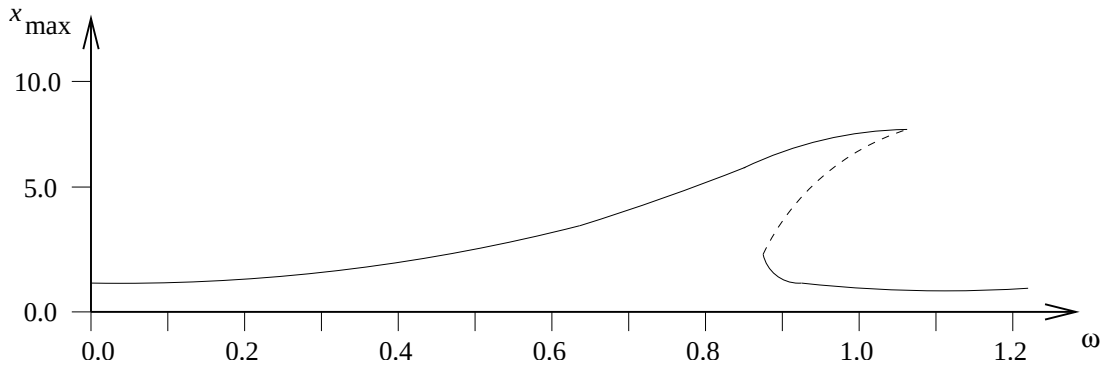


Figure 1.1: Sketch of the amplitude $x_{\max}$ for the stable deterministic solutions of equation (1.31) as a function of the parameter ω.

1.4 Example: stability of a nonlinear oscillator

Consider the following ODE, described in the introduction to [52];

$$2.56\ddot{x} + 0.32\dot{x} + x + 0.05x^3 = 2.5\cos(\omega_f t). \tag{1.31}$$

The lhs has a natural frequency $\omega = 0.625$. For values of ω_f from 0 to about 0.8, the system ends up in a steady-state harmonic, and the amplitude (in x) of this harmonic solution increases with increasing ω_f. Then, for a small interval around $\omega_f = 1$, two alterative stable steady states exist, and which one that is reached depends on the initial values $(x(0), \dot{x}(0)$. These two steady states have the same frequency, but differ in phase and amplitude. For $\omega_f = 1$ and $t = 0 \mod 2\pi$, one starts at $(x = 3.15, \dot{x} = 5.81)$ and has an amplitude of about 7, the other starts at $(x = -1.63, \dot{x} = 0.34)$ with an amplitude of about 2. For larger values of ω_f, at least up to 2.0, only one steady state exist, but this time with much smaller amplitude and the amplitude decreases with increasing ω_f. The amplitudes of the stable deterministic solutions are sketched in Figure 1.1.

With a small additive noise term, equation (1.31) could be rewritten as an SDE:

$$\begin{aligned} \mathrm{d}X &= Y\,\mathrm{d}t, \\ \mathrm{d}Y &= (-0.125Y - 0.390625X - 0.00195312X^3 + 0.9765625\cos(\omega t))\,\mathrm{d}t + 0.025\,\mathrm{d}W. \end{aligned} \tag{1.32}$$

For the simulation here, the initial value was chosen as a bivariate Gaussian PDF centered in $(\mu_{X_0} = 3.15, \mu_{Y_0} = 5.81)$, with $\sigma_{X_0} = \sigma_{Y_0} = 0.26$.

The initial behaviour of this system is illustrated with a collection of PDFs in Figures 1.2 to 1.5. These are obtained by using the numerical method that will be introduced in chapter 4, path integration (PI). The response PDF is sampled every $\Delta t = 2\pi/64$, and the first 500 samples are simply added together. A logarithmic color image of the process then shows the transient behaviour. This visualisation of the process is illustrative if one, like in this case, knows that the motion in the state space is counter-clockwise, and the PDF for these samples is sufficiently localized. In Figures 1.2, 1.3, 1.4, and 1.5 this transient

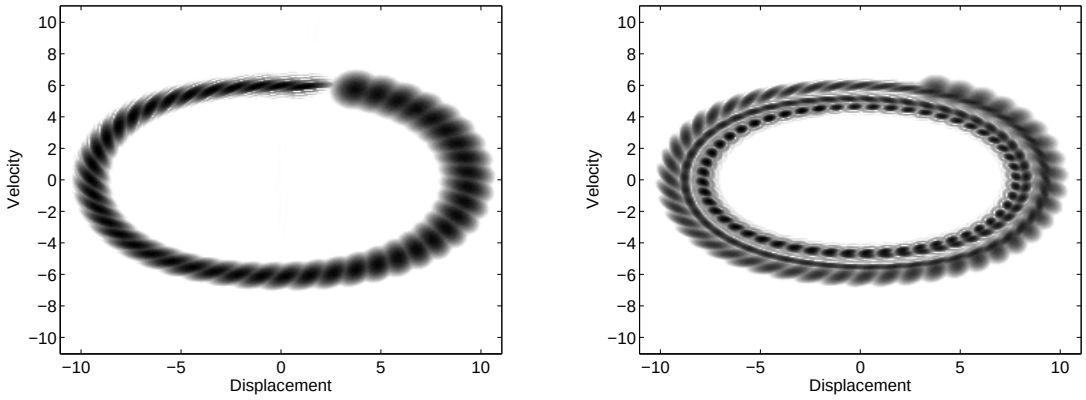


Figure 1.2: Two plots of the transient behaviour for the response PDF for equation (1.32) with $\omega = 0.6250$.

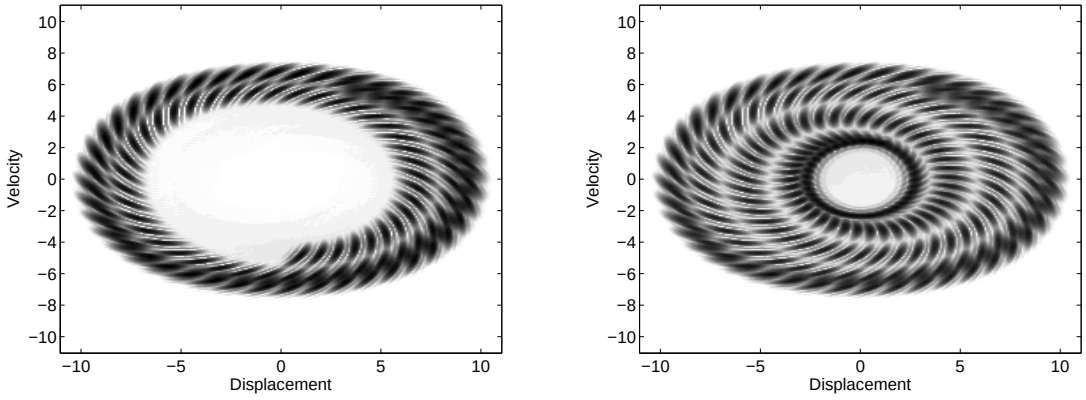


Figure 1.3: Two plots of the transient behaviour for the response PDF for equation (1.32) with $\omega = 0.8125$.

is shown for four different values of ω, the left side plots include fewer iterations than the right side plots.

The PI simulations replicate the behaviour of the deterministic system for both the lower and higher values of ω_f, i.e. Figures 1.2 and 1.5, corresponding to $\omega = 0.6250$ and $\omega = 1.1875$, accordingly. However, in Figure 1.4, where there should be two co-existing solutions, only the one with low amplitude appears, in spite of the specifically chosen initial value. The transient approximately follows the high amplitude solution for about one period, which prolongs the time to reach the steady state solution compared to initial points out of phase with this solution or with very different amplitude. Also the cases where only one deterministic solution exists seems to have a shorter transient. This indicates that the high amplitude solution is sensitive to perturbations, and possibly that there are too few initial points that end up at this stable solution. For $\omega = 0.8125$, only a high amplitude solution exists for the deterministic system. The solution with noise in Figure 1.3 indicates

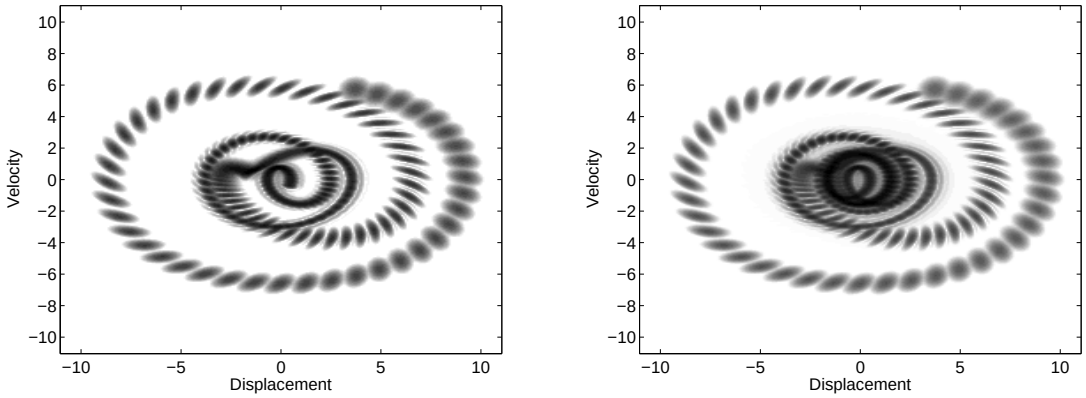


Figure 1.4: Two plots of the transient behaviour for the response PDF for equation (1.32) with $\omega = 1.0000$.

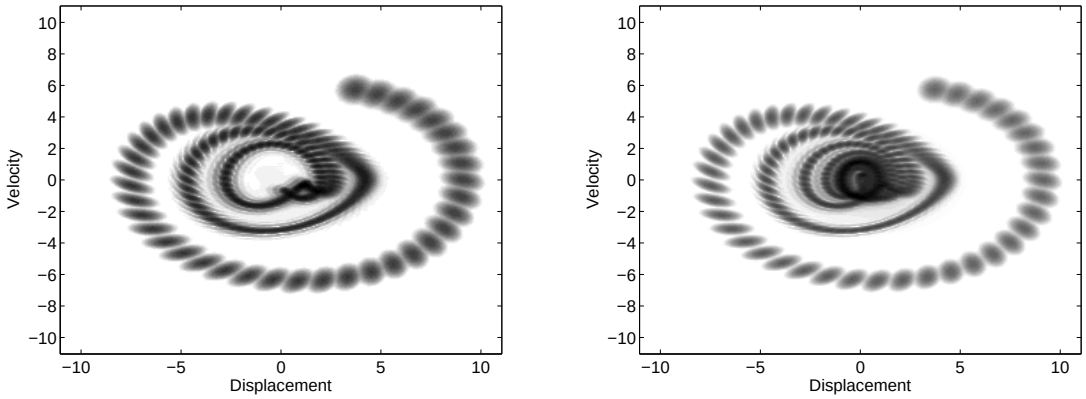


Figure 1.5: Two plots of the transient behaviour for the response PDF for equation (1.32) with $\omega = 1.1875$.

that a much lower period becomes stable for this case. This system also shows a quite abrupt shifting of phase, which results in one or a few quite sharp turns in the transient's path in the state space. This phase shift corresponds to a change between the two stable solutions of the deterministic system.

It should be stated that the reason a solution dissappears could be due to the numerical discretization. The time stepping method is not exact, but this error will be the same for most algorithms for the deterministic system. If the attractor for some solution is very narrow some places, it might not be well represented by the numerical grid, and therefore dissappear. This is almost equivalent to the solution being sensitive to perturbations. Another explanation is that the added noise will make the parametric picture from Figure 1.1 more smooth. If this is the case, the smoothening would in this case have a dramatic effect at some parameter values, as the deterministic system is discontinuous.

Chapter 2

Method and approximations

2.1 Time discretization and schemes

Writing an SDE or an ODE in discretized time is really an approximation of the original dynamical system by a new discretized dynamical system one.

To simplify the notation in this chapter, consider first the one-dimensional SDE from equation (1.9). This will be discretized in time as a forward equation using transition functions $\widetilde{a}$ and $\widetilde{b}$ for the deterministic and stochastic parts respectively, $t = t' + \Delta t$ and $X_(k+1)$ and $X_{(k)}$ approximating X_t and $X_{t'}$, respectively.

$$X_{(k+1)} = X_{(k)} + \widetilde{a}(X_{(k)}, t)\Delta t + \widetilde{b}(X_{(k)}, t)\Delta t. \tag{2.1}$$

Instead of a continuous variable t, an iterative mapping $x_{(k)} \to x_{(k+1)}$ is obtained, for integer values k. This map could also depend on t. A few ways to select the functions $\widetilde{a}$ and $\widetilde{b}$ will be discussed in this chapter, along with results and potential problems, and then finally generalized to the higher order or multi-variable case. For some schemes, $\widetilde{a}$ could depend on b, and $\widetilde{b}$ on a. Observe that the functions $\widetilde{a}$ and $\widetilde{b}$ also potentially could involve subdividing the major time step Δt, and also involve combinations of different numerical methods. This idea will be explored a bit in section 2.1.5, and when time-stepping is applied to PI simulation of impact problems and signum nonlinearities in appendices C, D, E, and F.

Discussing many numerical time-stepping schemes in detail is beyond the scope of this text. Kloeden and Platen, [28], give a detailed survey that covers the major part of the book. including applications and examples. A time-stepping scheme can be explicit or implicit, where the explicit is ready to use, while the implicit requires solving some (often non-linear) equation. The latter is typically needed for stiff problems, where the system gives solutions with very different time scales. This is illustrated for linear ODEs in most introductory books for numerical ODE methods, also [28]. An approved definition of stiffness for non-linear systems seems to be missing, one proposal is found in [6]. Here, the selected systems studied are non-stiff, hence explicit methods should suffice. Next, a numerical scheme can be strong or weak. The scheme is developed based on some criterion

for the convergence of the approximants $X_{(1)}, X_{(2)}, \ldots, X_{(k)}$ to the true X_t as $k \to \infty$, $\Delta t \to 0$, $t = k\Delta t = \text{const}$. For *weak convergence*, the criterion is that for any test function $g : \mathbb{R} \to \mathbb{R}$,

$$\begin{aligned} \lim_{k\to\infty} \int_{-\infty}^{\infty} g(x)\, \mathrm{d}F_{X_{(k)}}(x) &= \int_{-\infty}^{\infty} g(x)\, \mathrm{d}F_X(x), \\ \lim_{k\to\infty} \int_{-\infty}^{\infty} g(x) f_{X_{(k)}}(x)\, \mathrm{d}x &= \int_{-\infty}^{\infty} g(x) f_X(x)\, \mathrm{d}x. \end{aligned} \tag{2.2}$$

Basically, this means that the cumulative probability function F for the approximant equals the cumulative probability function for the solution almost everywhere. The second line assumes that the PDFs also exist, which is implicitly assumed here. Strong convergence is based on criterions that $X_{(k)}$ should approach X_t, either with probability one, or in mean square. In this project, the aim is to compute PDFs of the process. Hence, it seems reasonable to focus on the weak convergence time stepping schemes. However, there should be no problem in adapting strong and/or implicit methods in the numerical algorithm.

As we are concerned with stochastic differential equations, the time step length for the deterministic case is limited by how well we can follow the stochastic part of the dynamics. While studies of deterministic systems, i.e. to compute Poincaré maps, requires a very high accuracy, path integration seems to perform well also for lower order methods. In addition, the (total) time step must be the same for all grid points. However, some systems will perform much better with some time stepping methods than others. The most obvious reason is due to the properties of the deterministic system, like high stiffness, strong sensibility to initial values, high curvature of solution trajectories in the state space, or abrupt changes in the underlying vector field. These properties could hold for just some parts of the solution space, and still be problematic. For some very special problems, the interplay between the stochastic and deterministic parts of the system makes approximation by simple discretized versions difficult, and for multiplicative or parametric noise, the $b_j(x)$ functions in equation (1.12) could cause the problems. These latter cases will not be discussed in this work.

2.1.1 Time step

In Johnsen's thesis, [20], the effect of the time step for some systems have been studied for path integration. The main result is that a fraction of the "characteristic time step" of the system gives a reasonable error-control for one-dimensional problems based on the Stratonovic SDE.

$$\begin{aligned} \Delta t_{\max} = K t_{\text{char}} &= K \min_{x \in \{x : f(x) \le 10^{-6}\}} \frac{b^2(x)}{\widehat{a}(x)^2}, \\ \widehat{a} &= a(X_t, t) + \frac{1}{2} b(X_t, t) \frac{\partial b}{\partial x}(X_t, t), \end{aligned} \tag{2.3}$$

is the estimated maximal time step $\Delta t_{\max}$ for equation (1.9), and this is extended to more dimensions. Further, a value $K = 0.3$ is proposed based on numerical simulations on

systems with a known stationary solution.

However, this is only studied for one- and two-dimensional problems with a very fine mesh, where reduced time step always gave a better result. This is stated explicitly. Johnsen also observes that the interpolation error at each iteration will be the same regardless of the time step, and thereby give a larger total error for shorter time steps, as more iterations are needed to reach stationary state. In practice, especially for high-dimensional problems, the mesh has to be coarse for the simulation to run efficiently, so the time step cannot be chosen arbitrarily small.

It is expedient to state that equation (2.3) is not a final answer to the stability boundary for any system and any discretization. For example, it will be shown in section 4.2 that further approximations in the integration method could require even shorter time steps. As stated in the previous section, properties of the vector fields, especially in the state-space variables where noise does not enter explicitly, could also require a smaller integration time step.

2.1.2 Exact deterministic time-stepping

All linear and some non-linear ODEs can be solved analytically, so that one directly can compute the short time flow function ϕ for the deterministic system.

$$\mathbf{x}_{(k+1)} = \phi(\mathbf{x}_{(k)}, t; \Delta t).$$

It is not obvious that this is preferred even if such a function exist, especially if evaluating ϕ requires many operations or evaluations of non-linear equations, or if ϕ changes behaviour in a non-trivial manner when parameters in the ODE is varied.

For some linear systems, one can also solve equation (2.1.2) for Δt when $\mathbf{x}_{(k+1)}$ and $\mathbf{x}_{(k)}$ are known or partly known.

2.1.3 Euler methods

The simplest discretization procedure that can be adopted for a numerical solution of the SDE (1.12) is the basic version is the *Euler-Maruyama* approximation,

$$\mathbf{X}_{(k+1)} = \mathbf{X}_{(k)} + a(\mathbf{X}_{(k)}, t')\Delta t + \sum_{j=1}^{r} b_j(\mathbf{X}_{(k)})\Delta W_{t'}^j. \tag{2.4}$$

Considering only the deterministic part of the SDE (1.12), equation (2.4) reduces to the *Euler* approximation $\mathbf{x}_{(k+1)} = \mathbf{x}_{(k)} + a(\mathbf{x}_{(k)}, t')\Delta t$. This approximation is accurate to order $\mathcal{O}(\Delta t^2)$.

With one additional term, and skipping the trival indeces of a and b, the *Milstein scheme* reads

$$\mathbf{X}_{(k+1)} = \mathbf{X}_{(k)} + a\Delta t + \sum_{j=1}^{r} b_j \Delta W_{t'} + \frac{1}{2}\sum_{j=1}^{r} \left(b_j^\top (\nabla b_j^\top)\right)^\top \left((\Delta W_{t'}^j)^2 - \Delta t\right). \tag{2.5}$$

The vector ∇ and the matrix $\nabla b_j^\top$ should be written out explicitly for clarification.

$$\nabla = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_n}\right)^\top,$$
$$\nabla b^\top = \begin{bmatrix} \frac{\partial b_1}{\partial x_1} & \frac{\partial b_2}{\partial x_1} & \cdots & \frac{\partial b_r}{\partial x_1} \\ \frac{\partial b_1}{\partial x_2} & \frac{\partial b_2}{\partial x_2} & \cdots & \frac{\partial b_r}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial b_1}{\partial x_n} & \frac{\partial b_2}{\partial x_n} & \cdots & \frac{\partial b_r}{\partial x_n} \end{bmatrix}.$$

If the b_js are constant, all the derivatives, and thereby also the last term, disappears. This means that for additive noise, there is no difference between the Milstein and the Euler-Maruyama schemes. Otherwise, an important fact to notice is that the extra term in equation (2.5) also includes a deterministic term. This corresponds to the conversion term between the Itô and Stratonovich SDEs in equation (1.11).

2.1.4 Runge Kutta methods

To improve the accuracy of the discretization process in following the evolution in time of the deterministic part of the system, a Runge-Kutta approximation could be implemented. These are higher-order schemes that in principle can be viewed as an advanced subdivision of Δt into smaller Euler steps.

The second order Runge-Kutta method is accurate to order $\mathcal{O}(\Delta t^3)$, and is for an ODE written

$$\begin{aligned} \mathbf{x}_{(k+1)} &= \mathbf{x}_{(k)} + a(\boldsymbol{\xi}, t' + \frac{\Delta t}{2})\Delta t, \\ \boldsymbol{\xi} &= \mathbf{x}_{(k)} + a(\mathbf{x}_{(k)}, t')\Delta t. \end{aligned} \tag{2.6}$$

The intermediate value $\boldsymbol{\xi}$ is identical to the Euler step. As stated earlier, Writing equation (2.7) as $\mathbf{x}_{(k+1)} = \widetilde{a}(\mathbf{x}, t)\Delta t$, noise could be included by the trivial Maruyama addition to the Euler equation (2.4).

$$\mathbf{X}_{(k+1)} = \mathbf{X}_{(k)} + \widetilde{a}(\mathbf{X}_{(k)}, t')\Delta t + \sum_{j=1}^{r} b_j(\mathbf{X}_{(k)})\Delta W_{t'}. \tag{2.7}$$

Formally, this has the same weak order as the Euler-Maruyama approximation in equation (2.4). Again, as in equation (2.5), the additional term from the Milstein scheme can be added to equation (2.7) to obtain a formal 1.5 order scheme.

The fourth order Runge-Kutta method, which is accurate to order $\mathcal{O}(\Delta t^5)$, is normally

written

$$\begin{aligned}
\mathbf{x}_{(k+1)} &= \mathbf{x}_{(k)} + \frac{\Delta t}{6}(\mathbf{q}_{\mathrm{A}} + 2\mathbf{q}_{\mathrm{B}} + 2\mathbf{q}_{\mathrm{D}} + \mathbf{q}_{\mathrm{D}}),\\
\mathbf{q}_{\mathrm{A}} &= a(\mathbf{x}_{(k)}, t'),\\
\mathbf{q}_{\mathrm{B}} &= a(\mathbf{x}_{(k)} + \mathbf{q}_1\frac{\Delta t}{2}, t' + \frac{\Delta t}{2}),\\
\mathbf{q}_{\mathrm{C}} &= a(\mathbf{x}_{(k)} + \mathbf{q}_2\frac{\Delta t}{2}, t' + \frac{\Delta t}{2}),\\
\mathbf{q}_{\mathrm{D}} &= a(\mathbf{x}_{(k)} + \mathbf{q}_3\Delta t, t' + \Delta t).
\end{aligned} \tag{2.8}$$

Numerically, the author has found that the following algorithm, which formally is identical to equation (2.8), gives a slightly better performance at the same CPU-time. The performance is measured both in terms of how close a step backwards in time and subsequently a step forwards is to the original starting point, and how well simple exponential test equations are followed.

$$\begin{aligned}
\mathbf{x}_{(k+1)} &= \frac{1}{3}(\boldsymbol{\xi}_{\mathrm{A}} + \boldsymbol{\xi}_{\mathrm{B}} + \boldsymbol{\xi}_{\mathrm{D}}),\\
\boldsymbol{\xi}_{\mathrm{A}} &= \mathbf{x}_{(k)} + a(\mathbf{x}_{(k)}, t')\frac{\Delta t}{2},\\
\mathbf{q}_{\mathrm{B}} &= a(\boldsymbol{\xi}_{\mathrm{A}}, t' + \frac{\Delta t}{2})\frac{\Delta t}{2},\\
\boldsymbol{\xi}_{\mathrm{B}} &= \mathbf{x}_{(k)} + 2\mathbf{q}_{\mathrm{B}},\\
\widetilde{\boldsymbol{\xi}}_{\mathrm{B}} &= \mathbf{x}_{(k)} + \mathbf{q}_{\mathrm{B}},\\
\boldsymbol{\xi}_{\mathrm{C}} &= \mathbf{x}_{(k)} + 2a(\widetilde{\boldsymbol{\xi}}_{\mathrm{B}}, t' + \frac{\Delta t}{2})\Delta t,\\
\boldsymbol{\xi}_{\mathrm{D}} &= \boldsymbol{\xi}_{\mathrm{C}} + a(\boldsymbol{\xi}_{\mathrm{A}}, t' + \Delta t)\Delta t.
\end{aligned} \tag{2.9}$$

The purpose of this work is, as stated above, to work with non-linear systems with additive noise. Experiments have shown that the most important is to approximate the deterministic terms correctly. Equation (2.4) will therefore be replaced by what will be referred to as the Runge-Kutta-Maruyama (RKM) approximation

$$\mathbf{X}_t = \mathbf{X}_{t'} + \widetilde{a}(\mathbf{X}_{t'}, t')\Delta t + b(\mathbf{X}_{t'})\Delta W_{t'}. \tag{2.10}$$

There are unresolved problems concerning Runge-Kutta methods and other time-stepping methods, especially for non-linear problems. The major ones are listed in [7], and include the lack of a proper definition of non-linear stiffness, possible lack of stability, and the counter-intuitive effect of increased error with shorter time steps. Further, even if a Runge-Kutta scheme converges, one could be observing a so-called ghost attractor, which only occurs due to the numerical discretization. This is an equivalent statement to the one in section 1.3.5, that the discretized system by itself is a new system with different properties than the continuous one.

2.1.5 Taylor methods

An ODE based on a sufficiently smooth vector field a, can be differentiated repeatedly with respect to time.

$$\begin{aligned}
\dot{x} &= a(x,t),\\
\ddot{x} &= \frac{\mathrm{d}a(x,t)}{\mathrm{d}t} = \frac{\partial a(x,t)}{\partial t} + \frac{\partial a(x,t)}{\partial x}\dot{x} = \frac{\partial a(x,t)}{\partial t} + \frac{\partial a(x,t)}{\partial x}a(x,t),\\
x^{(3)} &= \frac{\mathrm{d}^2 a(x,t)}{\mathrm{d}t^2} = \frac{\partial^2 a}{\partial t^2} + 2\frac{\partial^2 a}{\partial x \partial t}a\frac{\partial^2 a}{\partial x^2}a^2 + \frac{\partial a}{\partial t}\frac{\partial a}{\partial x},\\
&\ldots,
\end{aligned}$$

where the arguments of $a = a(x,t)$ have been skipped for simplicity. Using this relation, the solution of the ODE after a time step Δt can be expressed as a Taylor series, here written in just one dimension for simplicity;

$$\begin{aligned}
x_{(k+1)} &= x_{(k)} + \sum_{j=1}^{\infty} \frac{1}{j!} x_{(k)}^{(j)} (\Delta t)^j \\
&= x_{(k)} + a(t, x_{(k)})\Delta t + \frac{1}{2}\left(\frac{\partial a}{\partial t}(x_{(k)}, t) + \frac{\partial a}{\partial x}(x_{(k)}, t)a(x_{(k)}, t)\right)(\Delta t)^2 + \ldots.
\end{aligned} \tag{2.11}$$

The main disadvantage of the Taylor method is that all the derivatives must be computed. However, for a number of algebraic expressions in $a(x,t)$, the terms in the Taylor series either eventually becomes zero (polynomials) or ends up in periodic patterns or geometric series (exponentials, trigonometric functions, hyperbolic functions, square roots etc.) This creates a possibility of creating efficient computer code to evaluate many terms efficiently, and only requires that the ODE is compiled to create such a code in advance. Automatic differentiation, typically applied to Taylor series, can be found a number of places in the litterature, e.g. [46] and chapter ten of [37]. Good software using this technique can be found on the internet, e.g. [9] and [21].[1]

The advantage of the Taylor method is that the error in following the dynamic behaviour of the system can be studied and controlled in runtime for each grid point and each time step independently, without extra CPU-time. This is because the Taylor method is easily performed with different order, and the error is expressed by the contribution of the last term. It is then possible to specify a relative or specific acceptance level of this error, and ensure that this error is constant over the whole grid, while you in general are limited to the general order of the method, expressed by the time step. For many systems, the Taylor series in equation (2.11) has a finite length, or the infinite sum can be analytically described. Then the approximant coincides with the exact time stepping method from section 2.1.2, and the error is zero.

Note that for SDEs, one still has to perform shorter time steps and handle the stochastic terms to obtain a good solution. In [28] both explicit and implicit Taylor series for SDEs are

[1] See also http://www.autodiff.org/ for tools, papers etc. on automatic differentiation.

developed, and they are again truncated in a number of different ways to obtain different strong and weak schemes. The main problem with PI, is again that the advanced schemes introduce non-Gaussian noise and noise entering into more state-space variables. This complicates the implementation and increases CPU-time.

Therefore, implementations of PI using Taylor are based on just using Maruyama or Milstein terms for the stochastic parts of the SDE. The implementation has been based on the program "Taylor" by Jorba and Zou, [21]. However, for the systems studied here, the Runge-Kutta methods performed just as well for the time steps needed, and because the "Taylor" program required recompilation for every change of parameters in the SDE, it became too cumbersome for many studies. However, the idea is good, and the code available for systems where higher time-stepping accuracy is an issue.

2.1.6 The trapezoidal rule

The objective is to integrate the SDE. All the step methods above have been based on the idea of following the deterministic trajectories through the time step. However, it is also possible to increase the accuracy of the Euler method by using the trapezoidal integral approximation with one step. This is especially relevant for problems of the form $\dot{X} = Y$, so assume this in the following. The one-step numerical scheme for X becomes

$$X_{(k+1)} = X_{(k)} + \frac{1}{2}(Y_{(k+1)} + Y_{(k)})\Delta t. \tag{2.12}$$

Note that this idea is clearly different from the derivation of the Runge-Kutta schemes, since $Y_{(k+1)}$ is the current grid point and not the integration variable. In other words, this is a simple implicit method. Since $Y_{(k+1)}$ normally is unknown, this must be found by one of the other time-stepping schemes, and this could involve also calculating a value for $X_{(k+1)}$. In spite of using a high order Runge-Kutta method, discarding this value of $X_{(k+1)}$ and replacing it by the trapezoidal rule from equation (2.12) could reduce the discretization error.

Similarly, if a multistep method is used, intermediate values for Y between $Y_{(k)}$ and $Y_{(k+1)}$ could, if suitably adapted, be used to set up a trapezoidal rule with a higher partitioning, or other similar schemes like the Simpson integration rule.

2.2 Other numerical methods of solving SDEs

This is not a survey of numerical methods for solving SDEs, but a brief description of two methods that are mentioned and used in the thesis and the appendices. Some variations on the Path Integration method will be described in section 4.5.

2.2.1 Markov Chain Monte Carlo simulation

This is a brute-force method, quite easily implemented in a direct way, creating samples of the process for some time histories. A very positive feature is that only the statistics of

interest needs to be recorded from the sampling algorithm. If this has lower dimensionality than the system but can be estimated from any sample space, e.g. a mean value or a standard deviation, Monte Carlo has a clear advantage.

The main challenges are simulation of low-probability events, i.e. the tails, and estimating a sufficiently smooth or accurate probability density from the samples obtained from the simulations. A high-dimensional joint probability density will require extremely many samples to achieve some desired spatial accuracy. Note that also a conditional density, say $f(x|y = y_0)$ for continuous random variables x and y, generally requires the same number of samples as the joint density $f(x, y)$.

When performing a Monte-Carlo simulation on a dynamical system, it is expedient to make the following considerations. First, one will need one or more starting points, that often will be arbitrarily or randomly selected. From the starting point, there will be a transient time where the dynamics should converge towards the stationary behaviour. The behaviour in this transient should not be included in estimates of variance or grid selection, unless the starting point is defined to be an initial value for the SDE.

Independence of samples later on in the simulation requires a waiting time between each sample at least equal to the length of the transient time. However, the sampling rate depends on what statistic is of interest. Especially extreme values of the response will be found more efficiently when sampling at every time step. For sufficiently large levels, one can assume that each upcrossing of this level is independent of the others. With a fixed sampling interval, the probability of observing the extreme value is smaller.

Similarly, Monte-Carlo simulations must be properly adapted to handle time dependence, and special considerations can be made for efficient estimation of e.g. the variation of the response process.

Monte Carlo has been used in this project, partly to visualize the process before running the main numerical method, partly to determine the best grid for the computational domain, and partly to estimate response PDFs or other statistics that can be compared to the results from the main program. If an initial value is given with the SDE in question, this should be the starting point for the Monte-Carlo simulation. Otherwise, some arbitrary starting point(s) must be chosen. In the latter case, this point might not be on the manifold of the stable solution, and the estimated PDF should not be influenced too much by the method of choosing the initial value. Therefore, the system should be run for a while so that most of the initial transitions (if any) die out, and so that the sample of the state space variables is independent of the starting point. After sampling this point, the same amount of time is needed to obtain a new independent sample.

However, the Monte-Carlo simulation can give more information than just independent samples for a PDF. Here, the Monte-Carlo simulation is used to obtain information on where the computational domain should be. If this is based on extreme values from the MC-simulation, every step should be checked if it is a new extremal value. Also other measures of the total variation of the system could be improved even with non-independent samples.

When the total timespan of the MC-simulation is large compared to the dynamics of the system, the lack of independence between consecutive samples is no longer a problem.

2.2.2 Moment equations

For some applications, the topic of interest is to compute the moments of the response of a stochastic system. The moment equations provide an exact theoretical formula for this, as a linear system of ODEs. The first problem encountered is that only special problems have closed form expressions for the moments. The author has worked on this in collaboration with PhD Çağdaş Kafali and Professor Mircea Grigoriu, creating a numerical software that automatically generates sequences of moment equations and solves them.[8] For most problems however, the linear system has infinite range, and the solution for a moment of some order depends on moments with higher order so that the system cannot be solved iteratively.

This means that one needs to truncate the number of equations by replacing higher-order expressions by some approximants. This is referred to as a closure technique. A frequently used truncation method is Gaussian closure, where high order moments are replaced by formulas for the respective Gaussian moments is frequently used, as in [61], [60], and [58]. This could be a crude approximation, and the linearity of the system is destroyed. A second problem is that high and low order moments' values will be of very different scales, so that the resulting ODE system will be almost singular. Solving the ODEs numerically could require a time-stepping method that is stable over a large range of time scales, and any singular value decomposition method will require high precision. Finally, the moment equations are based on and limited to Gaussian White Noise processes, though there are possibilities of applying nonlinear filter equations to obtain heavier tails than the Gaussian.

Yu, Cai and Lin (1997) [61] numerically solve the moment equations for a 2D SDE with a Gaussian closure over each time step for each grid point, and create a bivariate Gaussian TPD. This method is also used later by e.g. by Xie, Xu and Cai (2005) [58]. The reason this idea is not persued further in this project, is that moment equations easily become numerically unstable, as the parameter connecting lower and higher order moments are in very different scales, the Gaussian closure is too crude for many problems, and especially this is not valid for discontinuities, piecewise systems etc., and finally this approach is not as easily generalized to other noise processes.

Chapter 3

Interpolation methods

3.1 Interpolation overview

The choice of interpolation routine is very important in path integration. Christian Skaug [50] has shown that many problems will not converge to a reasonable approximation in two dimensions using only a bi-linear interpolation, even if the grid resolution is extremely high. The reason for this is that a linear interpolant consistently under-estimates peaks and other areas where the function has a negative second-order derivative, and over-estimates areas where the second-order derivative is positive. Therefore, in the last years, cubic spline interpolation has been chosen instead. This seems to work well for a number of systems. However, instabilities have frequently been observed, and this easily ruins the final result. These problems have not been documented widely, though most of the thesis of Vibeke Moe [34] discusses ways to overcome interpolation problems, although the discussion in section 5.4 will show that the main proposals in there, although intuitively reasonable, is generally not recommendable.

In addition, the interpolant is twice differentiable, which seems reasonable for probability density functions. Unfortunately, the cubic spline interpolation has some major drawbacks. Computing spline coefficients and evaluating the spline is a minor problem in systems with one or two dimensions, partly since standard packages are available (e.g. Fitpack). In higher dimensions, one must know at least the basic coding for tensor-product forms to attain a full interpolation. There are also three-dimensional interpolation codes in Fitpack, but they are slow, only approximative, or based on biquadratics, weighting, and least-squares methods. A simple tensor-product method with B-splines does not exist. The most significant problem is, however, that the interpolating spline will have unreasonable oscillations in intervals close to a large jump in the interpolation values. This effect will be illustrated in secton 5.4.1. This is more likely to occur in higher dimensions with low resolution, which might be why standard B-spline packages do not exceed two dimensions. In this thesis some methods to remove or reduce such oscillations are proposed, including locally increasing the grid resolution, using a more sophisticated low-oscillating and non-negative interpolation routine, or returning back to linear interpolation in some

"problematic" areas. To exercise a better control with the result of the interpolation, this project included implementing all parts of the interpolation code explicitly.

Parabolic splines do not seem to be nearly as widely used as the cubic splines. This could be because the construction is a bit less intuitive, one has less freedom to enforce endpoint conditions, and the interpolant is just once differentiable. The CPU-time in lower dimensional systems is practically the same as for cubic splines. However, because the support is smaller, the number of grid points needed in a high-dimensional space to construct the interpolant can be signifcantly reduced compared to the cubic case. Since the parabolic splines can be seen as an intermediate method between the linear and the cubic, one could expect the stability to be higher.

The B-splines methods will be presented here first in a general case as described in C. de Boor (1978) [10]. Then some simplified algorithms for a uniform grid is presented, including interpolation over a circle or angle. Tension splines will be briefly discussed as another method between linear and cubic interpolation.

In Naess and Mo 2004 [41] and in section 6.5, a conditional path integration method is suggested. Here, a the high-dimensional joint probability density of the state space variables is pointwise divided by the marginal density for some variables. This way, a more flat surface is obtained, which should be more suitable for linear interpolation in these variables, possibly with high accuracy even for a very coarse grid. The interpolated value of the pdf is then obtained by multiplying the interpolated values in the conditional density with the given marginal density. Unfortunately, this method also has one major drawback. In many problems, a cross section of the pdf could be very different from the marginal density, i.e. the conditional density could easily be less flat than the pdf. This causes a large interpolation error, which is typically difficult to detect and avoid. In addition, if the marginal density is unknown, this must also be calculated which requires some extra computational effort. One way to avoid this latter problem is to substitute the marginal density with an approximation, especially by assuming a marginal Gaussian distribution, however this seems to give a bias towards the Gaussian approximation. An ideal test case for this method is equation (1.25), which is Gaussian in y, but not in x, and with an analytical known solution.

Finally, interpolation in many dimensions will be described using tensor-product forms. It will also be shown that the interpolation method can be chosen independently in different variables, sometimes requiring that the sequence of operations are performed in the right order.

The new work in this chapter is the development and study of low-oscillating non-zero splines based on Bézier splines, automatic detection of unstable cubic and rectangular B-splines and automatic adjustment in such areas, and generalizing code so that all possible combinations of interpolation methods can be easily attained.

3.2 Knot sequences

> "We must define a knot, because if we do not, then we do not know what is a knot and what is not a knot."
>
> John Basias[1]

A *knot sequence* is an ordered sequence of numbers $\boldsymbol{\tau} = \{\tau_k\}_{k\in\mathbb{Z}}$ such that $\tau_{k+1} \geq \tau_k$ for all k. For the implementations described in this thesis, the knot sequence will have finite length and cover a finite interval $[p, q]$, i.e. the numbering will be $k \in \{1, 2, \ldots, K\}$ with $\tau_1 = p$ and $\tau_K = q$, $p < q$, so K is the number of knots.

Further, the interval $I_k = [\tau_k, \tau_{k+1}]$ with $k \in \{1, 2, \ldots, K-1\}$ will be denoted a *knot interval*. This notation is also valid for angular variables, which will be discussed in section 3.5. An angle θ is associated with every angle $\theta + n2\pi$, $n \in \mathbb{Z}$, so any point on the circle can be chosen as a starting point, and the interval will be chosen counter-clockwise with a length slightly smaller than 2π. However, one additional interval will be added, namely $I_K = [p, p + 2\pi]$, or equivalently $I_K = [q - 2\pi, p]$. Here, all integers k will be allowed, identifying $I_{K+1} = I_1$ $I_{K+2} = I_2$ etc. For knot intervals, the valid range of the index k will often be skipped due to the different possible applications.

A rigorous discussion of spline interpolation in general requires a definition of the intervals as right or left continuous. That is, one should define which interval the knot points belong to. Above, the intervals are overlapping. However, as long as the spline surface is continuous, and sufficiently smooth if a derivative is needed, this has no practical effect. Continuousness and smoothness is ensured by using sufficiently high order polynomials for the interpolating function(s), and require that all knots, except possibly those at p and/or q, are separated, i.e. that the knot sequence is strictly increasing between p and q. These requirements have been implemented for the work presented, and are therefore assumed from here on. For a more general discussion of multiple knots and discontinuities in spline interpolation, see e.g. [10] or [4].

The interpolation will be performed on some sequence of points $\{(x_l, v_l)\}_{l\in\mathbb{Z}}$, that will be denoted *nodal points*, and each v_l is a *nodal value*. In the practical situation of this project, the sequence $\{x_l\}_{l\in\mathbb{Z}}$ is a fixed partition of some interval of interest for some $l \in \mathbb{Z}$, but extended to the real line. These x_l are the *grid points*. The special case of periodic end point conditions, i.e. interpolation on a circle, has a different extension, which is discussed separately.

3.3 B-spline interpolation on a generalized grid

B-splines can be defined by a recursive formula. The zero degree B-splines are piecewise horisontal lines, with discontinuities at the knot points. To avoid these discontinuities, it

[1] Quote from a knot-theory course in the 2000 Hampshire College Summer Studies in Mathematics. Found on http://www.toroidalsnark.net/knotpuns.html, accessed August 2007. Note that the knots in the quote are not spline knots.

is just as simple to start the recurrence at the first degree polynomial interpolation, that is linear interpolation, where the basis functions for $k = 1, \ldots, K-2$ are

$$B_{k,2}(x) = \begin{cases} \frac{x-\tau_k}{\tau_{k+1}-\tau k}, & \text{for } x \in I_k, \\ \frac{\tau_{k+2}-x}{\tau_{k+2}-\tau_{k+1}}, & \text{for } x \in I_{k+1}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

The number 2 in $B_{k,2}$ refers to the *order* of the B-spline, which is the degree of the polynomials involved plus one. Formally, the knot sequence affects the shape of the basis functions, so the basis functions should be denoted $B_{k,j,\boldsymbol{\tau}}$. However, since the knot sequence will be kept constant, this index is skipped.

The recurrence relation for B-splines is written

$$\begin{aligned} B_{k,j}(x) &= \omega_{k,j}(x)B_{k,j-1}(x) + (1 - \omega_{k+1,j}(x))B_{k+1,j-1}(x), \\ \omega_{k,j}(x) &= \frac{x - \tau_k}{\tau_{j+k-1} - \tau_k}. \end{aligned} \tag{3.2}$$

It should be stated that there are a number of equivalent ways to define the B-spline series, especially with the single multiplicity of knots that is assumed here.

The basis for *parabolic B-splines* is achieved by inserting $j = 3$ in equation (3.2), and *cubic B-splines* are again computed from these with the same formula using $j = 4$. These interpolation schemes have order 3 and 4, i.e. degree 2 and 3, respectively.

When a proper knot sequence is chosen, and the basis functions are calculated, everything is ready for interpolation. As stated above, the interpolation is performed for some set of nodal points (x_l, v_l), $l = 1, 2, \ldots, M$, and should be done so that

$$\sum_{k=1}^{K} c_k B_{k,j}(x_l) = v_l \tag{3.3}$$

for every index l and some unknown constants c_k, $k = 1, \ldots, K$, which are denoted *spline coefficients*. The number of grid points, M, that ensures a unique and stable solution in c_k for this problem, depends on the order of the method, and the end-point condition. The different implemented possibilities uses $M = K$, and these will be discussed below. Basically, one grid point for each basis function is a good rule. For the further discussion on B-splines, the linear system (3.3) will be written in a matrix form,

$$S\mathbf{c} = \mathbf{v}, \tag{3.4}$$

where $\mathbf{c}$ contains the spline coefficients, $\mathbf{v}$ contains the nodal values to be interpolated, and $S(l, k) = B_{k,j}(x_l)$. The grid points must also be placed sufficiently evenly relatively to the knot sequence for this system to be solvable.

Finally, the resulting interpolating approximative function is obtained as

$$\widehat{f}(x) = \sum_{k=1}^{K} c_k B_{k,j}(x). \tag{3.5}$$

From the criterion on the B-spline coefficients in equation (3.3), it is immediately clear that $\widehat{f}(x_l) = v_l$ for $l = 1, \ldots, M$.

The simplest B-spline interpolation (within the restriction of continuity) is the order 2 B-splines from equation (3.1). When this is used, the knots and grid points should coincide to ensure stability, positivity, and efficiency of the interpolation. Figure 3.1 illustrates some of the problems that can occur if this rule of thumb is not followed.

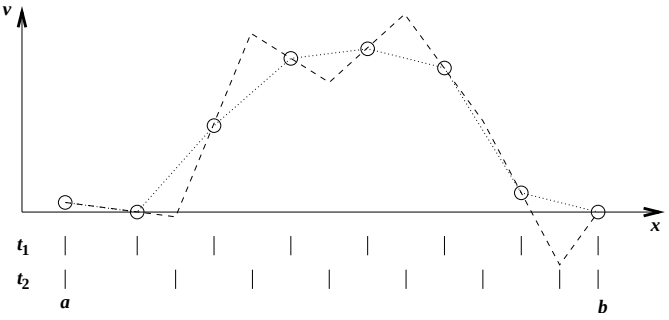


Figure 3.1: 1D interpolation with linear splines. The circles indicates points to interpolate along a uniformly spaced grid, and $\mathbf{t}_1$ and $\mathbf{t}_2$, indicated by vertical bars, are two different knot sequences. The first knot sequence is located at the grid points, and the linear interpolation (dotted) is trivial. With the latter sequence the resulting interpolation function (dashed) attain negative values and gives unreasonable placement of peaks.

Some trial and error indicates that also cubic B-splines easily become unstable if the knots are not at the grid points, as this results in a singular or almost-singular S matrix. In the iterative procedure that this paper is using, this stability problem seems to deteriorate the final result. So for cubic B-splines, every knot in the knot sequence will coincide with a grid point. The maximum for a basis parabolic spline is between τ_{k+1} and τ_{k+2}. The only implemented version using parabolic splines uses a uniform grid sequence, as discussed in section 3.4, together the zero outside end condition, which is described in section 3.3.2. In this case, the basis function is symmetric, and the maximum is centered between the knot points. The most stable interpolation is then obtained placing one grid point in the middle of each knot interval.

The matrix version of equation (3.3) will have a banded structure, where the width of the band around the diagonal only depends on how many grid points that belong to the support of each basis spline. Solving such a matrix equation is most efficiently performed through an LU-factorization, i.e. computing an upper-triangular matrix U and a lower triangular matrix L so that $S = LU$. It can be shown that also L and U will have a banded structure. As long as the grid is kept constant, the LU decomposition will not change, and could be done before iteratively performing interpolation on the system.

3.3.1 The not-a-knot end condition

With M gridpoints, with one knot for each grid point, and a double knot at each end, $M + 2$ conditions are needed to make a cubic B-spline interpolation unique. The M nodal

points to be interpolated are known, but still two conditions remain. These will usually be boundary conditions like the derivative and the curvature at the end points. However, in a general case, such boundary conditions are not known or difficult to use, e.g. if the derivative is infinite. Another option is to require complete smoothness at the second and second last grid point, i.e. continuity also in the third derivative. This is equivalent to removing the knots at these two grid points, and using a knot multiplicity of two at the end points. Due to the removed knot positions, this method is called the *not-a-knot* end condition. It has been shown [11] that this interpolation scheme converges strictly to any smooth function when the mesh size goes to zero.

3.3.2 The zero side end condition

The functions to be interpolated here are PDFs. Since the total mass of a PDF is one, it must asymptotically approach zero from above as the distance to the origin goes to infinity in all directions. Except for some special cases where the PDF has a fixed endpoint in one direction, the decay to zero should be smooth. Since the numerical algorithm requires the domain to be bounded, it is reasonable to assume the PDF is exactly zero outside of this domain, and search for an interpolation method that smoothly decays to zero as one crosses the boundary of the domain from the inside.

3.4 Splines in a uniform grid

When the knot sequence has constant increments $\tau_{k+1} - \tau_k = \Delta\tau$, the grid is uniform. This leads to so-called *cardinal splines*, where all the basis functions all have the same shape and are just translated along the axis. In the litterature cardinal splines are sometimes also called *periodic splines*, referring to this repetative pattern. A basis function can be expressed analytically in some reference domain, and then scaled and translated to give the complete basis of the cardinal splines. This is a computational advantage as this function can be calculated in advance, and also the calculation of spline coefficients can then be done faster than in the general case.

A cubic uniform B-spline is composed of four pieces. Every point except the knots are in the support of four basis functions with one piece each. Translated and scaled to $(0, 1)$, these four pieces are

$$\begin{array}{rcrcrcr} \frac{1}{6}x^3 & & & & & & \\ -\frac{1}{2}x^3 & + & \frac{1}{2}x^2 & + & \frac{1}{2}x & + & \frac{1}{6} \\ \frac{1}{2}x^3 & - & x^2 & & & & +\frac{2}{3} \\ -\frac{1}{6}x^3 & + & \frac{1}{2}x^2 & - & \frac{1}{2}x & + & \frac{1}{6}. \end{array} \tag{3.6}$$

The parabolic uniform B-spline has three pieces. In the reference domain $(-\frac{1}{2}, \frac{1}{2})$ they are

$$\begin{array}{rcrcr} \frac{1}{2}x^2 & + & \frac{1}{2}x & + & \frac{1}{8} \\ -x^2 & & & + & \frac{3}{4} \\ \frac{1}{2}x^2 & - & \frac{1}{2}x & + & \frac{1}{8}. \end{array} \tag{3.7}$$

The trivial linear uniform B-spline has two pieces that in the reference domain $(0, 1)$, namely x and $1 - x$. For all these cases it is simple to verify that the functions add to 1 for all x.

Consider the matrix version of one-dimensional interpolation problem from equation (3.3);

$$\mathbf{v} = S\mathbf{c}, \tag{3.8}$$

where $\mathbf{v}$ are the nodal values to be interpolated, $\mathbf{c}$ are the corresponding spline coefficients, and S is the spline matrix.

Assume that the values in $\mathbf{v}$ tend to zero on the border of the domain, and extend the grid to a few points with zero values in $\mathbf{c}$ as described in section 3.3.2. Now the spline matrix S becomes very simple. For cubic B-splines, equation (3.6) implies that the basis function has a maximum value of 4/6. The two neighbours both have the value 1/6 at the grid point. This gives

$$S = \frac{1}{6}\begin{bmatrix} 4 & 1 & 0 & \cdots & 0 & 0 \\ 1 & 4 & 1 & \ddots & 0 & 0 \\ 0 & 1 & 4 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & 4 & 1 \\ 0 & 0 & 0 & \cdots & 1 & 4 \end{bmatrix}. \tag{3.9}$$

Note that the first nodal points outside the original domain, i.e. the closest parts of the extended grid, by this has nodal values $v_0 = c_1/6$ and $v_{n+1} = c_n/6$. If c_1 and c_n are positive, this makes a small extrapolation feasible.

For parabolic splines, the matrix has the same shape as in equation (3.9), but with 6/8 on the diagonal and 1/8 on the sub- and super diagonals. Note that S in both cases is symmetric and positive definite.

To compute the spline coefficients of equation (3.8) the most efficient way is an LU-decomposition of S. Also this can be done analytically. If the matrix S has a diagonal with value d and ones on the sub and super diagonals, as for parabolic and cubic splines the LU-composition reads

$$\begin{aligned} S &= LU, \\ L(j,j) &= 1, \quad j = 1,, 2, \ldots, M \\ L(j+1,j) &= \frac{q_{j-1}}{q_j}, \quad j = 1, 2, \ldots M-1, \\ U(j,j) &= \frac{q_j}{q_{j-1}}, \quad j = 1, 2, \ldots M, \\ U(j,j+1) &= 1, \quad j = 1, 2, \ldots, M-1 \\ q_0 &= 1, \quad q_1 = d, \quad q_{j+1} = dq_j - q_{j-1}, \quad j = 1, 2, \ldots M, \end{aligned}$$

which is easily proved by evaluating the matrix product of L and U. The diagonal elements of U are the inverse of the elements on the super diagonal of L. The ones does not have to be saved, hence only M numbers need to be saved when pre-computing the LU-factorization.

For large matrices, the formula above could lead to numerical overflow for the q_j. However, it can be shown that

$$\lim_{j\to\infty} \frac{q_{j-1}}{q_j} = \frac{d - \sqrt{d^2-4}}{2}.$$

This convergence is fast enough that the limit value and its inverse can be used instead of the fractions in the procedure above when the counter j is about 20, and thereby avoid overflow. The LU-decomposition can also be performed numerically quite efficiently, and this has been implemented for the general grid. However, if the numerical method requires frequent updates of the grid resolution, the method above will be the faster without lack of accuracy.

3.5 Cardinal splines on a circle

If the variable is an angle, a uniform partitioning will give points separated by $\frac{2\pi}{M}$, and every grid point will have two neighbours. The spline matrix will then contain be non-zero also at the positions $(M, 1)$ and $(1, M)$. The cubic uniform B-spline matrix becomes

$$S = \frac{1}{6}\begin{bmatrix} 4 & 1 & 0 & \cdots & 0 & 1 \\ 1 & 4 & 1 & \ddots & 0 & 0 \\ 0 & 1 & 4 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & 4 & 1 \\ 1 & 0 & 0 & \cdots & 1 & 4 \end{bmatrix}.$$

The LU-factorization will be almost the same as above, except the last row in L and the last column in U will be non-zero. The total storage required is then $2M$.

One could also compute general splines on a circle using the recurrence relation from equation (3.2), and then generate the S matrix and its LU-factorization. This has not been implemented in this project, since no system has motivated a need for a globally higher grid resolution around some angles.

3.6 Tension splines

Tension splines can be expressed as a linear combination of 1, x, e^{wx} and e^{-wx}. This gives a linear interpolation when the tension factor w goes to infinity, and $\lim_{w\to 0^+}$ gives cubic B-splines. The basis can also be chosen as $\sinh(wx)$ and $\cosh(wx)$ instead of the exponential functions.

All the important properties of cubic B-splines, that is pointwise interpolation, smoothness, adding up to one, and the applicable end point conditions, are still retained with this

basis. Further, the construction of uniform splines periodic endpoint conditions follow the same procedure as cubic B-splines, and the LU-factorizations for both the periodic and the non-periodic cases give the same pattern of non-zero elements in the L and U matrices as for cubic splines, just slightly different numbers. Hence, the tension spline LU matrices are tri-diagonal for a uniform grid, so the cpu-time should be basically the same.

To avoid the problem of artificial inflection points or extremal points for a specific interpolation problem, one could use tension splines, and gradually increase tension until a suitable result has been obtained. The author's experience with this idea is however that the problematic interpolation behaviour often persists up to very high tension, at which point the interpolant has the unfortunate property of consistently under-estimating peaks and over-estimating tails like the linear interpolant. Other problems include that a strict criteria for choosing the tension parameter has not been found, and changing the tension parameter requires a full re-calculation of the LU-factorization and spline coefficients. Theoretically one could also use different tensions at different intervals, and still retain a smooth curve. However, every change of tension locally affects the interpolant globally, and now the basis splines also depends on all the tension values globally, so its practically impossible to express then analytically. Both these facts implies a longer cpu-time and much more complicated code. Therefore, the idea of using tension splines to avoid unstable interpolation was abandoned.

3.7 A Bézier-based interpolation method

Here a new interpolation method is presented. The requirements for the method is

1. Pointwise interpolation of all the known function values.
2. Sufficiently fast.
3. Facilitate the use of a more coarse grid in all spatial directions than the cubic spline method.
4. No explicit assumptions on the shape of the density or marginal density (except some degree of smoothness).
5. No systematic under-estimation or over-estimation of peaks or tails.
6. Always return positive interpolation values when interpolating a positive function.
7. Give the possibility of adding boundary values for the pdf where these are known without much extra work.
8. The approximating function or surface should look "reasonable", i.e. there should be no numerical oscillations, but the method should give local extremal values also outside of the known function values.

The idea is to use the Bézier spline representation of the surface as the smooth approximation. The result of pointwise dividing the pdf by this approximation gives a more flat surface. This surface can then be interpolated by (bi)linear, or even cubic, splines. Eventually, the approximation of the pdf is obtained by multiplying with the Bézier surface.

3.7.1 Bézier splines

The basis of the Bézier splines is the same as for the B-splines. However, instead of computing spline coefficients, the original function values are used. That is, instead of equation (3.5), use $M = K$ and

$$\widetilde{f}(x) = \sum_{k=1}^{K} v_k B_{k,j}(x). \tag{3.10}$$

This does not give interpolation, unless inside intervals where three or more points form a straight line, so the resulting function is only approximating. It can be shown that the Bézier approximation is contained in the convex hull of the function values, so the error is systematical, in the sence that peaks or intervals where $f''(x) < 0$ normally will be underestimated and long tails or intervals with $f''(x) > 0$ usually is over-estimated. The Bézier curve is used in computer graphics and modelling because it is simple to evaluate and represent the shape of structures, functions and surfaces well. A survey of Bézier splines and a simple discussion of B-splines and other approximative splines can be found in [4]. Notice also that the second order Bézier spline coincides with the second order B-spline, and is hence an interpolating piecewise linear function.

3.7.2 The Bézier method procedure

To simplify the details of this method, and because this is efficient for computer programming, assume that the function f that is to be interpolated is known on a uniform infinite grid. As we will regard f as a density, assume $f(x) \geq 0$ for all x. To illustrate the procedure, we begin by a function of one variable. Without loss of generality, we assume that the function f is known for integer values, i.e. we have given $f(k)$ for $k = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$. Also, the discussion is based on a cubic Bézier spline, however the idea is the same for a rectangular spline or higher order splines.

The values of the approximating Bézier function $\widetilde{f}$ at the grid points are found using equation (3.6) and (3.10), which gives

$$\widetilde{f}(k) = \frac{1}{6}f(k-1) + \frac{2}{3}f(k) + \frac{1}{6}f(k+1). \tag{3.11}$$

A pointwise division gives then the pointwise function $\widetilde{q}(k)$,

$$\widetilde{q}(k) = \begin{cases} 0 & \text{if } f(k) = 0, \\ \frac{f(k)}{\widetilde{f}(k)} & \text{otherwise.} \end{cases} \tag{3.12}$$

This works, since $f(k) > 0$ and $f(x) \geq 0$ for all x by equation (3.11) implies $\widetilde{f}(k) > 0$. Let x be the point that is to be evaluated. That is, we want to approximate $f(x)$. Let $\lfloor k \rfloor$ be the largest integer smaller than or equal to x. A linear interpolation of $\widetilde{q}$ is

$$\widehat{q}(x) = (1 - x + \lfloor k \rfloor)\widetilde{q}(\lfloor k \rfloor) + (x - \lfloor k \rfloor)\widetilde{q}(\lfloor k \rfloor + 1). \tag{3.13}$$

Then we approximate $f(x)$ by

$$\widehat{f}(x) = \widetilde{f}(x)\widehat{q}(x). \tag{3.14}$$

Remember from equation (3.10) that $\widetilde{f}$ is described by smooth polynomials, so the Bézier-spline is defined for all x. Accordingly, from equation (3.13), $\widehat{q}(x)$ is defined for all x and continuous, so the result of equation (3.14) is a continuous function.

3.7.3 Properties of the Bézier based method

Firstly, direct substitution shows that $\widehat{f}(k) = f(k)$ for all k, so the spline is interpolating. Even though this is not a B-spline, it is in fact of order 5 (degree 4), it consists of piecewise polynomials, so using the word splines is justified.

If $f(k) \geq 0$ for all k, the method above gives $\widetilde{f}(x) \geq 0$ for all x, and thereby $\widetilde{q} \geq 0$, $\widehat{q} \geq 0$, and $\widehat{f} \geq 0$. This is a good property for probability densities.

$\widehat{f}(x)$ is bounded by $\frac{23}{20}\left[(1 - x + \lfloor k \rfloor)f(\lfloor k \rfloor) + (x - \lfloor k \rfloor)f(\lfloor k \rfloor + 1)\right]$ if $f \geq 0$ and the grid is uniform. In plain text this means that the deviation above a linear approximation is never larger than 15%. For the cubic B-spline, no such upper limit exist. It is not possible to give a lower limit of $\widehat{f}$ relative to a linear approximation, as the relative difference diverges to infinity as $x \to 1^+$ or $x \to 3^-$ in Figure 3.2 in the examples below in section 3.7.6. One of the other examples in that section, the right plot in Figure 3.2, is the case when this maximum is reached. This relative upper bound was found through the following observations:

- Linear interpolation depends only on the two nearest neighbouring grid points of x, $\widetilde{f}$ only depends on the four nearest neighbour points. So for $\widehat{f}$ only four nodal points (x_{-1}, v_{-1}), (x_0, v_0), (x_1, v_1), and (x_2, v_2) needs to be considered.

- Since both linear interpolation and $\widehat{f}$ are interpolants, there is no relative error at the grid points. So the maximal error with respect to the linear interpolant will be searched for inside the open interval (x_0, x_1).

- The relative error is scale independent in both x and v. Since the grid is uniform, calculations are simplified by choosing $x_{-1} = -1, x_0 = 0, x_1 = 1, and x_2 = 2$. This gives $\lfloor k \rfloor = 0$.

- If $v_0 = 0$ and $v_1 = 0$, $\widetilde{q}$ will be zero in the whole interval (x_0, x_1), and hence the linear interpolant and $\widehat{f}$ coincides. To obtain the maximum error relatively to the linear interpolant, at least one of the nearest nodal values must be non-zero. Since the problem is symmetric, this choice is arbitrary, so assume $v_0 > 0$.

- The independence of scale in v-direction can now be used to fix $v_0 = 1$.
- Write out the expression for the spline and the linear interpolant with repect to the nodal values v_{-1}, v_1, and v_2, $v_{-1}, v_1, v_2 \geq 0$ and evaluated at x, $0 < x < 1$. The linear interpolant is simply $q(x) = v_0(1-x) + v_1 x$.
- The ratio $\widehat{f}(x)/q(x)$ is maximized by computing the partial derivatives in x, v_{-1}, v_1, and v_2, setting the resulting expressions to zero, and solving the set of four equations and four unknowns.

The smoothness of $\widehat{f}$ is generally not any better than the smoothness of $\widehat{q}$. Using a linear interpolation for $\widehat{q}$, as done here, means that the derivative of $\widehat{f}$ does not exist for integral values for general functions f. However, this does not mean that the interpolation is coarse.

The left and right derivatives when $x \to m$ are

$$\widehat{f}'(x^-) = \widetilde{f}'(x^-)\widehat{q}(x^-) + \widetilde{f}(x^-)\left[\widetilde{q}(m) - \widetilde{q}(m-1)\right], \tag{3.15}$$

$$\widehat{f}'(x^+) = \widetilde{f}'(x^+)\widehat{q}(x^+) + \widetilde{f}(x^+)\left[\widetilde{q}(m+1) - \widetilde{q}(m)\right]. \tag{3.16}$$

Since $\widetilde{f}$ is differentiable everywhere and $\widehat{q}$ is continous, the left and right derivatives will in many cases be close to each other. For some special functions, they are equal, as will be seen from the examples below. The requirements for these functions could be found by setting the left and right derivatives equal to each other. E.g. it is always differentiable in points k where $f(k) = 0$.

In the two-dimensional case, still with an integer grid and cubic Bézier splines, the procedure would be the following, assuming that $\lfloor j \rfloor$ and $\lfloor k \rfloor$ are the largest integers such that $\lfloor j \rfloor \leq x$ and $\lfloor k \rfloor \leq y$, and the counters j and k are run over the whole (extended) domain $\mathbb{Z}$.

$$\begin{aligned}\widetilde{f}(j,k) = {} & \frac{1}{36}f(j-1,k-1) + \frac{1}{9}f(j-1,k) + \frac{1}{36}f(j-1,k+1) \\ & + \frac{1}{9}f(j,k-1) + \frac{4}{9}f(j,k) + \frac{1}{9}f(j,k+1) \\ & + \frac{1}{36}f(j+1,k-1) + \frac{1}{9}f(j+1,k) + \frac{1}{36}f(j+1,k+1)\end{aligned} \tag{3.17}$$

$$\widetilde{q}(j,k) = \begin{cases} 0 & \text{if } f(j,k) = 0, \\ \frac{f(j,k)}{\widetilde{f}(j,k)} & \text{otherwise.} \end{cases} \tag{3.18}$$

$$\begin{aligned}\widehat{q}(x,y) = {} & (1 - x + \lfloor j \rfloor)(1 - y + \lfloor k \rfloor)\widetilde{q}(\lfloor j \rfloor, \lfloor k \rfloor) + (1 - x + \lfloor j \rfloor)(y - \lfloor k \rfloor)\widetilde{q}(\lfloor j \rfloor, \lfloor k \rfloor + 1) \\ & + (x - \lfloor j \rfloor)(1 - y + \lfloor k \rfloor)\widetilde{q}(\lfloor j \rfloor + 1, \lfloor k \rfloor) + (x - \lfloor j \rfloor)(y - \lfloor k \rfloor)\widetilde{q}(\lfloor j \rfloor + 1, \lfloor k \rfloor + 1).\end{aligned} \tag{3.19}$$

$$\widehat{f}(x,y) = \widetilde{f}(x,y)\widehat{q}(x,y). \tag{3.20}$$

This extension of equations (3.11), (3.12), (3.13), and (3.14) is trivially extended to higher dimensions.

3.7.4 Number of arithmetic operations

Using this method on an n-dimensional system will require one division, $n+1$ multiplications with a constant, and 3^n additions per grid point to calculate $\widetilde{q}$. For each interpolated value, you need to calculate the value of the four non-zero splines for this point and add their contributions, do a linear interpolation (2 additions and 2 multiplications), and multiply these two results together.

For a cubic spline, all the points in the grid is traversed n times to compute the spline coefficients when utilizing the tensor product form. The n-dimensional spline coefficients are just the 1-dimensional spline coefficients in the last variable of the $n-1$-dimensional spline coefficients of the previous variables. The computations for each row or column could be done with $M-1$ multiplications with a constant, $2M-1$ variable multiplications, and $2M-2$ additions, assuming a uniform grid of size M. The interpolated value is easier to compute when you have the coefficients, as you can skip the linear interpolation and the last multiplication.

Assume a grid with M points in each direction where K values are to be found by interpolation. If you forget the computation of the B-spline basis functions, the total number of computations is as follows. The new method requires M^n divisions, $M^n(n+1)$ constant multiplications, $3K$ multiplications, and M^n+2K additions. The cubic B-spline requires (at least) $M^{n-1}n(M-1)$ constant multiplications, $M^{n-1}n(2M-1)$ multiplications, and $M^{n-1}n(2M-2)$ additions.

Remember that $\widetilde{f}$ is just used as an approximation of the shape of the function. When the density changes very little at each iteration, the approximation might not have to be updated so often. This means that the number of computations can be reduced significantly to just the $3K$ multiplications and $2K$ additions in addition to finding the value of the Bézier spline. There is no trick like this for the B-spline, where the spline coefficients must be calculated again each iteration.

Pre-calculating the basis spline reduces the cost to $4M^n$ multiplications for the B-spline, and $5M^n$.

3.7.5 Boundary values

Let the grid be uniform with integer values, but assume now that $f(k)$ is only known or defined for an interval, say from 0 to M. Scaling and translating the grid does not affect the behaviour of the interpolation, so this is still an assumption without loss of generality for any uniform grid.

When extending f to $\mathbb{Z}$, four basis splines that will contribute to the value of $\widetilde{f}$ in the interval are then unknown, namely those scaled with $f(-2)$, $f(-1)$, $f(M+1)$, and $f(M+2)$. All other points outside the interval can be assumed to be zero (or any other value) without affecting the computations, since the cubic Bézier spline is local within the four nearest neighbouring nodal points.

If it is known that f should go smoothly to zero on the upper boundary, it is reasonable to set $f(M+1) = f(M+2) = 0$, and accordingly for the lower boundary if applicable.

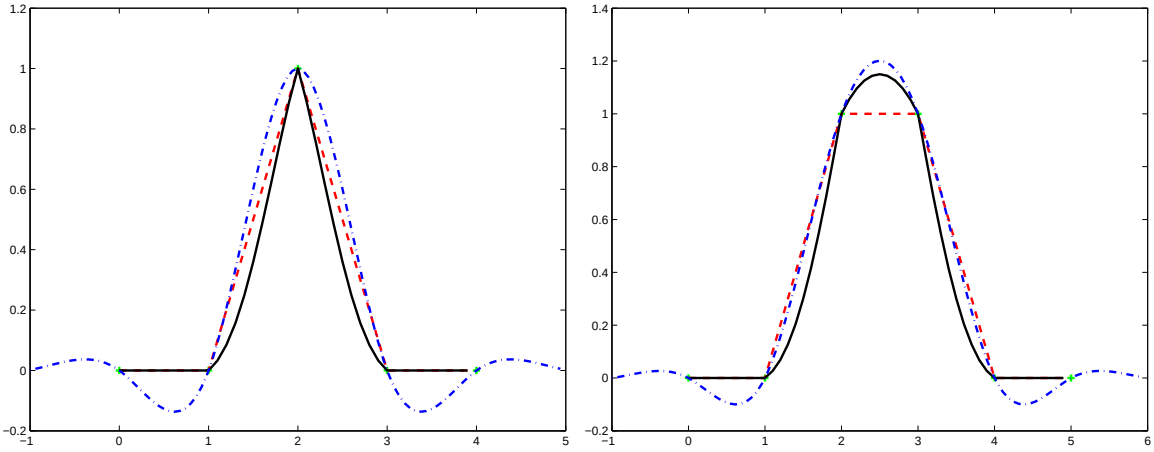


Figure 3.2: Left: Interpolation of "delta function", i.e. just one non-zero nodal value. Right: Similar result for the case of two non-zero nodal values. Red: Linear approximation, Blue: Cubic spline, Black: New method.

In other cases, these four values should be chosen so that the shape of $\widetilde{f}$ is similar to the shape of f. The interpolation will be better if $\widetilde{q}$ is mostly constant. Setting the values of the extended nodal points to zero at a boundary where the density does not go smoothly to zero could result in unwanted oscillations in the two intervals closest to this boundary.

3.7.6 Examples

The numerical delta-function and unit-function

Take the extreme case first. If the function is zero everywhere except in one point, what is the most reasonable interpolation? Well, it is difficult to say. In practice, a numerical method should maybe not continue if a situation like this occurs, unless this is used as a starting distribution. However, it will be good as an illustration, shown in the left of Figure 3.2, as the cubic spline interpolant clearly shows the unwanted oscillating behaviour and how it could be negative in such cases, as discussed above. Note also that the new interpolant is sharper than the linear on the spike, and clearly not differentiable.

Let us add more non-zero nodal values, gradually approaching a discretization of a unit step function. The result of two nodal values is shown in the right side of Figure 3.2, three and four non-zero nodal values in the left and right of Figure 3.3, respectively. Of course, now the linear interpolant is optimal where the density is flat. However, the new interpolant is quite good, given that it really is a high-order polynomial. Note that it gives a straight line in the middle when four or more points are aligned, and that it climbs up the step function with a steep gradient.

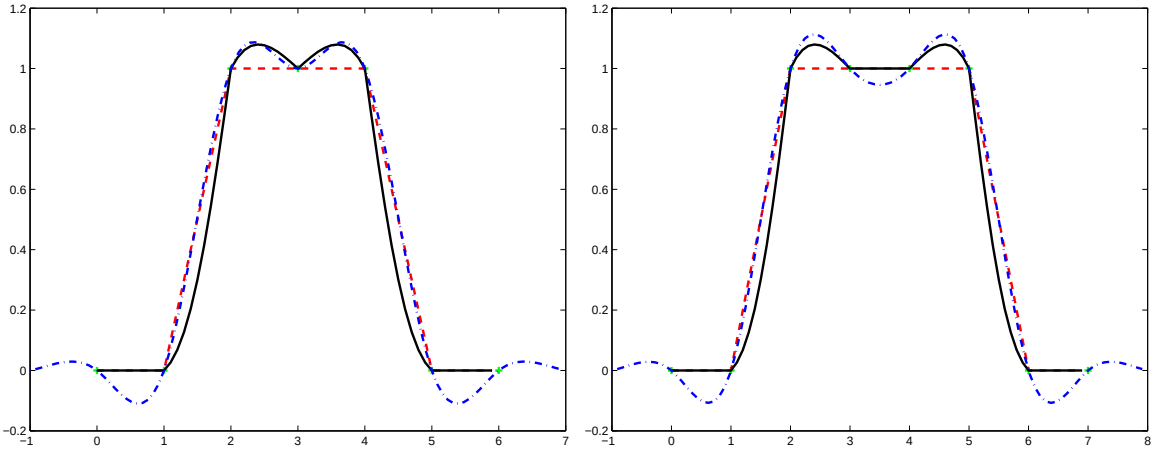


Figure 3.3: Three and four non-zero nodal values in the left and right plot, respectively. Red: Linear approximation, Blue: Cubic spline, Black: New method.

The linear function

One can easily check that four nodal points aligned on a straight line gives a linear interpolation in the middle knot interval. Start with $f(m) = \alpha m + \beta$, which gives

$$\widetilde{q}(m) = \frac{\alpha m + \beta}{\frac{1}{6}(\alpha(m-1)+\beta) + \frac{2}{3}(\alpha m + \beta) + \frac{1}{6}(\alpha(m+1)+\beta)} = 1. \tag{3.21}$$

In fact, not only is the denominator $\widetilde{f}(n)$ a straight line, $\widetilde{f}(x)$ is a straight line for all values of x when there are at least two nodal points on each side that also are on this line. Hence is $\widehat{f}$ a perfect interpolation in such intervals. The cubic spline will not adapt unless the line has infinite length. An illstration is shown in Figure 3.4.

Polynomials

One might expect that this new interpolation is perfect for any polynomial up to degree 4. This is however not the case. But the interpolation is good anyway for most cases. Remember that we want to interpolate probability densities, and there is usually no reason to believe that they behave locally like polynomials. An example, a second degree polynomial, is shown in Figure 3.5, including a more detailed plot of the error.

The exponential function

The exponential function without a constant term is not perfectly interpolated. This is an interesting case anyway, because the interpolant $\widehat{f}(x)$ is differentiable everywhere. This is seen by setting $f(m) = \alpha e^{\beta m}$ and calculating

$$\widetilde{q}(m) = \frac{\alpha e^{\beta m}}{\frac{1}{6}\alpha e^{\beta(m-1)} + \frac{2}{3}\alpha e^{\beta m} + \frac{1}{6}\alpha e^{\beta(m+1)}} = \frac{1}{\frac{2}{3} + \frac{1}{6}(e^{\beta} + e^{-\beta})}. \tag{3.22}$$

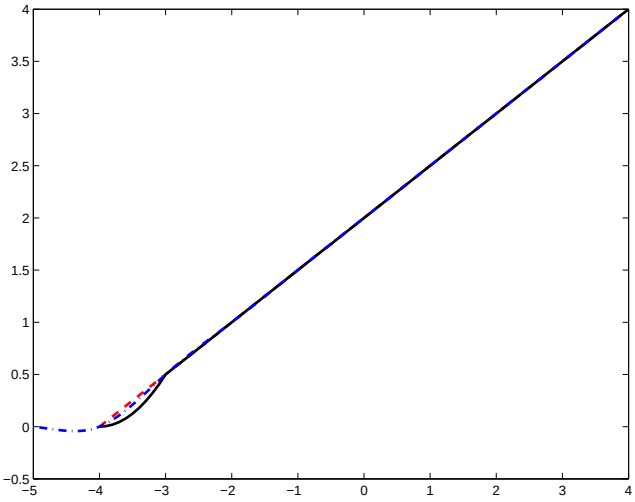


Figure 3.4: The interpolants of a linear function. Red: Linear approximation, Blue: Cubic spline, Black: New method.

An example is shown in Figure 3.6. When $\widetilde{q}$ is a constant, the linear interpolation is perfect and differentiable, so the only approximation is $\widetilde{f}$. In a logarithmic plot, $\widehat{f}$ is very close to a straight line. [2]

The Gaussian distribution

The new interpolation method underestimates the normal distribution slightly, both close to the top and in the tails. Here, the B-spline performs better. However, the new method does not consistently underestimate peaks and overestimate tails, which is the main problem with the linear interpolant. In fact, close to the boundary, where you have to choose a boundary condition for the splines, the new method performs slightly better, as both the absolute and the relative interpolation error decreases to the tails. This is not the case for the B-spline or the linear approximation. For the example plots in Figure 3.7 it is assumed that all nodal values outside of the interval are zero.

[2]This also holds for imaginary β, hence the cosine and sine functions are smoothly interpolated even if the grid spacing is not a quotient of the period. This has not been studied further, since this project is restricted to interpolation of probability densities where f is always non-negative.

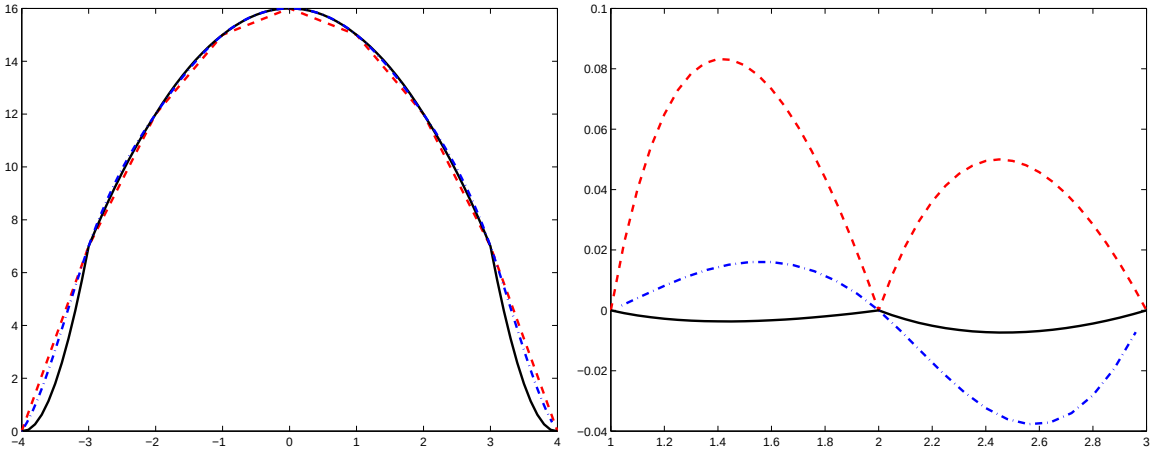


Figure 3.5: Left: The quadratic function $16 - x^2$ and the same interpolants as in the previous figures. Right: The same function, but focusing on the error by computing $\sqrt{16 - \widehat{f}(x)} - x$.

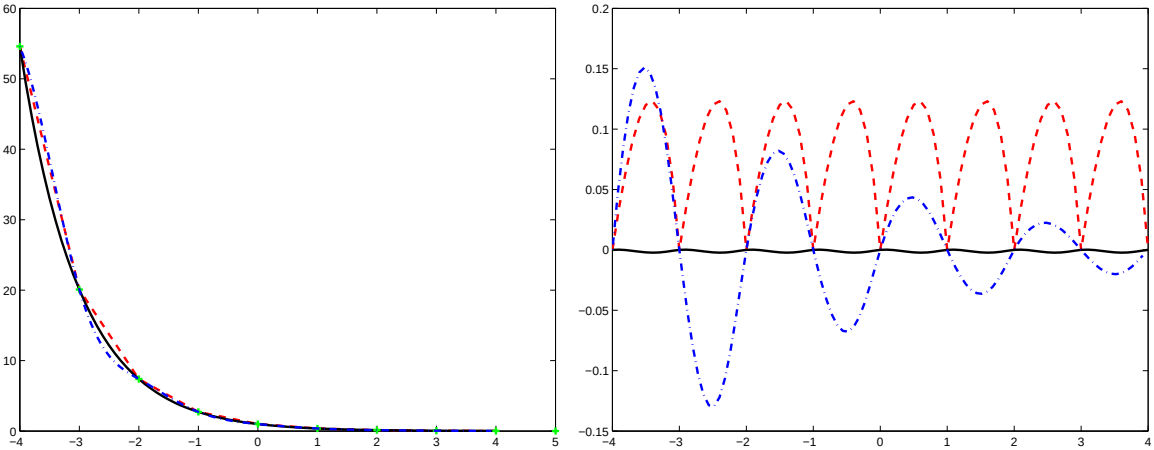


Figure 3.6: Left: the interpolants of an exponential function $y = \exp(x)$. Right: the error for the same interpolation, obtained by computing $\log(\widehat{f}(x)) - x$.

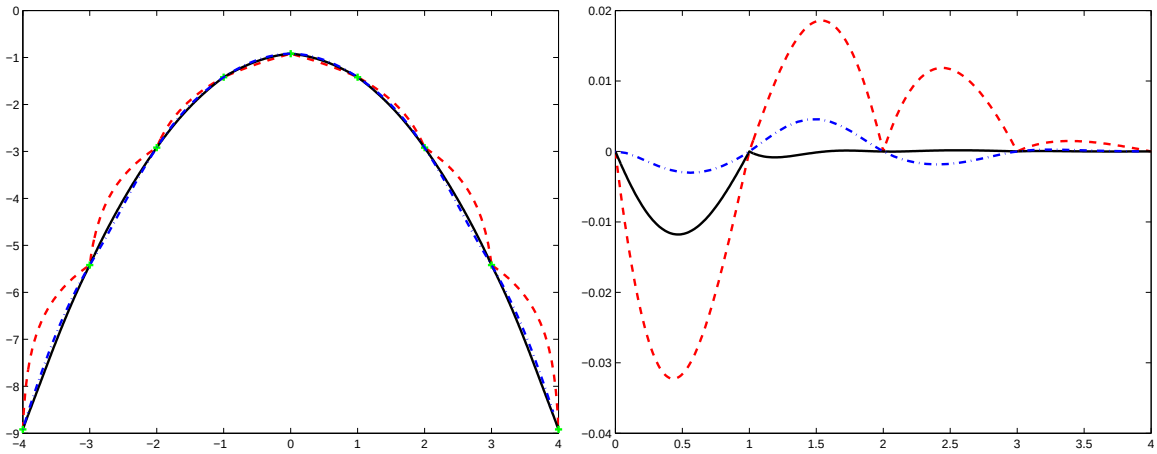


Figure 3.7: Left: A Gaussion function with standard deviaton 1 and mean value 0 is interpolated, and $\log(\widehat{f}(x))$ is plotted. Right: The same interpolation, where $\widehat{f}(x) - f(x)$ is presented.

Chapter 4

Path integration

4.1 A Markov integral operator

In this section it will be shown how an SDE, equation (1.12), in a natural way leads to a Markov integral operator.

The first important choice is to restrict the analysis and numerical implementation to solve a time discretized version of equation (1.12), like one of equations (2.4), (2.7), (2.8), or (2.11). For simplicity, the discussion is based on equation (2.4), however the idea is the same for all the discretization methods, and therefore $a(x)\Delta t$ is replaced by $\widetilde{a}(x)\Delta t$. It follows from the discretized equations and the fact that the Wiener process has independent increments that the sequence $\{\mathbf{X}_{n\Delta t}\}_{n=0}^{\infty}$ is a Markov chain. It is also observed from equation (2.4) that the associated transition probability density function (TPD), which is denoted by $f_{tt'}(\mathbf{x}|\mathbf{x}') = f(\mathbf{x}, t|\mathbf{x}', t')$, is a Gaussian probability density function (PDF) since $\Delta W_{t'}$ is a Gaussian variable for every t'.

Before proceeding, it is expedient to introduce one simplification implied by the dynamic systems considered here, which amounts to setting $r = 1$ and $b_1(\mathbf{x}, t) = \gamma \mathbf{e}_n$. So the noise process is additive, meaning that the variance over a small time step is independent of the state space variables. In addition, the stochastic excitations only enter through the last component of the SDE. The latter is not a basic limitation of the theory, but serves to simplify the discussion. A generalization will be discussed by the end of this section.

From the assumptions above we may now write down the TPD corresponding to equation (2.4)

$$f_{tt'}(\mathbf{x}|\mathbf{x}') = \prod_{i=1}^{n-1} \delta(x_i - x_i' - \widetilde{a}_i(\mathbf{x}', t')\Delta t) \cdot f_{tt'}(x_n|\mathbf{x}') \tag{4.1}$$

where $\mathbf{x} = (x_1, \ldots, x_n)^T$, $\delta(\cdot)$ denotes the Dirac delta function symbol and

$$f_{tt'}(x_n|\mathbf{x}') = \frac{1}{\sqrt{2\pi} s(\mathbf{x}')} \cdot \exp\left\{ -\frac{(x_n - x_n' - \widetilde{a}_n(\mathbf{x}', t')\Delta t)^2}{2s(\mathbf{x}')^2} \right\} \tag{4.2}$$

where $s(\mathbf{x})^2 = \sum_{j=1}^m b_j(\mathbf{x})^2 \Delta t$. Hence, $f_{tt'}(\mathbf{x}|\mathbf{x}')$ can be interpreted as a degenerate multi-dimensional Gaussian PDF. We shall assume throughout that $s(\mathbf{x})^2 \geq K\Delta t$, where K is a positive constant.

Starting with an initial density of states $f_{\text{init}}(\mathbf{x})$ the study of the evolution in time of this density is based on the following formal expression of the total probability law

$$f_t(\mathbf{x}) = \int_{\mathbb{R}^n} f_{tt'}(\mathbf{x}|\mathbf{x}') f_{t'}(\mathbf{x}')\, \mathrm{d}\mathbf{x}' \tag{4.3}$$

where the (joint) PDF of $\mathbf{X}_t$ at time t is denoted by $f_t(\mathbf{x}; t)$, and $f_0(\mathbf{x}) = f_{\text{init}}(\mathbf{x})$. To ensure the validity of the integration range, let f be zero at any points $\mathbf{x}'$ where the formal expression for $f(\mathbf{x}'; 0)$ is undefined. Equation (4.3) now has to be interpreted in the light of equation (4.1). To obtain the appropriate result, the inverse and the implicit function theorems from section 1.1 will be used. Let $\tilde{\mathbf{x}} = (x_1, \ldots, x_{n-1})^T$, whenever $\mathbf{x} = (x_1, \ldots, x_n)^T$ is an n-dimensional vector or vector function. Consider the two equations $\mathbf{x} = h(\mathbf{x}')$ and $\tilde{\mathbf{x}} = \tilde{h}(\mathbf{x}') = \tilde{h}(\tilde{\mathbf{x}}', x_n')$, where $h(\mathbf{x}') = \mathbf{x}' + \widetilde{a}(\mathbf{x}')\Delta t$. Let

$$\begin{aligned} J_h(\mathbf{x}') &= \det(D_{x'} h(\mathbf{x}')) = \det\left(\frac{\partial(h_1(\mathbf{x}'), \ldots, h_n(\mathbf{x}'))}{\partial(x_1', \ldots, x_n')}\right), \\ J_{\tilde{h}}(\tilde{\mathbf{x}}', x_n) &= \det(D_{\tilde{x}'} \tilde{h}(\tilde{\mathbf{x}}', x_n)) = \det\left(\frac{\partial(h_1(\mathbf{x}'), \ldots, h_{n-1}(\mathbf{x}'))}{\partial(x_1', \ldots, x_{n-1}')}\right). \end{aligned} \tag{4.4}$$

It follows immediately that $J_h(\mathbf{x}') \neq 0$ and $J_{\tilde{h}}(\mathbf{x}')) \neq 0$ for every $\mathbf{x}'$ for sufficiently small Δt. This means that for every point $\mathbf{x}$ there is an open neighbourhood U containing $\mathbf{x}$ where a smooth inverse $g = h^{-1} : h(U) \to U$ exists. Also, for a fixed point $\mathbf{x}$, let $\boldsymbol{\xi}' = (\xi_1', \ldots, \xi_n')^T = g(\mathbf{x})$. Then there is an open interval U_n containing ξ_n and a function $\tilde{g} : U_n \to \mathbb{R}^{n-1}$ so that the equation $\tilde{\mathbf{x}} = \tilde{h}(\tilde{g}(x_n'), x_n')$ is satisfied for every $x_n' \in U_n$, and $J_{\tilde{h}}(\tilde{g}(x_n')) \neq 0$ for $x_n' \in U_n$. Write $\tilde{g}(x_n') = \tilde{g}(\tilde{\mathbf{x}}, x_n')$ to make explicit the dependence on $\tilde{\mathbf{x}}$. It is clear that the function $\tilde{g}$ can be suitably modified outside U_n so that U_n can be extended to $\mathbb{R}$.

The introduction of this auxiliary system will simplify the discussion without imposing an important restriction. The reason for this is that in the limit when $\Delta t \to 0$, $U_n \to \mathbb{R}$. Assuming that $U_n = \mathbb{R}$, the following relation expressing the combination of equations (4.1) and (4.3) is then obtained

$$\begin{aligned} f_t(\mathbf{x}) &= \int_{\mathbb{R}} f_{tt'}(x_n \,|\, \tilde{g}(\tilde{\mathbf{x}}, x_n'), x_n')\, f_{t'}(\tilde{g}(\tilde{\mathbf{x}}, x_n'), x_n')\, |J_{\tilde{h}}(\tilde{\mathbf{x}}, x_n')|^{-1}\, \mathrm{d}x_n' \\ &= \int_{\mathbb{R}} H(x_n', \mathbf{x}; t) f_{t'},\, \mathrm{d}x_n'. \end{aligned} \tag{4.5}$$

Let us show that equation (4.5) defines an operator $P_{tt'} : \mathcal{D} \to \mathcal{D}$. $\mathcal{D} = \{f : \int_{\mathbb{R}^n} f\, \mathrm{d}\mathbf{x} = 1\}$ is the space of probability densities over the state space. $f(\,\cdot\,; t)$ is well-defined for every

$f(\,\cdot\,;t') \in \mathcal{D}$. To this end, let $f(\,\cdot\,;t') \in \mathcal{D}$. Then

$$\begin{aligned}
&\int_{\mathbb{R}^n} f(\mathbf{x};t)\,\mathrm{d}\mathbf{x} \\
&\quad= \int_{\mathbb{R}^{n+1}} f_{tt'}(x_n \,|\, \tilde{g}(\tilde{\mathbf{x}}, x_n'), x_n')\, f(\tilde{g}(\tilde{\mathbf{x}}, x_n'), x_n'; t')\, |J_{\tilde{h}}(\tilde{\mathbf{x}}, x_n')|^{-1}\,\mathrm{d}x_n\,\mathrm{d}\tilde{\mathbf{x}}\,\mathrm{d}x_n' \\
&\quad= \int_{\mathbb{R}^n} f(\tilde{g}(\tilde{\mathbf{x}}, x_n'), x_n'; t')\, |J_{\tilde{h}}(\tilde{\mathbf{x}}, x_n')|^{-1}\,\mathrm{d}\tilde{\mathbf{x}}\,\mathrm{d}x_n'. \qquad (4.6)
\end{aligned}$$

Define $J_{\tilde{g}}(\tilde{\mathbf{x}}', x_n') = \det(D_{\tilde{x}}\tilde{g}[\tilde{\mathbf{x}}, x_n'])$ similarly to the Jacobians in equation (4.4). From the equation $\tilde{\mathbf{x}} = \tilde{h}(\tilde{g}(\tilde{\mathbf{x}}, x_n'), x_n')$, it is obtained that $I = D_{\tilde{x}}\tilde{\mathbf{x}} = D_{\tilde{x}'}\tilde{h}[\tilde{g}(\tilde{\mathbf{x}}, x_n'), x_n']\, D_{\tilde{x}}\tilde{g}(\tilde{\mathbf{x}}, x_n')$, where I denotes the $n \times n$ identity matrix. That is, $1 = J_{\tilde{h}}(\tilde{\mathbf{x}}, x_n') J_{\tilde{g}}(\tilde{\mathbf{x}}, x_n')$. Introducing the change of variables $(\tilde{\mathbf{x}}, x_n') \to \mathbf{x}' = (\tilde{g}(\tilde{\mathbf{x}}, x_n'), x_n')$ into equation (4.6), it follows that

$$\begin{aligned}
&\int_{\mathbb{R}^n} f(\mathbf{x};t)\,\mathrm{d}\mathbf{x} = \\
&\int_{\mathbb{R}^n} f(\mathbf{x}';t')\, |J_{\tilde{h}}(\tilde{\mathbf{x}}, x_n')|^{-1}\, |J_{\tilde{g}}(\tilde{\mathbf{x}}, x_n')|^{-1}\,\mathrm{d}\mathbf{x}' = \\
&\int_{\mathbb{R}^n} f(\mathbf{x}';t')\,\mathrm{d}\mathbf{x}' = 1. \qquad (4.7)
\end{aligned}$$

Hence, $f(\,\cdot\,;t) \in \mathcal{D}$, which proves that $P_{tt'} : \mathcal{D} \to \mathcal{D}$.

With some restrictions on the dynamics and a non-zero variance everywhere, the operator P also defines a map between densities in a smaller space, $\mathcal{D}_0^C = \mathcal{D} \cap \mathcal{C}_0$, where $\mathcal{C}_0$ is the space of bounded functions that go to zero at infinity. That is,

$$P_{tt'} : \mathcal{D}_0^C \to \mathcal{D}_0^C,\ \mathcal{D}_0^C = \{f \in \mathcal{D} : f(\mathbf{x}) < \infty\, \forall\, \mathbf{x}, f(\mathbf{x}) \to 0\, \forall\, ||\mathbf{x}|| \to \infty\}.$$

This investigation will be performed in two steps, first checking that the new density goes to zero at infinity, and second to ensure that it is bounded everywhere.

For every choice of ϵ and K, one can choose a compact set $V' \subset \mathbb{R}^n$ such that $\int_{V'} f(\mathbf{x}')\,\mathrm{d}\mathbf{x}' > 1 - \epsilon$ and $f(\mathbf{x}') < K \forall x \notin V'$. For a given Δt, $V = h(V')$ is compact. The mapping $h : V' \to V$ is a continuous homeomorfism for a smooth dynamical system, i.e. when there are no shocks or limit transitions present. This also defines a variable transformation also in the last variable,

$$\xi = x_n' + \widetilde{a}_n(\mathbf{x}', t') = h_n(\mathbf{x}', t'),$$

which then has a continuous Jacobian, and the Jacobian is bounded for $\mathbf{x}' \in V'$. From here on, the function argument $\tilde{\mathbf{x}}, \xi$ will be used indicating that the last element for the point $\mathbf{x}$ is substituted by the variable ξ, which now is the integration variable over the last state space dimension.

Now regard the image of $f(\mathbf{x}')$ in V after this transformation and denote it $f^\dagger(\mathbf{x}) = f(h^{-1}(\mathbf{x}))$. This $f^\dagger$ is then a bounded continuous function in $\mathcal{C}_0$. (Note, $f^\dagger$ is not a

probability density.) The integral in equation (4.5) is then along a straight line through $\mathbf{x}$ parallell to the x_n axis. Using that $g = h^{-1}$ and $|J_h(\tilde{\mathbf{x}}, \xi)|^{-1} = |J_g(\tilde{\mathbf{x}}, \xi)|$, the formula reads

$$f(\mathbf{x}, t) = \int_{\mathbb{R}} f_{tt'}(x_n | g(\tilde{\mathbf{x}}, \xi)) f^\dagger(\tilde{\mathbf{x}}, \xi) |J_g(\tilde{\mathbf{x}}, \xi)| \, \mathrm{d}\xi. \tag{4.8}$$

Consider a sequence of points $\mathbf{x}$ with increasing distance from V. Because $p_{tt'}$ is centered in $\mathbf{x}$, the contribution to the integral (4.3) of $f(\mathbf{x}')$, or the image $f^\dagger$ is reduced, whenever $f_{tt'}$ is bounded and goes to zero for $||\xi|| \to \pm\infty$. This is fulfilled for the criterion for the standard deviation already stated, namely that $(s(\mathbf{x}'))^2 > c\Delta t > 0$ for every $\mathbf{x}'$. In that case, also $f(\mathbf{x})$ must go to zero as $||\mathbf{x}|| \to \infty$.

Inside this integral, $f \to 0$ for $|\xi| \to \infty$, and $f_{tt'} \to 0$, for additive noise like the normal distribution. So the integral will converge everywhere unless the Jacobian diverges to infinity too fast. In other words, the variable transformation provides a criterion that the density $f(\mathbf{x})$ is surely bounded at $\mathbf{x}$. Using the limit comparison test with $1/\xi$ one obtains

$$\lim_{\xi \to \pm\infty} \frac{\xi}{|s(g(\tilde{\mathbf{x}}, \xi))|} \exp\left\{-\xi^2 / [2s(g(\tilde{\mathbf{x}}, \xi))^2]\right\} |J_g(\tilde{\mathbf{x}}, \xi)| = 0. \tag{4.9}$$

Otherwise, the boundedness will depend on how fast $f(\mathbf{x}', t')$ decreases. The criteria that has been set for the variance function s^2 and the dynamics, here g, are sufficient for the operator to map $\mathcal{D}_0^C$ to itself. These are hence not strict limitations for when PI is applicable.

For the numerical solution of the problem of establishing the PDF $f_{n\Delta t}(\mathbf{x})$ of the state space vector $\mathbf{X}_{n\Delta t}$ given that the PDF of the initial state $\mathbf{Y}_0$ is equal to $f_{\text{init}}(\mathbf{y}) \in \mathcal{D}$, it is necessary to study the compound effect of the Markov operators $P_i = P_{i\Delta t,(i-1)\Delta t}$, $i = 1, 2, \ldots$. Defining $P_{(i)} = P_i \circ \ldots \circ P_1$, where $\circ$ denotes operator composition, then clearly $f_n(\mathbf{x}) = P_{(n)} f_0(\mathbf{x})$. If there exists a stationary PDF, $f^*(\mathbf{x})$ say, then this can be estimated by calculating $f_n(\mathbf{x})$ for sufficiently large values of n.

The existence of a stationary density for this kind of operator is determined not only by how the kernel depends on the time parameters, but also on how the stationarity concept itself is interpreted, as will be discussed below. Also note that a sequence of densities in $\mathcal{D}_0^C$ could converge to a density which is not in $\mathcal{C}_0$.

The choice of discretization scheme will of course affect the shape of the functions s and h, and hence g, but the general discussion above is still valid. An important point is to ensure that the integral remains one-dimensional, so terms that gives dispersion in other variables than the last should be avoided. Finally, the integral will typically also be discretized, using $\xi \in \{\xi_1, \xi_2, \ldots \xi_M\}$ where the number of integration points M and the upper and lower integration value ξ_1 and ξ_M are chosen to give a good numerical estimate. A trivial integration scheme is then

$$f(\mathbf{x}, t) = \sum_{j=1}^{M-1} f_{tt'}(x_n | g(\tilde{\mathbf{x}}, \xi_j)) f^\dagger(\tilde{\mathbf{x}}, \xi_j) |J_g(\tilde{\mathbf{x}}, \xi_j)| (\xi_{j+1} - \xi_j). \tag{4.10}$$

The integral can of course also be performed using a more advanced scheme like Simpson's method or the trapezoidal scheme. Note that this integration is different from the time stepping method, where one also can apply such a scheme, as seen in section 2.1.6.

The result above can directly be generalized to muliple additive noise processes, $r > 1$, if one assumes $b_j(\mathbf{x}, t) = \gamma_j \mathbf{e}_{n-r+j}$ for $j = 1, \ldots, r$. That is, each independent noise processes enters the SDE through only one state space variable, and $b_j(\mathbf{x}, t)$ is orthogonal to $b_k(\mathbf{x}, t)$ for all $j \neq k$, $1 \leq j, k \leq r$. For additive noise, this is not really a restriction, since this always can be obtained by linear combinations of the rows in the SDE and renaming and renumbering of variables. In that case, the domain denoted U_n above, will be an r-dimensional subset of $\mathbb{R}^n$, and will be extended by a proper modification of the backwards time step to $\mathbb{R}^r$. The integral after the variable transformation will now be a full integral over $\mathbb{R}^r$ following the coordinate axis.

4.2 Numerical approximation of volume change

To implement the Markov operator numerically, some additional obervations have been made. First, U_n is extended to $\mathbb{R}$ by approximating $g(\mathbf{y})$ by $\widehat{g}(\mathbf{y}) = \mathbf{y} - \widetilde{a}^-(\mathbf{y})\Delta t$; where $\widetilde{a}^-$ is a Runge-Kutta step backwards in time starting in $\mathbf{y}$. This adjustment also ensures the extension of the domain O_n to $\mathbb{R}$ in the above discussion for $\tilde{g}$. The accuracy of this approximation is as good as the initial time discretization of the SDE.

The basis for the numerical algorithm is equation (4.8). Since g and h are not expressed directly, but rather as a Runge-Kutta procedure through the system equations $a(\mathbf{x})$, it is efficient to approximate the Jacobian by the Jacobian matrix $Da(\mathbf{x})$ of the deterministic part of the system equations. To manage this, a new variable $\mathbf{u}$ is introduced, as a velocity vector

$$\begin{aligned} \mathbf{u} &= \dot{\mathbf{x}} = a(\mathbf{x}), \\ \dot{\mathbf{u}} &= \sum_{j=1}^{n} \frac{\partial a}{\partial x_j} \dot{x}_j = Da(\mathbf{x})\mathbf{u}. \end{aligned}$$

This differential equation is integrated from $\mathbf{u}(t)$ to $\mathbf{u}(t')$, assuming that $Da(\mathbf{x})$ is constant in t over the sufficiently small time step. One then obtains

$$\mathbf{u}' = \mathbf{u}(t') = e^{-Da(\mathbf{x})\Delta t}\mathbf{u}(t). \tag{4.11}$$

The velocity vector can also be defined from the function g;

$$\begin{aligned} \mathbf{x}' &= g(\mathbf{x}), \\ \mathbf{u}' &= \dot{\mathbf{x}}' = Dg(\mathbf{x})\dot{\mathbf{x}} = Dg(\mathbf{x})\mathbf{u}. \end{aligned} \tag{4.12}$$

Comparing equations (4.11) and (4.12) shows that the correction factor $|J_g|$ can be approximated by $\left|e^{-Da\Delta t}\right|$. This can again be evaluated as $\exp(-\Delta t \sum_{i=1}^{n} \frac{\partial a_i}{\partial x_i})$. This is proven

directly using the properties of the exponential function and the formulas for determinant and trace of a square matrix M in terms of the eigenvalues $\lambda_j(M)$, $j = 1, \ldots, m$;

$$|M| = \prod_{j=0}^{n} \lambda_j(M), \qquad \lambda_j(e^M) = e^{\lambda_j(M)},$$

$$\prod_{j=1}^{n} e^{\lambda_j} = e^{\sum_{j=1}^{n} \lambda_j}, \qquad \operatorname{tr}(M) = \sum_{j=1}^{n} \lambda_j(M).$$

$$|e^M| = \prod_{j=0}^{n} \lambda_j(e^M) = \prod_{j=0}^{n} e^{\lambda_j(M)} = e^{\sum_{j=1}^{n} \lambda_j(M)} = e^{\operatorname{tr}(M)}.$$

4.3 Example: The Van der Pol equation

To illustrate the method described above, consider the differential equation

$$\ddot{X}_t + 2\zeta[X_t^2 - 1]\dot{X}_t + X_t = \gamma N_t. \tag{4.13}$$

This is a simple van der Pol equation. Such equations are often used to model electrical and biological oscillations. The typical response of such a system is a limit cycle, similar to the one obtained by a linear filter, except that the time spent in some parts of the cycle is much higher than other parts, giving two distinct peaks. In addition, the limit cycle only depends on the parameters of the ODE, not the initial value.

First, consider this system in the state space $(X_t, \dot{X}_t)$. With $\gamma = 0$, this system is purely deterministic. It has an unstable equilibrium at the origin, and a non-trivial stable ring-shaped attractor around the origin. The flow from every initial point except the origin will spiral clockwise towards the stable attractor as $t \to \infty$. If time is reversed, the origin is a stable point for all initial points inside the ring, and every point outside will spiral counter-clockwise off to infinity.

In state space variables, with dimension as subscript, equation (4.13) is written

$$\begin{aligned} \mathrm{d}X_1 &= X_2 \,\mathrm{d}t, \\ \mathrm{d}X_2 &= (-2\zeta[X_1^2 - 1]X_2 - X_1)\,\mathrm{d}t + \gamma \,\mathrm{d}W. \end{aligned} \tag{4.14}$$

To simplify the discussion, assume that the Euler-Marayama form is used for the time steps. The basic ideas will be the same for the Runge Kutta method. The deterministic part of equation (4.14) is then

$$\begin{aligned} x_1 &= h_1(x_1', x_2') = x_1' + x_2'\Delta t, \\ x_2 &= h_2(x_1', x_2') = x_2' - (2\zeta[(x_1')^2 - 1]x_2' + x_1')\Delta t. \end{aligned} \tag{4.15}$$

As the noise is accumulated to the end of the time step, this system should be considered for fixed values of x_1. Combined, it gives a cubic polynomial in x_2' where $x_2 \to \mp\infty$ and

$x_1' \to \pm\infty$ when $x_2' \to \pm\infty$. However, since the value of Δt is small it is also clear that x_2 should be close to x_2' for some values of x_2', and be an increasing function of x_2' in some interval. This means that the cubic equation has three solutions, at least for some points (x_1', x_2'), and h is in principle not invertible. However, the flow of the deterministic part of the SDE from t' to t is always a homeomorfism. (Time can be reversed in the deterministic system, the spirals described above do not intersect, and every point maps to just one other point in the state space over Δt.)

So instead of trying to actually invert h, consider the Euler approximation with a negative time step;

$$\begin{aligned} x_1^\dagger &= g_1(x_1, x_2) = x_1 - x_2\Delta t, \\ x_2^\dagger &= g_2(x_1, x_2) = x_2 + (2\zeta[(x_1)^2 - 1]x_2 + x_1)\Delta t. \end{aligned} \tag{4.16}$$

Now, it is straightforward to compute $h(g(\mathbf{x}))$;

$$\begin{aligned} h_1(x_1^\dagger, x_2^\dagger) &= x1 + (2\zeta[x_1^2 - 1]x_2 + x_1)\Delta t^2, \\ h_2(x_1^\dagger, x_2^\dagger) &= x2 - \{(2\zeta x_2[x_2\Delta t - 2x_1] - 1)x_2 \\ &\quad - 2\zeta([x_1 - x_2\Delta t]^2 - 1)(2\zeta[x_1^2 - 1]x_2 + x_1)\}\Delta t^2. \end{aligned}$$

So $h(g(\mathbf{x})) = \mathbf{x} + \mathcal{O}(\Delta t^2)$; the error is on the same order as the error of the step method.

The Jacobian for the function g is $J_g = 1 + 2\zeta(x_1^2 - 1)\Delta t + (4\zeta x_1 x_2 + 1)\Delta t^2$, while the numerical approximation is $\det(\exp(-Da(x)\Delta t)) = \exp(2\zeta(x^2 - 1)\Delta t) = 1 + 2\zeta(x_1^2 - 1)\Delta t + 2\zeta^2(x_1^2 - 1)^2\Delta t^2 + \mathcal{O}(\Delta t^3)$, again correct to the order of our time step error.

In fact, the numerical approximation of $|Dg|$ is also used for the higher order time stepping methods. Using a second order Runge-Kutta method for g,

$$J_g = 1 - \Delta t\left(\frac{\partial a_1}{\partial x_1} + \frac{\partial a_2}{\partial x_2}\right) + \frac{\Delta t^2}{2}\left(\frac{\partial a_1}{\partial x_1} + \frac{\partial a_2}{\partial x_2}\right)^2 + \mathcal{O}(\Delta t^3), \tag{4.17}$$

the same series as the Taylor form of $\det(\exp(-Da(x)\Delta t))$.

4.4 Volume change at discontinuities

First, consider for constants $\alpha_1 \neq \alpha_2$, $\alpha_1, \alpha_2 > 0$ and some gluing point ξ the discontinuous equation

$$\mathrm{d}x_t = \begin{cases} -\alpha_1\,\mathrm{d}t, & \text{for } x_t \leq \xi, \\ -\alpha_2\,\mathrm{d}t, & \text{for } x_t > \xi. \end{cases} \tag{4.18}$$

This systems shifts every point, and thereby every interval, down to smaller values along the x-axis. As long as the shift does not move any point past ξ, the flow is given by the equation $x_{t+\Delta t} = x_t - \alpha_j\Delta t$, where $j = 1$ if $x_t \leq \xi$ and $j = 2$ for $x_t > \xi$. Considering two points defining an interval of length Δx, it is clear that the interval's length is preserved when shifted by a constant $-\alpha_1\Delta t$ or $-\alpha_2\Delta t$. Now consider the interval $[\xi, \xi + \Delta x]$, and

shift this down by a Δt chosen such that the upper limit of the interval ends up at $x = \xi$. The constant shift of this upper limit gives $\Delta x = \alpha_2 \Delta t$. The lower point of the interval is in the same time shifted to $\xi - \alpha_1 \Delta t$, so the size of the interval is now $\alpha_1 \Delta t = \Delta x \alpha_1 / \alpha_2$, i.e. the interval is scaled by a factor α_1/α_2. This argument is independent of Δx. Any sub-interval will first be shifted down to ξ without change of length, then scaled by this factor, and finally shifted further down in the remaining part of the time step.

This result can be generalized. Consider a system $\mathrm{d}x_t = a(x_t)\,\mathrm{d}t$ that possesses a discontinuity at $x = \xi$, such that $\lim_{x \to \xi^-} a(x_t) = \alpha_1$ and $\lim_{x \to \xi^+} a(x_t) = \alpha_2$, any infinitesimal interval crossing the discontinuity will be instantly scaled by a factor α_1/α_2, if crossing from $x > \xi$ to $x < \xi$, and the factor is α_2/α_1 if the crossing is from $x < \xi$ to $x > \xi$. This scaling comes in addition to the smooth scaling by nonlinear dynamics described in section 4.2, which must be calculated for each side of the crossing point ξ separately, as described for

4.5 Various implementations of Path Integration

In addition to the procedure discussed above, there is (at least) one other possible way of discretizing the mapping from t' to t for the integral. Instead of a uniform partitioning of the region $U_n \subset \mathbb{R}$ in equation (4.5) and stepping backwards, one could estimate the original region Ω', perform a uniform partitioning of this and step forwards. In this case, there is no correction factor in the last variable. In other words the Jacobian will just be $|J_{\tilde{g}}|$ instead of $|J_g|$, and only correct for the integration of the delta-functions. This could result in an improved accuracy for the numerical method, depending on the complexity of the domain Ω' and the approximation method. Also, since the Jacobian over the variables without direct input noise in many cases will be constant or closer to a constant, the time step might be made longer.

Previous implementations of PI, see [20], [34], [50] has mostly been based on this method. When there is only one noise process and a one-dimensional integral, the domain Ω' will be a one-dimensional curve. This has been approximated by a straight line between the chosen end points of the integral. As the time step Δt is small, this is usually a good approximation. The deviation between the points reached when stepping forwards and the original domain Ω is then small, and has been neglected.

However, for highly nonlinear problems, especially systems containing signum-kind nonlinearlities, piecewise smooth problems, and impact systems, the region Ω' could be far from a straight line. One could then improve the accuracy by fitting another function, i.e. a higher order polynomial. Though simple in its formulation, this approach easily increases the complexity and cpu time of the program code, since one now should discretize along a the curve length, and ensure a stable interpolation or curve fitting of Ω'. As shown in section 4.4, the Jacobian can be expressed analytically for some of these systems. Also, the FFT implementation of the PI method, described in section 5.3, is based on the path integral being represented as a pure convolution. This requires the full Jacobian method that is used in this thesis.

As a remainder, a more different version of PI is briefly discussed in section 2.2.2. There are also other methods that are denoted path integration that are based on only stepping forward in time. These are not discussed here. However, Vibeke Moe [34] discusses such methods in the beginning of her thesis, and one can otherwise look up cell-mapping techniqes, e.g. [51].

4.6 Example: Hysteretic system

The thesis of John Magne Johnsen[20] is finalized by an article where hysteretic systems are studied using path integration. However, the results clearly deviates from what is found in other papers, and probably what the authors found by other techniques, i.e. Monte Carlo simulations. The authors discusse possible reasons for their erroneous results, but discards all the explanations that are presented one by one. This article however resigns with the following;

> the results we obtain by the path integral solution technique deviate from those of the original sources. Of course, there may be several causes for this. If our results are in error, the two most likely causes for this seem to be intrinsic difficulties connected with the numerical calculation of the path integral solution of the specific examples considered, or to bugs in the computer program. The last factor does not appear to be involved. This conclusion is based on the fact that the methodology and the computer program have been tested on a variety of two-dimensional (2D) and three-dimensional (3D) problems which have analytical solutions, and in all cases complete agreement has been found.

The article then cites three papers where simpler SDEs have been investigated with PI. However, in all the test cases, the correction factor due to integration of the delta function in the TPD, discussed in the preceeding section, is barely a constant, when implemented as discussed in section 4.5. For the hysteretic system, this is no longer the case. So this should be a perfect example to validate that this factor is important and gives better results.

After the article discussed above, it seems that no studies of hystretic systems with PI have been published. Computer code obtained from earlier PhD students in this field only solves systems where the delta integration factor is a constant, and does not correct for this constant, and no earlier work discusses the need for this correction of the path integral.

One of the hysteretic systems that were studied can be written in the following form;

$$\begin{aligned}
\dot{X} &= a_1(X,Y,Z) = Y, \\
\dot{Y} &= a_2(X,Y,Z) = 2\xi\omega_0 Y - aX - bZ + \sqrt{2\pi G_0}N, \\
\dot{Z} &= a_3(X,Y,Z) = \left(-\gamma|Y|\cdot Z|Z|^{n-1} - \beta Y|Z|^n + AY\right)/\eta, \\
\xi &= 0.1, \quad \omega_0 = 1.0, \quad a = \frac{1}{21}, \quad b = \frac{20}{21}, \\
\eta &= 1.0, \quad \gamma = 0.5, \quad \beta = 0.5, \quad A = 1.0, \quad n = 1, \quad G_0 = 0.1.
\end{aligned} \tag{4.19}$$

The partial derivatives for system (4.19) with the parametric values inserted, and the corresponding correction factor are

$$\begin{aligned} \frac{\partial a_1}{\partial x} &= 0, \\ \frac{\partial a_2}{\partial y} &= -0.2, \\ \frac{\partial a_3}{\partial z} &= -0.5|y| - 0.5y\,\mathrm{sgn}(z). \end{aligned} \tag{4.20}$$

Since noise enters through the second variable Y in equation (4.19), the contribution of the last row in equation (4.20) cannot be approximated by performing the path integral directly in the state space of the previous time step.

The implementation of PI in [20] gives the standard deviations $\sigma_x = 1.62$, $\sigma_y = 0.693$, and $\sigma_z = 0.444$, which clearly deviated from the results in [43] of $\sigma_x \approx 2.5$, $\sigma_y \approx 0.8$, and $\sigma_z \approx 0.55$. Note that the latter are approximated, both by the numerical method used in the paper, and because the numbers are found from manually inspecting figure plots. A quick implementation with the new PI method described in this text, using a grid of $128 \times 128 \times 128$, a time step $\Delta t = 0.02$, and using FFT to compute the convolution integral, gives $\sigma_x = 2.4053$, $\sigma_y = 0.8310$, and $\sigma_z = 0.5718$, which is very close to the results of the original article. The mass after each time step, before rescaling was $1 - 6.5 \cdot 10^{-5}$ after the transient time, which also indicates that the correction factor approximation is not very crude. As discussed in section 5.3, the FFT method is not exact, and gives some error, though this should be compensated by the higher grid resolution than in [20] which has between 42 and 62 grid points in each direction. Also, the sensibility to the grid resolution was tested in [20], and was found to be small. The stationary PDF of equation (4.20) does not have a complicated structure, so the interpolation method or grid resolution should not be an important concern. As in most other examples, a standard fourth order Runge-Kutta method was used, but no additional improvement of the accuracy of the time stepping scheme.

It is not clear if the code from [20] lacked the correction factor, or if there were other problems, and it is no longer available. However, the factor is ignored in the equation in the thesis where the PI scheme is derived for the several variables problem. To verify that this is the cause of the problem, the new PI program was run again, commenting out the Jacobian correction factor, but otherwise keeping the same settings. This gave the results $\sigma_x = 1.8133$, $\sigma_y = 0.7177$, and $\sigma_z = 0.4510$, quite close to the results in [20]. So the author concludes that this is the cause of the error. The probability mass after each iteration for this implementation is also further away from 1, namely $1 - 1.1 \cdot 10^{-2}$, which typically indicates a lack of precision.

By this, it is shown that PI is suited to solve also hysteretic systems and systems where strong nonlinearities appears in variables where noise does not enter directly, as long as the implementation includes a correct integration of the delta functions of the TPD. Further, the study indicates that the approximation of the correction factor is quite good for this quite nonlinear system. Some devation should be expected, since the numerical method is

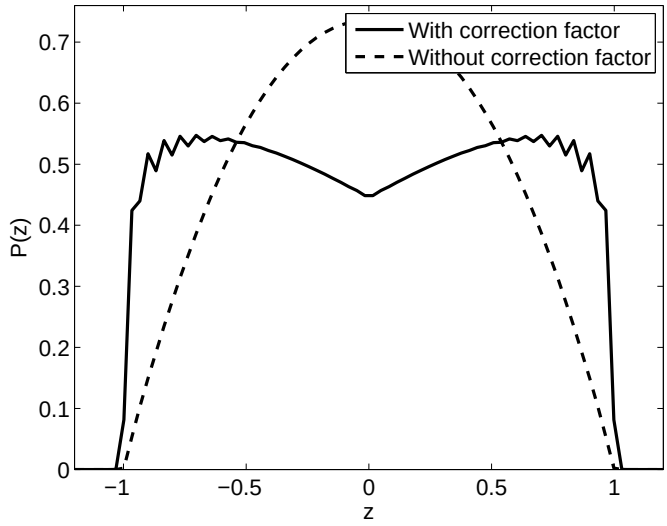


Figure 4.1: Marginal PDFs of the last state space variable for the hysteretic system. One simulation, with correction factor, is assumed to be a fairly good approximation, the second curve is obtained when ignoring this factor. The oscillations close to ± 1 for the good approximation is an error caused by interpolation problems in this region of the full 3D PDF, which is almost inevitable with the steep gradient which is also seen in the marginal plot.

based on this approximation, in addition to interpolation and a discretized time. Finally, this study is an indication that this correction factor is a new and important development, even though the mathematical idea behind is quite simple.

Note that although only standard deviations are reported here, the erroneous integration of the delta functions also changes the shape of the PDF quite a lot, both in the transient and in the stationary situation. Here, the marginal PDF in the first two state space variables x and y are close to Gaussian, and only the standard deviation is wrong. However, the marginal PDF of the filter z is very different in the two cases, as is shown in Figure 4.1.

4.7 Example: van der Pol equation with filtered noise

A good and more dramatic example to show how the correction factor is necessary to obtain a correctly shaped stationary PDF is a van der Pol (vdP) equation (4.13) from section 4.3 driven with some kind of filtered noise, now written

$$\ddot{X} - \mu(1 - X^2)\dot{X} + X = f(t). \tag{4.21}$$

It is known that the vdP system for $f(t) = 0$ has a limit cycle, and with $f(t) = \gamma N_t$ gives a PDF over this limit cycle with two distinct peaks. Many researchers also add a cubic term to systems similar to equation (4.21) to obtain Duffing-van-der-Pol equations, or harmonic

forcing functions to obtain chaos or other effects. Hence, one should know that this system can be solved correctly with PI.

The $(1 - X^2)$ term in equation (4.21) will also appear in the correction factor. If white noise enters directly through $f(t)$ in equation (4.21), this can be covered by the integral, when implemented as described in section 4.5. However, if $f(t) = Z$, where Z is defined through another SDE, the nonlinearity will appear in the delta function.

So a simple filtered noise example based on equation (4.21) can be written

$$\begin{aligned} \mathrm{d}X &= Y\mathrm{d}t, \\ \mathrm{d}Y &= (\mu(1 - X^2)Y + X + Z)\mathrm{d}t, \\ \mathrm{d}Z &= -\kappa Z\mathrm{d}t + \gamma\mathrm{d}W. \end{aligned} \tag{4.22}$$

This system has been studied in [50], but only for one choice of parameters, namely $\gamma = 1$, $\kappa = 0.5$, and $\mu = 0.05$. It is reported that the results give good results compared to Monte-Carlo simulations. This should mean that the correction factor has been implemented well for the work in this thesis, although it has not been described. However, only marginal log-densities are shown for this system for some reason, and the marginals deviate very little from a Gaussian distribution. The reason for the almost Gaussian shape is the very small value of μ, which makes the nonlinearity almost neglectable. It is also much smaller than the $\mu = 0.25$ used in the discussion of the 2D system in the same thesis.

A 2D plot of the joint PDF $f(X, Y)$ clearly shows the effect of the correction factor. First, choose $\mu > 0.1$, $\kappa > 0.1$ and $\gamma = 1$ so that the nonlinearity dominates. The correct PDF, obtained with the correctly implemented version of PI, then has a cycle with two peaks. Further, it is zero or has a minimum at the origin. Without the correction factor, the author could not obtain this minimum at the origin. Instead, the stationary PDF becomes flat over the area confined inside the cycle described above and zero outside.

A correctly shaped 2D marginal PDF is shown in [50] for a Duffing/van der Pol oscillator with a linear oscillator equation for $f(t)$,

$$f(t) = Z, \quad \ddot{Z} + \omega_z^2 Z + 2\kappa_z\omega_z\dot{Z} + W = 0, \tag{4.23}$$

is reported in [50]. This is the only result from previously published work with PI that the author cannot reproduce without the correction factor.

4.8 Periodicity in SDE and PDF

Some general comments should be made for SDEs that are periodic with a period $t_p = 2\pi/\omega$, e.g. containing a term $\cos(\omega t)$. Referring back to the numerical solution procedure based on the Markov operator, it is clear from an inspection of the stochastic kernel in equation (4.5) that $H_{tt'}(\cdot, \cdot) = H_{t+t_p,t'}(\cdot, \cdot)$. The associated Markov operator P_{ts} is therefore also periodic in the same sense with period t_p. This leads, as one should expect, to eventual periodicity of the propagated densities in the case of positive damping, that is $\delta > 0$. Hence

it is assumed that $f_{X\dot{X}}(x,\dot{x};t) = f_{X\dot{X}}(x,\dot{x};t+t_p)$ when $t \geq t'$ for a sufficiently large t'. Letting $T = mt_p$ for $m \in \mathbb{N}$, one obtains the asymptotic relation

$$\lim_{m,T\to\infty} F_{\hat{X}(T)}(\xi) - \exp\left(-m\int_{t'}^{t'+t_p} \nu_X^+(\xi;t)\,\mathrm{d}t\right) = 0. \tag{4.24}$$

Introducing the averaged density

$$\bar{f}_{X\dot{X}}(x,\dot{x}) = \frac{1}{t_p}\int_{t'}^{t'+t_p} f_{X\dot{X}}(x,\dot{x};t)\,\mathrm{d}t, \tag{4.25}$$

clearly leads to the relation

$$\int_{t'}^{t'+t_p} \nu_X^+(\xi;t)\,\mathrm{d}t = t_p\int_0^\infty \bar{f}_{X\dot{X}}(\xi,\tau)\tau\,\mathrm{d}\tau \tag{4.26}$$

The expression for the extreme value distribution can now in the limit $T \to \infty$ be rewritten as

$$F_{\hat{X}(T)}(\xi) = \exp\left(-\bar{\nu}_X^+(\xi)T\right) \tag{4.27}$$

where the average upcrossing rate is defined as follows

$$\bar{\nu}_X^+(\xi) = \frac{1}{t_p}\int_{t'}^{t'+t_p} \nu_X^+(\xi;t)\,\mathrm{d}t = \int_0^\infty \bar{f}_{X\dot{X}}(\xi,\tau)s\,\mathrm{d}\tau. \tag{4.28}$$

Hence, for prediction of extreme values, it is the average density that is important and not the density at a specific point in time, which is the information contained in the Poincaré map. Another important aspect here is that the concept of stationary density makes perfect sense after averaging, and there exist criteria which allows one to conclude the existence of such average stationary densities for a given system of the type studied here [29].

Chapter 5

Combining PI and interpolation

With the interpolation routine, the numerical implementation of the PI routine is complete. However, there are a number of important considerations to be made at this stage. Unfortunately, the choice of interpolation routine can be crucial to obtain a reliable result, especially for highly nonlinear systems. This chapter investigates most of these challenges, but also some opportunities for efficient computer programming that appears. By the end of the chapter, some examples are presented where combining interpolation with PI is a special challenge. This is also the case for some of the appendices. Appendix B requires an angle as a state-space variable, and the nonlinearities in appendices C, D, E, and F all required some special consideration for the nonlinearities. For an overview at this point, it might be an idea to check the float diagram of the PI routine on page 4 of appendix A, which is the article [32].

5.1 Choice of global interpolation method

For simple additive noise, the simple one-dimensional system from equation (1.16), and the similar two-dimensional system from (1.25) have been run using cubic B-splines and rectangular splines in a uniform grid, as discussed in section 3.4, and the corresponding tension splines from equation (3.6). The result is, for all the chosen parameter values, that cubic B-splines give slightly better results than the rectangular splines and tension splines with low tension, $w < 5$. For splines with higher tension, the final PDF, especially for the two-dimensional case, deviates more from the correct stationary distributions.

It should be mentioned that one could use a local grid for interpolation of especially difficult areas. This could cover some subset of the state space, while a coarser global grid handles interpolation outside, or different grids can cover disjoint parts of the state space. This idea has not been implemented in this project, basically because the typical behaviour of the systems studied here gives a PDF which needs the same resolution in basically the whole state space.

5.2 Eigenproblem formulation

If the SDE is autonomous, the PI trajectories will be the same for every iteration, and also the values of the basis for the interpolation routine will be the same. For most situations, it will be much more efficient to compute all of this at once in the beginning, and then using these pre-computed values at each iteration.

Remember from equation (3.5) that the B-spline interpolation is a linear combination of the B-spline coefficients, and that these coefficients are expressed as a linear combination of the interpolated values. This means that equation (4.10) can be further rewritten. For simplicity, start with a one dimensional SDE and a uniform discretization of the path integral. Further, the PDF at the previos time step is represented as $\widehat{f}(x')$. Then $\tilde{\mathbf{x}}$ from equation (4.10) can be skipped, $x_n = x$, and one can write

$$\begin{aligned}
\xi_{j+1} - \xi_j &= \Delta\xi_j, \quad j = 1, \ldots, M-1, \\
\alpha(x, \xi_j) &= f_{tt'}(x|g(\xi_j))|J_g(\xi_j)|\Delta\xi, \\
f(x,t) &= \sum_{j=1}^{M-1} \alpha(x,\xi_j) f^\dagger(\xi_j) \\
&= \sum_{j=1}^{M-1} \alpha(x,\xi_j) \widehat{f}(g(\xi_j)) \\
&= \sum_{j=1}^{M-1} \alpha(\xi_j) \sum_{k=1}^{K} c_k B_{k,j}(g(\xi_j)) \\
&= \sum_{k=1}^{K} c_k \sum_{j=1}^{M-1} \alpha(\xi_j) B_{k,j}(g(\xi_j))
\end{aligned} \tag{5.1}$$

The sum $\sum_{j=1}^{M-1} \alpha(\xi_j) B_{k,j}(g(\xi_j))$ does not depend on the earlier PDF, and can be calculated in advance. If one further includes the spline matrix S from equation (3.4) the PI iterations can be written in a linear form;

$$\mathbf{v}_{(j+1)} = AS^{-1}\mathbf{v}_{(j)}. \tag{5.2}$$

Equation (5.2) defines the (incremental) transition matrix, AS^{-1}. So now A is the matrix that maps the spline coefficients from the PDF at iteration $i-1$ to the values at i by the inner sum of equation (5.1). This extends to higher dimensional systems, also if the number of state-space variables where noise enters r is less than the dimension n. One should then use a natural ordering of the data values. That is just a way to create a one-dimensional structure from a rectangular grid. The natural ordering for a two-dimensional grid with indices $\{j,k\}_{j=1,\ldots,M_x,k=1,\ldots,M_y}$ would give indices $\{l\}_{l=1,\ldots,M_xM_y}$ where each l uniquely represents one index pair (j,k). Storing data sequentially with repect to the first index gives the formula $l = j + (k-1)M_x$, while it should read $l = (j-1)M_y + k$ if one wants sequential storage wrt the second index. If the PDFs values $\mathbf{v}$ are represented

as vectors, using a natural ordering of the grid points, A and S become ordinary matrices also in the higher-dimensional case. This method is implemented in a very efficient way in the numerical code developed in this project. A similar optimization was developed by John Magne Johnsen in [20], but the implementation is unfortunately no longer available.

Since the SDE is assumed to be autonomous, there exists a stationary PDF for this system unless the diffusion term pulls mass off to infinity. If this PDF can be well represented by the interpolation method in the discretized truncated computational domain, it is expected that the PI method will converge to a PDF $\mathbf{v}^*$ that is close to this solution.

It is a fact from linear algebra that the $\mathbf{v}_{(j)}$ in equation (5.2) for large j converges to the eigenvector assosiated with the largest eigenvalue of AS^{-1}, or if the largest eigenvalue has higher multiplicity to a linear combination of the associated eigenvectors. Since the eigenvectors are orthogonal, this implies that if only one eigenvector exist for the largest eigenvalue, it will be the unique solution of the PI algorithm, except for some constant factor to ensure positivity and mass 1. In other words, given the transition matrix AS^{-1} it will for many cases be possible to show that any initial distribution will converge to the same solution. If this eigenvector attains both positive and negative values, the interpolation method is not stable. All other eigenvectors, that are stationary points of the system, will in this case have negative values and add up to zero.

Do also note that the largest eigenvalue has a direct interpretation for our case. It is simply the probability mass from one iteration to the next when the stationary PDF is obtained. So this is a number that should be very close to 1.

The stationary eigenproblem could also be presented wrt the spline coefficient vector $\mathbf{c} = S^{-1}\mathbf{v}$. Then S becomes a mass matrix and A a stiffness matrix.

$$\lambda S\mathbf{c} = A\mathbf{c}. \tag{5.3}$$

If a more direct solver is used to find the eigenvector, the eigenproblem can be modified to amplify the solution corresponding to the eigenvalue closest to $\widehat{\lambda}$;

$$\lambda S(A - \widehat{\lambda}S)^{-1}\mathbf{v} = \frac{1}{\lambda - \widehat{\lambda}}\mathbf{v}. \tag{5.4}$$

Or, the pseudo-inverse of $(A - S)$ would be a good candidate for a preconditioning matrix for this problem.

A last method that should be considered in future work is the QZ-method which is based on a factorizations of A and S;

$$AQ = ZT, \tag{5.5}$$
$$SQ = ZU, \tag{5.6}$$

such that Q and Z are unitary (n, k) matrices, and T and U are upper-triangular (k, k) matrices for some $k < n$. From the eigenvector for the generalized eigenproblem with T and V, one can approximate or calculate the eigenvectors of interests with a few different methods. For this and other similar algorithms, see Golub and van Loan, [16].

For high-dimensional problems, the matrices are so large that computation of eigenvalues and eigenvectors might be impractical. The straight-forward iterations from an initial distribution might be the best option. Even though the A matrix will be very sparse, it will require more memory than all other parts of the program. The current implementation of PI only stores non-zero values of A as one long array, and has another integer array that maps the indices and values of A.

5.3 Path integration by Fourier transformation

Recently, the PI technique has been implemented by performing the convolution integral by a Fourier transformation. This idea seems to have been persued simultaneously at NTNU [26] and in Palermo [3]. The first assumptions needed, is that only purely added noise terms enter the SDE, i.e. that the standard deviation of the stochastic term is constant over the state space at a specific time. Note that the stochastic term could be time dependent.

Numerically, the Fourier transformation will be performed by a discrete Fourier transformation (DFT), most efficiently implemented as the routine known as fast Fourier Transform (FFT). Say the original sequence is $\mathbf{x} = (x_1, x_2, \ldots, x_M)$ and $\widehat{\mathbf{x}} = (\widehat{x}_1, \widehat{x}_2, \ldots, \widehat{x}_M)$ is the correponding DFT sequence. The forwards and backwards DFT is then written

$$\widehat{x}_j = \frac{1}{M} \sum_{k=1}^{M} x_k e^{-i2\pi(j-1)(k-1)/M}, \text{and} \tag{5.7}$$

$$x_k = \sum_{j=1}^{M} \widehat{x}_j e^{i2\pi(j-1)(k-1)/M}, \tag{5.8}$$

respectively.

As long as the forwards and inverse DFT is performed using the same grid, and the characteristic function is known analytically, one should expect stable results. However, to avoid spurious modes from appearing in the solution, the transformed PDF and the characteristic function must be well represented by the Fourier expansion. In other words, incorrect oscillations from high-frequency components, especially in the tails, will occur if the PDF is discontinuous, has a discontinuity in the derivative, or in other ways enhance high-frequency components in the spectrum. The basic assumption of the Fourier transform is also a periodicity of the function with period equal to the length of the state-space domain. This implies that the limit value of the function and its derivatives at both sides of the domain must coincide to avoid a discontinuity. Mathematically,

$$\lim_{x \to x_{\min}^{+}} D^k f(x) = \lim_{x \to x_{\max}^{-}} D^k f(x).$$

A large number of FFT versions are available in various languages, to download from the internet or as built-in packages. A review of the different options is beyond the scope of this text, but it should be mentioned that not all FFT routines requires the number of points

to be a power of two as in equations (5.7) and (5.8), and that given some pre-calculation a DFT of size M can always be computed in the order of $KM \log_2 M$ operations. Also, it is usually a good option to extend the vector of interest by zeros (padding) to obtain a length of $M = 2^m$ for some integer m. However, since the FFT basically is the bottleneck for the CPU-time of the PI algorithm, one could then increase the grid resolution up to M grid points in the variable(s) where noise enters without much loss of time.

In this project, an FFT algorithm developed by D.M. Monro [36] has been used. The program computes the DFT of $(x_1 + ix_{M/2+1}, x_2 + ix_{M/2+2}, \ldots, x_{M/2} + ix_M)$ into say $(z_1, z_2, \ldots, z_{M/2})$. Based on the observation that equation (5.7) has a symmetry of $\widehat{x}_j = \widehat{x}^*_{M/2+j}$, [35] shows that $4\widehat{x}_j = z_j + z^*_{n/2+j} - i \exp[-i2\pi(j-1)/n(z_j - z^*_{n/2+j})]$. In other words, one splits the sequence of values into two halves, which are treated as the real and imaginary parts of a DFT of half the length. The true DFT can then be reconstructed by a linear unraveling of the real and imaginary parts of the final result. Before the inverse routine is invoked, this linear operation is also inverted. Since only the result of the convolution and not the DFT itself is of interest, this scrambling and unscrambling can be avoided by performing the multiplication as a slightly more complicated linear transformation.

5.4 Interpolation in log-domain

A major part of the thesis of Vibeke Moe (1997) [34] is devoted to an idea that interpolation of probability densities should be performed in a logarithmic domain. Assume as before that $v_j = f(x_j)$ is known for a sequence of grid points $\{x_j\}$, $j = 1, \ldots, M$. Instead of constructing an interpolant based on the v_js, construct the interpolant $\widehat{g}(x)$ based on $\widetilde{v}_j = \log(v_j)$. Then, compute $\widehat{f}(x) = \exp(\widehat{g}(x))$. This $\widehat{f}$ is clearly an interpolated function, and it will be positive everywhere. Results presented in [34] are convincing, but the succeeding research performed by Christian Skaug [50] did not follow this path, and the idea has not been mentioned afterwards.

The idea is intriguing, since many standard probability densities can be written as $f(x) = K \exp(g(x))$ with K constant and g a polynomial of low degree (typically less than five), it intuitively seems reasonable that interpolation by polynomials of a sequence of values $\widetilde{v}_j = \log(v_j) = \log(K) + g(x_j)$ should be more accurate than on a sequence $v_j = K \exp(g(x_j))$.

This would certainly be correct if $g(x)$ is perfectly represented by the interpolant, however this is rarely the case. It should be clear that a linear combination of cubic polynomials cannot equal a polynomial of higher order anywhere. Global polynomials of lower degree can be perfectly represented by B-splines, as shown in [10] who also gives the general spline representation for these splines. A problem is that any numerical error in the PDF at the knot points will cause the interpolant to be of full degree, and that generally the PDF of a nonlinear problem is not an exponential of a polynomial.

The major problem with this method is the handling of the boundary of the calculation domain. As the PDF approaches zero, the logarithm will go to $-\infty$. In addition, assuming

that this curve is a polynomial shows that also the derivative, the curvature, and all other non-zero derivatives will have large absolute values. Such curves are generally difficult to interpolate. Moe discusses many of the problems encountered with various ways of defining appropriate boundary conditions, but finds no general solution.

Experiments show that the exponential of an interpolant amplifies over-estimated interpolation error. A good analytical description of this phenomenon is difficult to obtain. However, observe that if $\widehat{g}(x) = \log(f(x)) + \varepsilon(x)$ where ε is the error, $\widehat{f}(x) = f(x)\exp(\varepsilon(x))$. So the error is basically proportional to $f(x)$, and $\exp(|\varepsilon(x)|)$ is a bigger error than $\exp(-|\varepsilon(x)|)$. Even if one is interested in the tail behaviour, experiments have shown that an accurate description of the PDF where most of the probability mass is located is necessary.

Another appropriate comment is that the exponential function can be well approximated by local polynomials, and the error by a B-spline will be about the same as for the truncated Taylor polynomial of the same order. This observation is in fact similar to the observation that the tension splines from section 5.1 also locally behave similar to polynomials. The boundary conditions $f(\mathbf{x}) \to 0$ as $||\mathbf{x}|| \to \infty$ are simpler to work with, and as Vibeke Moe states in the thesis abstract, "a particular efficiency enhancing scheme can no longer be used" with the logarithmic method. In conclusion, the method's weaknesses are so serious that it cannot be recommended for general problems.

5.4.1 Local modification to remove instabilities

It is seen that cubic and rectangular B-splines work well with PI, except in cases where spurious oscillations occur locally, with devestating results. How these problems were avoided earlier has not been documented, but code from C. Skaug includes setting negative interpolation values to zero and some removal of singular delta-shaped peaks near boundaries of the domain.

The third and fourth order B-splines are chosen because of their ability to curve. When locally this effect is not wanted, it seems reasonable to use a linear interpolation instead. A linear interpolant is easy to compute, so the only problem is to find a well-functioning criteria for deciding which intervals a linear interpolant should be used. It seems reasonable that this decision should be based on the PDF values, which means that this could change from one time step to the next. Negative interpolated values will be easy to spot, but unreasonable oscillations could also be all positive, and the largest values in spurious oscillations cause just as many problems for the correct convergence of PI as negative values.

If the PDF has a slow variation compared to the grid points, the B-spline coefficients should not deviate too much from the grid points, as the interpolant should be close to the Bézier curve generated by the PDF values. One would also expect that the spline coefficients are positive. Oscillations where the interpolant attains values smaller than the values of the closest nodal points must be caused by at least one negative spline coefficient. The spurious oscillations seen in numerical simulations with PI have had negative spline coefficients neighbouring large compensating spline coefficients. From this, the idea

came to use linear interpolation where negative coefficients were observed at one of the nearest neighbouring grid points. Results indicate that very few SDEs now cause the huge interpolation errors, and spurious oscillations are strongly reduced.

If the interpolation is stable in most of the domain with a B-spline, this the time efficiency scheme from section 5.2 can be applied, with just a direct integration in the problematic areas.

5.5 Path integration and linear operators

One feature of PI that has not been discussed in previous work by Naess et al. is the possibility of obtaining additional information about the probability density with just a small increase in cpu time. Let G be an operator working on the last state space variable, e.g. differentiation or integration with respect to x_n.

The total probability law, equation (4.3), gives

$$Gf_t(\mathbf{x}) = G\int_O f_{tt'}(\mathbf{x}|\mathbf{x}')f_{t'}(\mathbf{x}')\,\mathrm{d}x' = \int_O Gf_{tt'}(\mathbf{x}|\mathbf{x}')f_{t'}(\mathbf{x}')\,\mathrm{d}x', \tag{5.9}$$

so the requirement on G is really just that it commutes with the integration operator. Since $f_{tt'}$ usually is an analytical expression or fairly easily evaluated, $Gf_{tt'}$ should have the same property. Equation (5.9) can be integrated simultaneously with the original PI operator, the only modification needed might be to ensure a numerically valid approximation of also this new integral by adjusting truncation limits and the discretization appropriately.

The reason linear operators working on other state space variables has not been included in the discussion so far, is that their appearance in the delta functions and the restriction of the integration domain could cause problems. However, in the general theory - and when integrating in more dimensions, the argument should also be valid there.

Because interpolation has been a huge challenge in numerical PI, an especially interesting use of equation (5.9) is to compute the partial derivatives of the PDF in the x_N direction. These derivatives could then be used to find better interpolants, e.g. piecewise defined in each interval. This way the interpolant would also have correct derivatives at the interpolation points when starting at the correct distribution at a given time. This approach would still create an interpolant with continuous derivative, and also continuous curvature if this is included, even if its pieces are very different kinds of functions. This gives a huge flexibility, but requires that one has a method to choose and compute suitable functions that can fulfill these constraints.

This approach would be especially useful for processes where the last state space variable has a marginal PDF with large slope or curvature that causes problems with B-splines, e.g. the Rayleigh-process where the PDF is almost vertical for values close to zero.

Another feature is that the adaption of the piecewise function is purely local, so that the effect of smearing out local structure seen with cubic B-splines might be reduced. This would be of particular interest for systems with a rich structural detail, i.e. chaotic

attractors. This also implies that there is no global sensitivity to end point conditions, which clearly is an advantage.

With the first derivative, one could still use cubic polynomials - the main difference from cubic B-splines is then that the computed derivatives are used as the constraint instead of requiring a continuous second derivative. Scaled and translated to the interval $x \in [0, 1]$ where $f(0)$, $f(1)$, $f'(0)$, and $f'(1)$ are known, the intermediate cubic curve is

$$\begin{aligned}\widehat{f}(x) = &\{f'(0) + f'(1) + 2(f(0) - f(1))\}\, x^3 \\ &+ \{3(f(1) - f(0)) - 2f'(0) - f'(1)\}\, x^2 + f'(0)x + f(0).\end{aligned}$$

However, one could now easily check each interval for false local minima, maxima or inflection points, and choose a different function if this seems more appropriate. The coefficients of the cubic polynomial are linear functions of the PDFs values and derivatives, and hence a fast numerical method based on precalculating the transition matrix as in section 5.2 is also feasible. This would also be the case for a large class of "basis" functions. However, this would requre a recalculation of parts of the transition matrix every time a local function is substituted by a function of a different class.

A main drawback is that there is no obvious way of extending this interpolation procedure efficiently to more than one dimension. Basically, one would have to interpolate along grid lines in the last state space variable first, and then compute B-splines coefficients and interpolated values in the other variables at the points of interest, which seems to be time consuming. Even if the partial derivatives in every direction could be obtained using equation (5.9), the complexity of finding piecewise functions fulfilling all end point criterias would be huge.

5.6 Parametric representation

One of the problems of applying the PI technique to higher dimensional systems is the representation of the PDF by the probability density function at some (grid) points in the state space, which leads to the need for high dimensional interpolation.

For complex systems, there is no reason to believe that the joint PDF can be well represented by some analytical function, and finding such a representation and fitting parameters would be a difficult task.

However, it might be possible to fit a model in one or very few state-space variables at specific grid points in the remaining variables. Say the system contains n variables $x_1, x_2, \ldots, x_n$, all of them evaluated at a specific time, say t_0, and where the last k are weakly connected to the rest of the system and/or not of particular interest for e.g. tail probabilities. The idea is then to find a model for the conditional probability density $f(x_{n-k+1}, \ldots, x_n | x_1, \ldots, x_{n-k})$, in addition to the probability density for this grid point in the first variables $f(x_1, \ldots, x_{n-k})$. By storing the latter number, one can still easily perform integrals over the first variables, compute probability mass etc. In fact, since the domain in the last variables now can be arbitrarily large, the PDF should also be more accurate.

If one can find a model that gives a good representation with fewer parameters than the number of grid points otherwise required, there is a potential saving here, especially in reducing computer memory, but also in code complexity and possibly in shorter CPU time. An example of utilizing this, is discussed at the end of section 6.12.

This approach requires that it makes sense to interpolate the parameters from the grid points to the full state space, and that this can be done in a numerically stable way. E.g. if some parameters could become very large, or have a steep behaviour close to the borders of the domain, one could try representing the PDF by the inverse of this parameter instead, e.g. storing $1/\sigma^2$ or $1/\sigma$ instead of σ^2 or σ for a Gaussian model.

5.7 Interpolation error and PI step size

The PI method described in the previous chapters to a large extent decouples the spatial resolution and the time step. However, this is not completely true. Assume that the time variable should go over a fixed interval $t \in [t_0, t_1]$. For a given spatial resolution, there will for most problems be a limit to how many partitions this interval can be divided in, i.e. how small the time steps could become, before the numerical error increases. The reason for this is the accumulated effect of the interpolation error.

The best way to visualize this, is from a one-dimensional example where the exact solution to the system is known, say the Langevin equation. The exact solution is an exponential function. If this is sampled in discrete points, the interpolant will not be exact, except at the sampled points. If the simulation is started from the exact distribution, and the time stepping method is exact, there will be an error and thereby a small adjustment of the pdf due to interpolation error in the first iteration, for any time step.

The size of an interval containing $(100 - \epsilon)\%$ of the probability mass is approximately proportional to $\sqrt{\Delta t}$. This is where the significant contribution to the path integral

When the time step is large, the path integral for a given point gets a significant contribution from a large interval where the interpolant is too large some places and too small other places. This partially averages out the interpolation error, and should make it signifcantly smaller than the other errors of the system.

By choosing the time step sufficiently small, the path integral for a given point gets the significant contribution from a small interval. If the interpolant is smaller than the exact PDF in this interval, the integral will give a too low value, and vice versa. Now, repeat this experiment, but with half the time step and two iterations. The adjustment of the PDF at the first iteration is so small that the second interpolant is basically equal to the first, and the two steps both contributes with approximately the same error, so the error is doubled. Further, the interval that one integrates over is smaller, so local over-estimation or under-estimation by the interpolant is not as likely to cancel out for one specific point. One could argue that since the time step is smaller in the second case, the interpolation will be performed closer to grid points, at least in the variables not integrated over, and that this should reduce the interpolation error. However, the contribution from the neighbouring grid points basically is what represents the evolution of the PDF. In other words, although

the interpolaton error might be smaller, the change of the PDF from one time step to the next depends strongly of the influence of the neighbouring grid points. Their contributions still have the same error.

Exact analysis of the total numerical error is difficult.

5.8 Discontinuities in the SDE and piecewise systems

In some cases, a direct implementation of PI will perform well for discontinuous problems. However, there are important pitfalls here. One of them illustrates one way time step and coupling could be related. Consider the following SDE:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[(1+R\,\mathrm{sgn}(X\dot{X}))\dot{X}\right]+\Omega^2 X=\gamma N_t. \tag{5.10}$$

The parameter $R\in(0,1)$ describe the amount of nonlinearity in the moment of inertia. Equation (5.10) is transformed to the following coupled model.

$$\begin{aligned} Y &= (1+R\,\mathrm{sgn}(X\dot{X}))\dot{X},\\ \dot{X} &= \frac{Y}{1+R\,\mathrm{sgn}(XY)},\\ \dot{Y} &= -\Omega^2 Y+\gamma N_t. \end{aligned}$$

This, and a few other similar systems, are studied by PI in appendices C and D.

Not only is the system discontinuous, but the resulting PDF will also be discontinuous at $X=0$. At $Y=0$ there is no discontinuity, since $\dot{X}$ goes to zero from both sides. A global smooth interpolation method will not handle the discontinuity well, therefore consider a local (subdomain) method. Since the system is symmetric, the PDF should be the same at the two points (x,y) and $(-x,-y)$, it suffices to focus on just one half plane. Note that the deterministic trajectories of the system are ellipses in all four quadrants of the grid. There is no additional damping, so the accuracy of the final PDF will depend on how well the nonlinearity at $x=0$ is modelled.

It seems reasonable that there should be no grid points at $x=0$, since the PDF is discontinuous there. So the first positive gridpoint is located at $x=\xi<\Delta x$. Assume first that we use very small time steps. The trajectories backwards from the grid points $x=\xi$ will then never reach $x=0$. The only interpolated values we get, will be based on an extrapolation of the grid values at ξ, $\xi+\Delta x$ etc. However, all these values will eventually depend on the extrapolated values again. The splitting of the domain into different interpolation areas hence breaks the dynamics at $x=\xi$. So a slight under-estimation in the extrapolation, will give too little flow from $x\approx 0$ to $x=\xi$, and all grid points along or close to the following elliptic curve will be affected. The path integral for a given grid point will follow a line in the x',y' domain almost parallell to the x'-axis. This means that all the contributions to the integral will be based on basically the same extrapolation. Similarly, if the density is slightly over-estimated, the result is a globally extra mass.

For each trajectory crossing $x = 0$, section 4.4 has shown that the Jacobian delta-correction term would multiply the interpolated value by $(1+R)/(1-R)$. When R is close to one, this factor will be huge. If trajectories crossing $x = 0$ occurs only for grid points where $y > \eta$ for some $\eta > 0$, the $(1 + R)/(1 - R)$ would cause the PDF to turn into the shape of a discontinuous doughnut or torus with half-axis η in the y-direction. A slightly larger time step could similarly make the hole in the resulting torus smaller. With a long time step, the trajectory ends up in the other part of the domain. Again, extrapolation will be a problem unless the trajectory ends up at an x-value $x < \xi - \Delta x$, i.e. passes the largest negative grid point in x.

Of course, all the trajectories will hit the nonlinearity at some point and get the "boosted" effect. One solution is to make two separate domains, one with the first grid point at $x = 0^+$, the next at $x = \Delta x$ etc., and one grid where $x = 0^-$ is the highest grid value, then $x = -\Delta x$ etc. That is, the grid values represent the left and right limits at $x \to 0$, respectively. The problem with this method is the endpoint condition. The not-a-knot condition from section 3.3.1 gives a polynomial spanning two intervals, interpolating at three points in the x grid. This easily causes oscillations in this area, and switching to a linear approximation did not give good results either. Again, the main difficulty is that the system is so dependent on high accuracy in the interpolant around the discontinuity.

Another solution, that worked surprisingly well, was to make the grid non-uniform with high density of grid points around $x = 0$, and ignore that the PDF is discontinuous. The grid with M gridpoints from $x_{\min}$ to $x_{\max}$, $x_{\min} = -x_{\max}$, with M even, was initialized with the following algorithm:

- Choose the grid value closest to the discontinuity, $\Delta_{\min}x$.
- Calculate $\alpha = (\log(x_{\max}) - \log(\Delta_{\min}x))/\log(M/2)$.
- Grid point number k above $M/2$ is $x_{M/2+k} = \Delta_{\min}xk^{\alpha}$, $k = 1, \ldots, M/2$.
- The grid is made symmetric by $x_{M/2-k+1} = -x_{M/2+k}$, $k = 1, \ldots, M/2$.

To clearify, one could also write

$$x_{M/2+k} = \Delta_{\min}x \left(\frac{x_{\max}}{\Delta_{\min}x} \right)^{\log(k)/\log(M/2)} . \tag{5.11}$$

Equation (5.11) easily verifies that $x_{M/2+1} = \Delta_{\min}x$ and $x_M = x_{\max}$. The advantage of using the power law k^{α} for the grid is that the transition from a coarse to a fine grid is smooth, for a sufficiently large M. This seems to be an advantage for obtaining a stable interpolant.

Similar adjustments are also important for systems consisting of piecewise systems that are not discontinuous. One should consider to improve the grid resolution where the vector field is not smooth, and also use a divided time step to handle the dynamics across this area properly. However, the Jacobian is not an issue in that case.

5.9 Reliability and hard impact problems

A special case of discontinuity is typically obtained when one studies the reliability of a system, which here is the probability that the response stays within some domain U in the state space. To partly avoid the discontinuity, the author recommends to always calculate the conditional density for time t; $f(\mathbf{x}|\mathbf{X}_{\Delta t}, \mathbf{X}_{2\Delta t}, \ldots, \mathbf{X}' \in U)$, which allows $P(X_t \notin U) > 0$ and is smooth. If the stationary PDF $f(\mathbf{x})$ is small at $x \in \partial U$, i.e. on the boundary, the author's experience is that no specific adjustment is needed to ensure a stable interpolation. More on reliability, the numerical implementation, and examples can be found in appendix D.

The same situation occurs for impact problems without penetration. That is, at some place in the state space, every trajectory is reflected - typically at some value of the displacement, with an immediate change of sign in the velocity. Also here, if this rarely happens and the probability density has a low value at the impact boundary, one might be able to neglect this for the interpolation. Otherwise, a not-a-knot end condition would be reasonable. Also here, one should consider using a method to split the time step to obtain accurate results. One such impact problem is studied in appendix E.

5.10 Example: The Rayleigh Process

The Rayleigh process, equation (1.21), is a simple but illustrative one-dimensional SDE which has been studied by PI in [23] and [20], and it's used in the modelling of the force from random sea-way and wind. For simpler reading, the SDE can be repeated:

$$\mathrm{d}X_t = -\frac{\gamma^2}{2}\left(X_t - \frac{\kappa}{X_t}\right)\mathrm{d}t + \gamma\,\mathrm{d}W_t, \quad X_t > 0, \kappa > 0. \tag{5.12}$$

For this system, the exact TPD is known and reads with $q = \exp(-\gamma\Delta t/2)$

$$p(x,t|x',t') = \frac{x}{\kappa(1-q^2)} I_0\left(\frac{qxx'}{\kappa(1-q^2)}\right)\exp\left(-\frac{x^2+q^2x'^2}{2\kappa(1-q^2)}\right), \tag{5.13}$$

where I_0 is the modified Bessel function of the first kind. Using the exact TPD, PI performs well, as shown in the references [23] and [20]. Hans Christian Karlsen [23] worked together with the author to solve this system with a normal approximation for the TPD as earlier. The motivation was to solve higher-dimensional systems where one or two possibly coupled systems as in equation (5.12) works together on some structure. Karlsen showed that the method that gives equation (5.13) could not be used to construct a higher-dimensional TPD.

John-Magne Johnsen [20] showed that a characteristic time as he defined it, see section 2.1.1, cannot be found for this system. The problem is that as $X \to 0$, $\mathrm{d}X \to \infty$, so the deterministic term will always dominate over the stochastic term in some region close to zero. This is also the reason that the response X never actually will become negative. A discrete-step simulation with fixed time steps that does not exploit equation (5.13) could

give negative values. This is basically the first challenge with PI; the approximated TPD must be truncated in some way at $x = 0$ and $x' = 0$.

The deterministic part of equation (5.12) can be solved analytically. Thereby, the time stepping can be performed as described in section (2.1.2). Euler and Runge-Kutta schemes are based on the value of the rhs of the system in some points. However, this changes rapidly as $X \to 0$, so these methods easily diverge from the correct deterministic path.

Unfortunately, the problem caused by the deterministic time step dominating over the random term could not be efficiently and accurately solved by simple truncation of the normal TPD, variable transformation, or scaling. [23] Some good results were obtained by using very short time steps and very high grid resolution and a few other "tricks" like truncation at $x = 0$, but a stable method has not been found.

The PDF of the system has $f(x) = 0$, but a non-zero gradient, at $x = 0$. This causes problems for the zero-side interpolant from section 3.3.2. Also the not-a-knot from section 3.3.1 had instabilities, and local linear interpolants seemed to steadily shift the PDF to higher values of x, basically by under-estimating important parts of the left tail. In one dimension, an option was to estimate the left tail by a local exponential function or polynomial function, requiring $f(0) = 0$ and $f(x) > 0$ for $x > 0$. In the higher dimensional systems, Karlsen concluded that the shift of the PDF was sufficiently small that the effect on the other state-space variables could be neglected.

5.11 Six-dimensional path integration

Basically every numerical method solving SDEs to find a probability density is based on some discretization of time and space. Here the so-called "curse of dimensionality" comes into play. If M grid points or cells are needed in each dimension to obtain a sufficient accuracy, the number needed just to represent the PDF a system with n dimensions is M^n, so the total CPU time will at least be of this order multiplied by the CPU time for each grid point, and usually the number of iterations or time steps. This will be the case for methods using finite differences, finite element techniques, and cell-to-cell mapping, to mention a few.

It should be mentioned that statistics based on realizations of the process, such as Monte-Carlo methods, does not suffer from this curse of dimensionality, as long as the objective function is of lower dimension and can be evaluated in all or most parts of the state space. So a mean value or variance could be very easily estimated even for a high dimensional system with such methods, while a conditional density or the probability in a specific part of the tail of the PDF cannot be found.

With the continuous increase of computer power, both in terms of speed and memory, one would still expect that each new numerical method proposed would be able to handle systems of higher dimensionality than the older, or that well-known methods would continuously be extended in this direction. However, this does not seem to happen very quickly, indicating that the curse of dimensionality is a serious limitation for every numer-

ical method. One reason is simply that the complexity of the computer code needed is too high, and that the power law of increasing dimension is much higher than any power law of increasing computer capabilities. The FPK equation gives a linear combination of partial derivatives in all dimensions that has to be solved.

This section presents an approach to overcome the problem of programming PI code to solve a six-dimensional SDE, as well as the first response PDFs for such systems. To manage this, an "extra" computer interface has been implemented, in short a computer program producing the PI code based on the general structure of the interpolation, the time steps and the integral. The resulting PI code has then been compiled and run. To verify that this very complex program actually solves six-dimensional SDEs, it has been applied to simple test cases with known analytical solutions, and a nonlinear system of two coupled oscillators with filtered noise input. This does not mean that the program is limited to such simple systems, as the theory and numerical implementation otherwise does not differ from what has been presented earlier.

5.11.1 Numerical implementation in n dimensions

Of course, many parts of a code is easily extended to higher dimensions by simple copy and paste and the addition of extra loops around central parts of the program. However, some features of the code is more complex.

One of the key elements of this version of PI is the spline interpolation routine. The choice of end-point condition requires special handling of each of the domain boundaries. In one dimension the spline method will have three forms; one for each of the two end points and one for the middle segment. In two dimensions, there are nine cases; four corners, four sides, and the remaining main part of the domain. Similarly, in n dimensions one needs to consider 3^n different cases to avoid out-of-bounds errors and include any given boundary conditions. In six dimensions, this alone results in $3^6 = 728$ cases in an if-statement, which is difficult to keep track of without the assistance of a computer. In principle, every point on a one-dimensional cubic spline interpolant is the result of a linear combination of four basis splines with their respective coefficients. In two dimensions, there are 16 such neighbours contributing to each point, and in general, there are 4^n contributing basis splines for each point. Going down to rectangular splines in all directions, this number is 3^n. With 3^n cases for boundary control and approximately 3^n neighbours to evalute for each case, a direct coding of the linear interpolation will contain the order of 3^{2n} array evaluations.

The relative error caused by the approximations at the computational boundary increases with dimension. The reason is basically that the boundary and the low-probability areas of the PDF represent a larger and more important part of the dynamics and the numerical representation with the increase in dimensionality. To show this, first count the number of grid points that does not have a full set of neighbours, and assume for simplicity that there are M gridpoints in each direction. In one dimension, there are then 2 end points, and $M - 2$ inner points. In n dimensions with M^n gridpoints in total, the number of boundary points is found from the formula $2\sum_{j=0}^{n-1} M^{n-j}(M-2)^j$. This

is shown in table 5.1 for the first eight dimensions, which clearly shows that boundary points consitutes a large relative part of the computational domain. If also the grid points neighbouring the bounday points were included, this effect would be even stronger. The dependence on tail behaviour is similarly important for the dynamics. The total probability mass within regions with low probability will also normally increase with dimension. To illustrate this, consider a set of n independent zero-mean Gaussian variables with unit variance, $Z_1, Z_2, \ldots, Z_n$. If one truncates the PDF of one variable Z_1 at the quantiles $-z_{\alpha/2}$ and $z_{\alpha/2}$, the probability mass remaining is $1-\alpha$. If the same truncation limits are used for both variables in a joint PDF of Z_1 and Z_2, the probability mass remaining is smaller, namely

$$P(-z_{\alpha/2} < Z_1, Z_2 < z_{\alpha/2}) = P(-z_{\alpha/2} < Z_1 < z_{\alpha/2})P(-z_{\alpha/2} < Z_2 < z_{\alpha/2}) = (1-\alpha)^2.$$

Correspondingly, the fixed truncation limit gives a probability mass $(1-\alpha)^n$ for n random variables, a decreasing figure for increasing values of n. This means that a truncation of the computational domain at e.g. some fixed number of standard deviations would affect the PDF more for a high-dimensional case than for a low-dimensional.

Dim	Ratio of border points	$M=16$	$M=32$	$M=64$
1	$\frac{2}{M}$	0.1250	0.0625	0.0312
2	$\frac{4M-4}{M^2}$	0.2344	0.1211	0.0615
3	$\frac{6M^2-12M+8}{M^3}$	0.3301	0.1760	0.0909
4	$\frac{8M^3-24M^2+32M-16}{M^4}$	0.4138	0.2275	0.1193
5	$\frac{10M^4-40M^3+80M^2-80M+32}{M^5}$	0.4871	0.2758	0.1468
6	$\frac{12M^5-60M^4+160M^3-240M^2+192M-64}{M^6}$	0.5512	0.3211	0.1734
7	$\frac{14M^6-84M^5+280M^4-560M^3+672M^2-448M+128}{M^7}$	0.6073	0.3635	0.1993
8	$\frac{16M^7-112M^6+448M^5-1120M^4+1792M^3-1792M^2+1024M-256}{M^8}$	0.6564	0.4033	0.2243

Table 5.1: Number of grid points at the border of the domain for different equally partitioned dimensions and for different grid sizes.

There are also other methods to perform the integral with different or no interpolation, i.e. by representing or approximating the PDF or parts of it by a parametric model as proposed in section 5.6, or using moment equations to estimate a non-degenerated TPD that can be integrated without interpolation, though these methods have some additional drawbacks. Such methods might prove to be necessary for problems of even higher order than 6.

Another important aspect to consider for such complex programs is the possibility of reducing CPU time by intelligent programming. One example is the n-dimensional time step method. If the time step in some of the variables are independent of the state of the others, these could be taken out of the innermost loop(s), thereby significantly reducing the number of times these variables are computed. One typical example is a filter equation producing coloured input noise to an oscillator. The state of the oscillator does not affect

the filter. Hence, the different states and time steps of the filter should be computed first, then for each of these all possible oscillator states are included. This of course affects the order of which nested loops should appear in the code.

Scaling of variables also becomes an important issue to ensure numerical stability, or special considerations must be taken. As an example, say the grid is uniform between -0.01 and 0.01 in all six state space variables. The values of the PDF at the grid points must then be in the order of $64 \cdot 10^{12}$ in average to obtain a mass of one. If the PI code for rescaling first adds the PDF values, and then mulitiplies with the increments, numbers really start to be large. With, say, 20 grid points in each direction, there are $64 \cdot 10^6$ grid points in total. The sum of the PDF values before multiplication with the grid increments would then be $4.096 \cdot 10^{18}$. This is still far below the maximal range of a single precision real, which is about 10^{38}, but without scaling one could end up with overflow or underflow. Another issue is that adding many numbers the direct way requires high precision to be stable. In six or more dimensions, a large part of the grid is close to the boundaries of the domain where the PDF value is small. Each number here could be too small to increment the sum, but the total contribution could still be signficant. Inaccurate methods or low precision arithmetic could then underestimate the total mass just by roundoff error.

As always, the time step must be sufficiently small to ensure both that the dynamics is followed properly, and that the deterministic and stochastic terms are balanced through the characteristic time step discussed by Johnsen [20]. In many dimensions, there will be more requirements on the length of the time step. At the same time, choosing a very short time step will cause long time to reach stationarity, and increase the risk of accumulating errors from the interpolation. This problem also increases with dimension.

Finally, spatial resolution is a major concern in PI, and especially difficult when the number of dimensions is high. If two variables are strongly correlated, the marginal PDF in these two variables will be a thin structure, difficult to represent in a coarse grid. This problem cannot be seen by the one-dimensional marginal densities. Similarly, the two-dimensional marginal PDFs might not reveal that the structure of the PDF is thin in a four-dimensional grid and so on. This requires careful planning when finding SDEs that are possible to solve with PI when grid resolution is limited. To illustrate this, consider two coupled oscillators driven by a slowly varying random process. They will typically be highly correlated in displacement and velocity. An illustration is shown in Figure 5.1, where only a few grid points actually is inside the interesting part of the joint PDF, although the grid is reasonable in both variables. This could be resolved by a linear transformation, but still the variables could be strongly correlated with the filter equation. In many nonlinear systems, the structure of the response PDF could be skinny along other seemingly arbitrary directions. An example is the Lorenz attractor discussed in section 6.15. In such cases, it is typically no way of eliminating this resolution problem by variable transformation.

5.11.2 6D results

For the numerical computations, 16 grid points were used in each direction, giving a total of about $16.8 \cdot 10^6$ grid points. The PI method used FFT and a rectangular spline method.

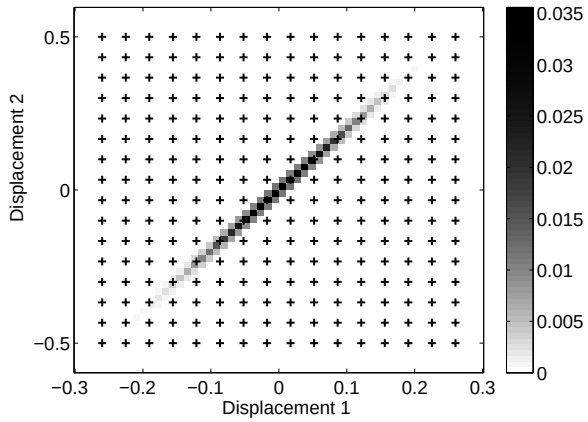


Figure 5.1: Joint displacement PDF of two coupled oscillators, shown for illustration. A grid discretized in 16 points in each direction could seem reasonable when only regarding the marginal densities. Unfortunately, only a few grid poins actually intersect the interesting part of the joint response PDF.

From table 5.1 it is seen that more than half of the gridpoints actually are located at the boundary of the domain, which potentially could cause numerical errors. Since they contribute significantly to the results, it should be clearly specified which method is used there. The boundary approximation used is the one described in section 3.3.2 with extrapolation extending out to the support of all non-zero basis functions. This choice is mostly due to the low number of grid points. With few grid points and a uniform grid, the outermost gridpoint cannot be too far out in the tail. A significant probability mass could be just outside this gridpoint, and assuming the derivative to be zero (or any other fixed value) could be very misleading. The not-a-knot endpoint method is not available for rectangular splines. For cubic splines, this would result in a quite large interval approximated by a single cubic polynomial, which could be too crude for valuable results.

The PI programs are written in FORTRAN 90, and compiled with a Sun Studio 10 Compiler using the "-fast -O5" optimization to be run in the background on an 6×1.2 GHz UltraSPARC-IV computer. 120 iterations took 9 hours for this toy example.

It is very clear from the authors' experience that both CPU time, computer memory and the complexity of the computer programming has made an extension to even more dimensions almost unthinkable. However, a higher grid resolution in six dimensions should be possible with a more powerful computer.

A six-dimensional PDF cannot be represented with ease, neither on paper nor on a computer screen. So one is practically forced to plot marginal densities or conditional densities, or to refer scalar statistics of the final distribution. As stated in the introduction, many such statistics would be more easily obtained using Monte-Carlo simulations. So the presentation of resulting response PDFs will here mainly be based on some conditional densities.

Three identical independent oscillators are modelled as a six-dimensional SDE based

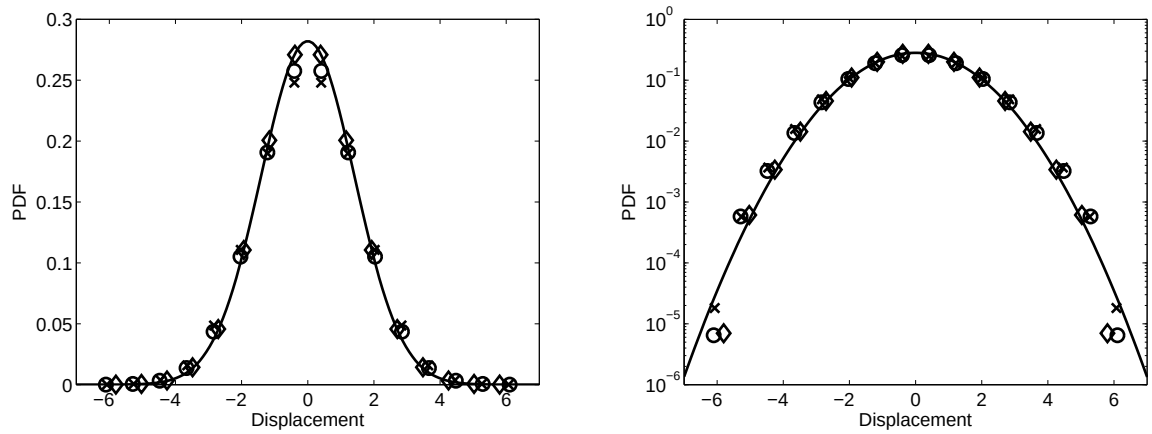


Figure 5.2: Displacement PDF of the linear oscillator (line) with the numerically obtained response PDFs for the three oscillators that were computed as a joint 6D system by PI. The three oscillators are marked with symbols x, circle, and diamond, respectively. The diamond corresponds to the forcing filter. The left plot is in a normal scale, the right plot has a logarithmic scale on the second axis.

on equation (1.23) with three independent Gaussian white noise processes W_1, W_2, and W_3;

$$\begin{aligned} \mathrm{d}X_j &= Y_j \,\mathrm{d}t, \\ \mathrm{d}Y_j &= (-\alpha X_j - 2\kappa Y_j)\,\mathrm{d}t + \gamma \,\mathrm{d}W_j, \quad \alpha, \kappa > 0, \quad j = 1, \ldots, 3. \end{aligned}$$

The parameter values used are $\alpha = 0.01$, $\kappa = 0.5$, and $\gamma = 0.2$.

This should from the definition of independent variables and equation (1.24) give a joint PDF

$$f(x_1, x_2, x_3, y_1, y_2, y_3) = K^3 \exp\left(\frac{2\kappa}{\gamma^2} \sum_{j=1}^{3} \alpha x_j^2 + y_j^2\right).$$

The time step for the numerical simulation was $\Delta t = 0.03$. The fact that the three oscillators were independent or that they were all linear and identical was not used to optimize the 6D PI code. The code was written to handle two coupled nonlinear oscillators driven by a filtered noise process. In the grid, the six variables are not completely identically represented. That is, loops for tensor product interpolation and time stepping are performed in a specific sequence that could affect the numerical stability of the variables differently. In addition, the grid in each variable was set using limits of 3.6 times the standard deviation of the variable in a short Monte-Carlo simulation, which means that the grid points are not identically located in each direction.

Figure 5.2 shows the numerically obtained response PDFs for displacement for the three oscillators, together with the analytical solution computed from equation (1.24) by integrating the y variable from $-\infty$ to ∞. Similarly, Figure 5.3 shows the velocity PDFs of the three oscillators obtained by PI together with the analytical PDF in the y variable. The

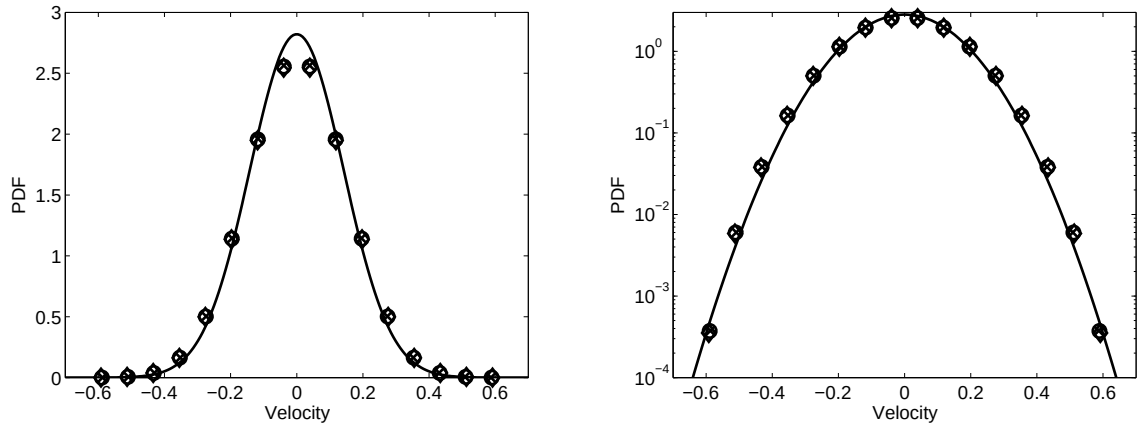


Figure 5.3: Velocity PDF of the linear oscillator (line) with the numerically obtained response PDFs for the three oscillators that were computed as a joint 6D system by PI. The symbols correspond to the previous figure. The left and right plot use linear and logarithmic scale, respectively.

peaks and the grid values at the boundaries are under-estimated, an effect that is typical for this implementation of PI with cubic or rectangular B-splines and the assumption that the PDF is zero outside of the domain. With the very low grid resolution, the results are promising. The fact that there are only small deviations between the PDFs for the three oscillators and that all PDFs are symmetric down to machine precision, indicate that the code does not have serious indexing mistakes. With the limitations of the grid resolution, a more detailed verification of the code accuracy is difficult, and extensions to coupled or non-linear systems did not seem to be feasable.

5.12 Example: Locally adjusted grid for a Toda oscillator

Also smooth systems might require an adapted grid to work well with PI. An example is a chaotic system described in [27]. It is developed from a nonlinear driven RLC circuit, where the variable is the electric charge on the capacitor. This system is also studied experimentically, verifying that period doubling and chaotic behaviour exist when the system is parametrically tuned. After normalization, the system is described in dimensionless form as

$$\ddot{x} + \kappa\dot{x} + (e^x - 1) = \alpha\cos(\omega t). \tag{5.14}$$

To study equation (5.14) as an SDE with PI, a small noise term is added to the right hand side, and the system rewritten as

$$\begin{aligned} \mathrm{d}X_t &= Y_t\,\mathrm{d}t, \\ \mathrm{d}Y_t &= (1 - \exp(X_t) - \kappa Y_t + \alpha\cos(\omega t))\,\mathrm{d}t + \gamma\,\mathrm{d}W_t. \end{aligned} \tag{5.15}$$

For this numerical calculation, the parameters used are $\kappa = \sqrt{0.0018}$, $\alpha = 10/3$, $\omega = 0.905$, and $\gamma = 0.005$, and there were 80 time steps per period.

Before trying to implement such a system in PI, one should try to get a picture of the dynamics of the system. For equation (5.15), Figure 5.4 shows plots of the response in the variables X_t and Y_t, respectively, from an arbitrary initial point $(X_0, Y_0) = (0.0, 0.3)$, a specific realization of the noise process, and time running from $t = 0$ to 520. The realization is obtained by a standard Monte Carlo simulation. Figure 5.4 indicates that the variability in the local maxima are much smaller than that of the local minima, and that the grid clearly should not be chosen symmetric around zero in the x direction. Figure 5.5 shows the same transient, this time plotting X_t against Y_t. The trivial observation is that the density is very high in a small area where $0 < x < 4$. More important is to observe that this area exhibit some structure shown in the zoomed image that one should expect is important for the global structure of the system. Further, the trajectories bend quite sharply from almost horizontal to vertical in this region, which also indicates that one should be especially aware of the dynamics here.

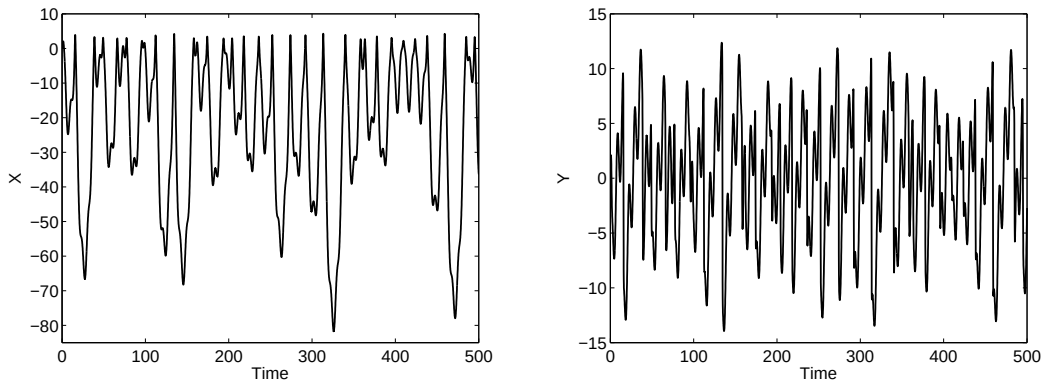


Figure 5.4: Response in the X and Y variables of the Toda oscillator, left and right plots, respectively. The plots are based on one Monte Carlo simulation.

In this example, PI fails to give a reasonable result with 101 grid points in X and Y with a uniform grid, and the mass after each iteration is only between 0.70 and 0.85. Instead of running this system with a very high resolution, the grid in the x direction should have high resolution in the region above zero. Figure 5.6 shows the PDF of system (5.15) at two time instances. These plots are split in two, one part for the PDF for x below zero and one for x above zero. Both use the same grid in the y direction, and the same number of grid points in x. In the PI implementation, there is only one grid, where 0 is grid point number 50 out of 101, and the grid is uniform on each side of 0. This choice makes it simple to compute probability mass and moments and other expectation values. The interpolation generally would be more stable if the grid density changed more gradually. Note that even though the grid is piecewise uniform, the basis functions for a uniform grid like equation (3.6) and (3.7) are not valid - in any interval.

In [27], this system is studied for a range of parameter values for the frequency and

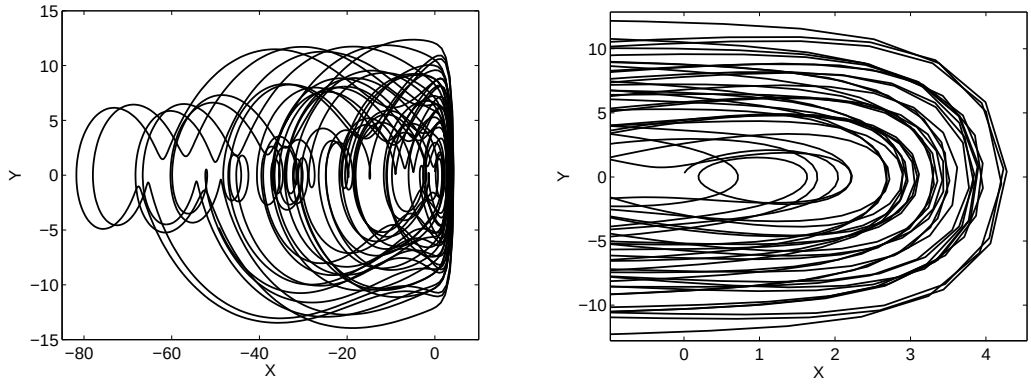


Figure 5.5: Response in the X and Y variables of the Toda oscillator, plotted against each other. The right plot is a zoom-in on the rightmost part of the left plot.

amplitude of the harmonic forcing. In this parameter space, many subharmonic periods of the response are found, including chaotic regions. A visual inspection of the response PDF for this system for the same parameter values does not give the same insight, as the PDF does not show how the dynamics affects individual points. Also for a chaotic system, the PDF can be stationary or periodic. The PDF for equation (5.15) is periodic with the same period as the forcing. However, a detailed study of the PDF over one period can indicate the subharmonic period of the system. Figure 5.7 shows one period for four different simulations, all of them sampled at 16 equi-distant time steps. When studying the "finger" shapes on the left of the PDF, one can see that they move from one snapshot to the next, so that the bottom "finger" takes the position of the second lowest after one period. Since the number of "fingers" are different for the different cases, the rotation speed of the PDF can be measured, which for the first to cases is $2\pi/(5\omega)$, and for the two last $2\pi/(6\omega)$. The deterministic oscillator has a very complicated parameter space distribution of different stable periodic solutions and chaotic behaviour [27]. Here it is just a toy example, so the behaviour for varying parameters is not studied further.

One important comment here, is that knowledge of symmetries and periods in the response PDF of a system can be useful to obtain faster convergence rates or more stable results. An example of this is studied in detail in appendix F.

The next chapter will discuss chaos, and the study of some chaotic systems.

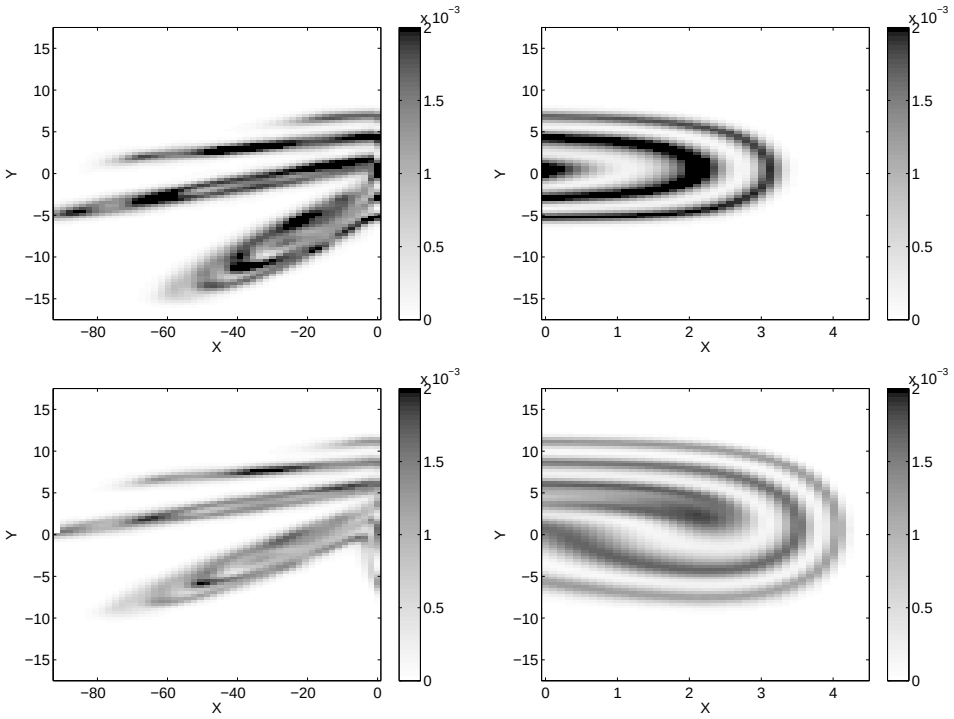
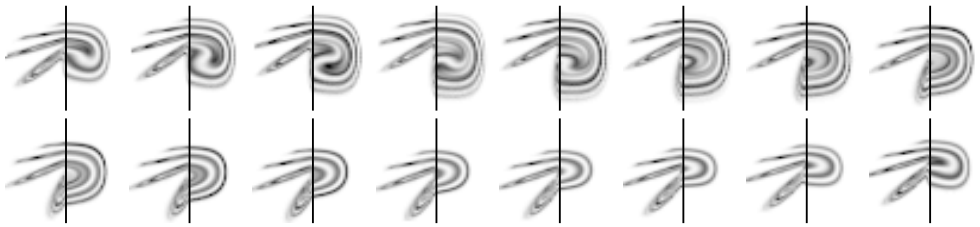
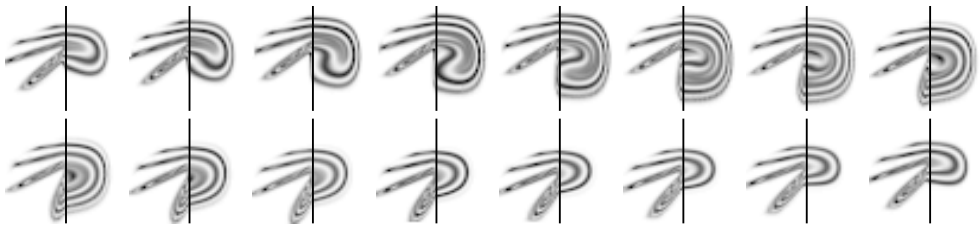


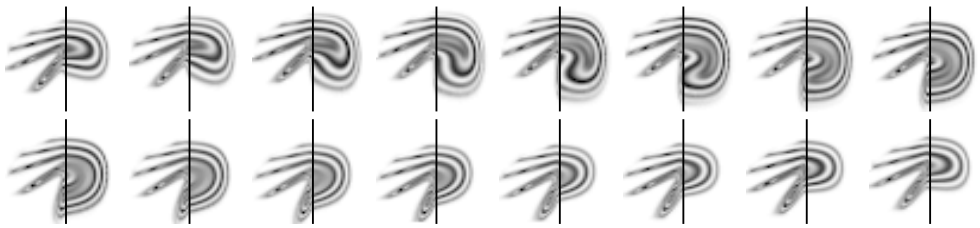
Figure 5.6: Four plots showing the response PDF of the Toda oscillator in x and y at two different time instances. The top is obtained after ten full periods, the bottom a quarter period later. Note that the resolution in the horizontal direction is very different in the left and right images, while the colourbar and the y axis is the same in all the four plots.



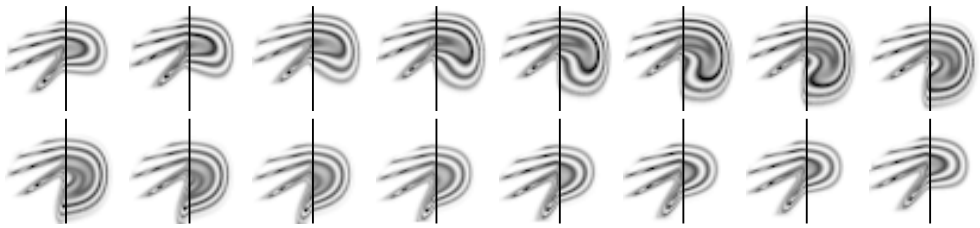
A: $\omega = 0.905$.



B: $\omega = 1.205$.



C: $\omega = 1.505$.



D: $\omega = 1.805$.

Figure 5.7: One period of the response PDF of the oscillator from equation (5.15) for four different values of ω. The vertical lines indicate $x = 0$, and are included because the grid is much denser for positive x than for negative x, similarly to the split PDFs in the previous figure.

Chapter 6

Chaos

6.1 Chaotic and stochastic response of nonlinear SDEs

The study of nonlinear SDEs with noise and chaos could be persued in many directions. One motivation could be to examine the properties of the underlying deterministic system, where the noise acts as a perturbation to study the system's stability, or as a means to apply stochastic methods to the system. In that case, the question is how results from studies of the stochastic system relates to the deterministic system, and to what extent these systems are comparable. Another related question is whether the introduction of noise terms to the deterministic system could lead to a radical change of the system's behaviour, i.e. introduction or removal of chaotic response or completely new response patterns.

Investigations of deterministic systems with chaotic response has showed that tuning of parameters often leads to bifurcations and sudden appearance of chaos. Similar behaviour is seen for stochastic systems, though the noise to some extent could be said to smear out the abrupt changes.

The comparisons to the stochastic system can be thought of as throwing a "stochastic blanket" over the deterministic attractor, so that the noise level potentially smears out some of the finest structure. However, this requires that no abrupt changes of the system behaviour occurs in the introduction of noise.

A general discussion of the effect of noise on chaotic systems is difficult, since the number of possible systems, noise processes and outcomes is infinite. This is illustrated in sections 6.17 and 6.18, which follow up on the general comments on effects of noise in section 1.3.9. As for the PI method in general, it is much more illustrative to focus the analysis on some specific systems. Note that the objective is to indicate the possibilities the PI method provides for such a study, rather than actually analysing the example systems in detail.

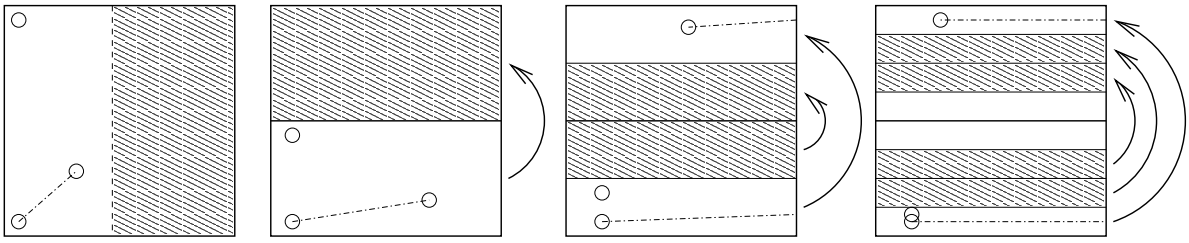


Figure 6.1: Illustration of a few iterations of the Smale horsehoe map, from left to right. In every iteration the right half of the square is cut off and folded over, as the arrows indicate. The two pieces are streached horisontally to twice the length, and shrinked horisontally to half the width, which directly relates to the Lyapunov exponents, $\ln 2$ and $\ln(1/2) = -\ln 2$. The illustration also shows three points and a line connecting two of them, and how these objects are propagated by the map.

6.2 The definition of chaos

First let a deterministic dynamical system define a diffeomorfism on the state space $\phi(\mathbf{x}, t) : \mathcal{X} \times \mathbb{R} \to \mathcal{X}$ for every $t \geq 0$, where $\phi(\cdot, 0)$ is the identity operator. That is, a deterministic version of the ϕ defined in section 1.3.4.

The dynamical system is said to be chaotic if the collection of points with periodic orbits is dense, it is topologically transitive, and sensitive to perturbations of the initial condition. Points with periodic orbits fulfills $\mathbf{x} = \phi(\mathbf{x}, n\tau)$ for all $n \in \mathcal{N}$ for some value τ, which will depend on $\mathbf{x}$. The chaotic attractor for the system is the closure of the dense set of points with periodic orbits, and the measure of the set of periodic points is zero.

The criterion "topologically transitive" means that if for any two intervals U and V on the system's attractor, there exist a t such that $\phi(U, t) \cap V \neq \emptyset$. This means both that the attractor cannot be separated into disjoint parts that never meets, and roughly that every starting interval will "fill up" the attractor when evolved for a long time by ϕ. This is equivalent to the term mixing. To illustrate this point, consider the so-called Smale's Horseshoe Map. This is a map that streches a rectangle horisontally, then breaks and folds it (or bends it into a horse-shoe shape, hence the name). This is illustrated by the first few iterations in Figure 6.1. The shaded region occupies the same area after every iteration, however, its gradually more complicated structure will seem to fill the square. In other words, the maximal distance from a point inside the square to the shaded region will constantly decrease.

Finally, sensitivity to perturbations is really a consequence of the two first criteria, which is proven in [2]. This is maybe the most practical way of determining if a physical system could be chaotic, and is well known as the butterfly effect.

In popularizations of chaos, e.g. [15], a typical conclusion is that chaotic systems are sensitive to perturbations always and everywhere, hence they cannot be controlled. Otherwise, one should be able to perturb the system at the right time and place by an infinitesimal amount – almost without effort – and this way ensure a specific outcome much later. A tantilizing idea, however not possible in practice. One should clearify that the statement

that this is impossible is a practical observation, especially observed for weather models, but not a mathematical condition. For constructed systems and some mechanical models, chaos can be controlled or reduced by artificial damping or controlled perturbations in some parts of the state space. Such control requires some specific knowledge of the system's behaviour in both space and time, which has been difficult to obtain by sampling from specific realizations.

Transitions from periodic to chaotic behaviour, through period doubling and other phenomena, is not discussed in this text. To investigate such behaviour with PI, one would choose some parameter values, study the system with these parameters, and compare the results. A simpler way to compare a system's behaviour under many values of a parameter has not been found.

6.3 Qualitative study of chaotic attractors by PI

6.3.1 Motivation

A few papers have been published the last years with studies of nonlinear or chaotic systems under the influence of noise – or other kinds of perturbations, and a variety of phenomena has been discovered. There are however many ways noise can be introduced, and very little has been done to look at differences between the response under various noise models.

The aim of this section is to show some of the possibilities of using path integration for systems where the noise is filtered, increasing the dimensionality of the problem to four, when a state-space variable is an angle. Although the systems considered have some very basic properties in common, the notion of convergence and how the attractor finally appears are very different for the different cases. This paper also extends Âprevious work [39, 42] where a single-degree of freedom ODE with chaotic response is studied by the addition of noise using path integration.

The Duffing oscillator has been studied extensively for many years because of its interesting dynamics combined with its simplicity, and it is still recognized that the Duffing class of oscillators plays a central role among harmonically excitated systems that exhibit chaotic behaviour [18]. Parameter ranges for different structure levels have been mapped quite detailed, e.g. in [14], and the Duffing model is also used to model more complicated structures of coupled nonlinear oscillators [38]. As the deterministic single-degree-of-freedom oscillator is more understood, the interest is turned to stochastic models and higher-dimensional problems.

To the authors' knowledge, no other methods have been used to produce full PDFs of four-dimensional nonlinear systems with this complexity, at least by the time this work was performed in 2003.

6.4 System description

A nonlinear single-degree-of-freedom (SDOF) system subject to periodic excitation is the basis for this analysis. This system is a Duffing oscillator where the response process $X(t)$ satisfies the following equation

$$\ddot{X}_t + \frac{1}{25}\dot{X}_t - \frac{1}{5}X_t + \frac{8}{15}X_t^3 = F(t), \tag{6.1}$$

where $F(t)$ is the stochastic excitation, with some specific properties. The motivation for this study is the corresponding deterministic ODE with $Y(t) = (2/5)\cos(\omega t)$, which has been studied extensively in [49] for various frequencies ω.

This system is separated into displacement X and velocity Y as

$$\dot{X}_t = Y_t \tag{6.2}$$

$$\dot{Y}_t = -\frac{1}{25}Y_t + \frac{1}{5}X_t - \frac{8}{15}X_t^3 + F(t) \tag{6.3}$$

The four kinds of excitation studied here, denoted A through D are

$$F_{\mathrm{A}}(t) = \frac{2}{5}\cos(\omega t) + \gamma_{\mathrm{A}} N_t, \tag{6.4}$$

$$F_{\mathrm{B}}(t) = \frac{2}{5}\cos(\omega t + \Phi_t),\ \dot{\Phi}_t = -\kappa_{\mathrm{B}}\Phi_t + \gamma_{\mathrm{B}} N_t, \tag{6.5}$$

$$F_{\mathrm{C}}(t) = \frac{2}{5}\cos(\Theta_t),\ \dot{\Theta}_t = \omega + \gamma_{\mathrm{C}} N_t, \tag{6.6}$$

$$F_{\mathrm{D}}(t) = Z_t,\ \ddot{Z}_t + \kappa_{\mathrm{D}}\omega\dot{Z}_t + \omega^2 Z_t = \gamma_{\mathrm{D}} N_t. \tag{6.7}$$

Here, N_t is the standard zero-mean white noise process or Gaussian white noise, also described as the formal derivative of a standard one-dimensional Brownian motion. This process has uncorrelated increments $\mathrm{E}[N_t N_{t+\tau}] = \delta(\tau)$, where $\delta(\cdot)$ is a Dirac measure at zero. The noise levels γ_{A}, γ_{B}, γ_{C}, and γ_{D} will be chosen so that the mean input energy is the same for all the four cases, according to the choice of the other parameters.

As in equation 6.2, the filter in case D will be split into two first-order SDEs by introducing the filter velocity $\Upsilon = \dot{Z}$.

From here on, $\mathbf{X}$ will be the vector of state space variables, which for the four cases read

$$\mathbf{X}_{\mathrm{A}} = [X, Y]^\top \in \mathbb{R}^2 \tag{6.8}$$

$$\mathbf{X}_{\mathrm{B}} = [X, Y, \Phi]^\top \in \mathbb{R}^3 \tag{6.9}$$

$$\mathbf{X}_{\mathrm{C}} = [X, Y, \Theta]^\top \in \mathbb{R}^2 \times \mathbb{S} \tag{6.10}$$

$$\mathbf{X}_{\mathrm{D}} = [X, Y, Z, \Upsilon]^\top \in \mathbb{R}^4. \tag{6.11}$$

The differences between the two three-dimensional systems B and C require a comment. Both are models where there is a perturbation in the frequency of the forcing, while

preserving the mean angular velocity around ω. In case B, the system is almost locked to a fixed phase at every time t, perturbed by Φ. Although Φ is strictly an angle, the noise variance is set small enough that the perturbation is much less than 2π. By this, it is reasonable to model this variable and the Brownian motion as a one-dimensional process on the real line. This also ensures that the forcing cannot stay a full period behind or ahead of the pre-set phase. In a realization of the system, any random delay compared to that phase at a specific time would make the damping effect try to "speed up" the angle, and the more the angle is ahead of the phase due to the random fluctuations, the larger the probability is for dominating lower frequencies. In case C, the filter does not inherit this kind of memory. The angle can be seen as a standard zero-mean Brownian motion on a circle superimposed on a fixed rotation, or a Brownian motion on the circle with a mean angular velocity, or say drift, ω. As the variance of the Brownian motion on the real line is increasing with time, the distribution on the circle converges towards a uniform distribution, which means that the phase for system C after a long time is impossible to predict with any accuracy, in contrast to system B. However, the value of Θ is the current phase value.

The spectrum of F_{C} in equation (6.6) is, according to [54],

$$S(\omega) = \frac{1}{2}\frac{(2/5)^2\gamma_{\mathrm{D}}^2(\omega^2+\omega^2+\gamma_{\mathrm{D}}^4/4)}{(\omega^2-\omega^2+\gamma_{\mathrm{D}}^4/4)^2+\omega^2\gamma_{\mathrm{D}}^4}. \tag{6.12}$$

This also shows the fairly obvious result that $\gamma_{\mathrm{D}} \to 0$ gives harmonic excitation.

The fourth filter, D, is an oscillator that also gives a mean frequency ω. As in case C, the phase will be distributed over the joint PDF, but can be retained from the values of the two last state variables. Here, the amplitude will also be heavily affected by the noise. The output of a linear filter with Gaussian noise is Gaussian, so the forcing spectrum is clearly different from the sinusoidal in the three first cases. To find a relevant comparizon, one could look at the resulting PDF conditioned on values of Z and Υ that corresponds to the harmonic forcing, which is an ellipse in the Z, Υ-space centered in the origin.

It is known that the purely deterministic system with harmonic forcing, $\gamma_{\mathrm{A}} = 0$ in equation (6.4), gives rise to a chaotic attractor. With all the noise models, it is expected that the response should be similar to this attractor, more so for noise with low intensity. As the noise makes a probability density, i.e. a surface, some of the fine structure of the chaotic attractor must dissappear, like a "stochastic blanket" is thrown over the attractor.

To be a relevant comparison, approximately the same amplitude should be delivered from the stochastic forcing models. This is clearly a question of definition, e.g. would the addition of zero-mean noise in case A give an addition in energy and mean amplitude, but a time average around the peaks would give the correct amplitude. For cases B and C, the amplitude is fixed, but the frequency variation would give a slight increase in energy to the system. In the last case D, a choice has to be made, as the force is just quasi-harmonic without a maximum amplitude. Here, the variance was chosen as the reference. The variance in Z of the system D is from equation (6.7) $\gamma_{\mathrm{D}}^2/(2\omega^3\kappa_{\mathrm{D}})$. The variance of the deterministic forcing $(2/5)cos(\omega t)$ is $2/25$. If these two are set equal to each other, a given damping κ_{D} implies a noise level $\gamma_{\mathrm{D}}^2 = 4\omega^3\kappa_{\mathrm{D}}/25$. This is equivalent to scaling the

forcing in equation (6.2). As the damping and noise intensity goes to zero, the response Z becomes more and more harmonic, as shown in Figure 6.2.

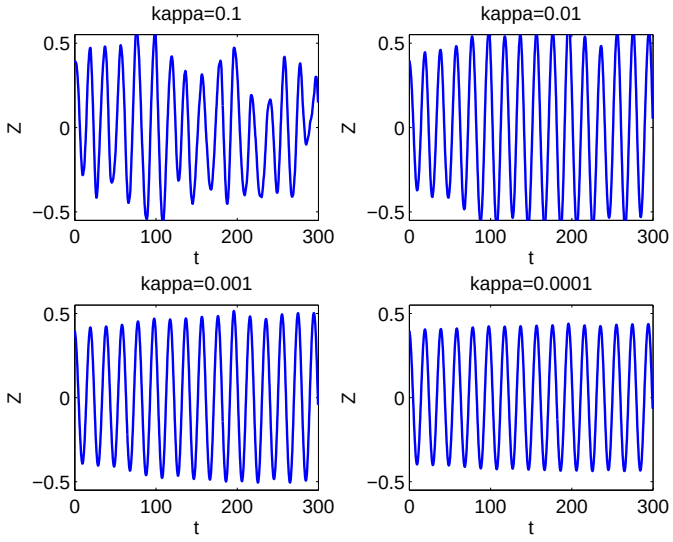


Figure 6.2: Four realizations of the two-dimensional filter with damping κ_{D} varying from 0.1 down to 0.0001, and noise level chosen correspondingly.

The concept of convergence for SDEs driven by harmonic excitation has been discussed in [42]. For the two first excitation models A and B, the response is directly depending on the value of the time t. Since the damping is positive, it can be assumed that the density becomes periodic after transients have died out, and the period will be $2\pi/\omega$. Existence and uniqueness of such a periodic attractor for the response PDF is difficult to prove for a nonlinear system. However, if a PDF $f^{\dagger}$ at time t_0 is found to be periodic with period $2\pi n/\omega$, then the density $f^{\ddagger}$ obtained at time $t_1 = t_2 + 2\pi/\omega$ also has the same period. The average over all the obtained PDFs at $t_1, t_2, \ldots, t_n$, $t_j = t_1 + 2\pi j/\omega$ must then have period $2\pi/\omega$.

For the third and fourth cases, there is no explicit time dependence; the system is autonomous, and a stationary distribution could exist.

6.5 Implementation: conditional PI with cubic splines

For case B, the stationary density of the Φ_t process is Gaussian with density $f(\phi) = K\exp(-\kappa_{\mathrm{B}}x^2/\gamma_{\mathrm{B}}^2)$, i.e. zero mean and variance $\gamma_{\mathrm{B}}^2/2\kappa_{\mathrm{B}}$. ($K$ is just the constant scaling.) The marginal PDF of Φ from X, Y, Φ must also be the same Gaussian distribution, as the three-dimensional system only has introduced "auxilliary" variables X and Y and then they are integrated out for the marginal. In case D, the joint stationary - and marginal - PDF of Z, Υ is bivariate Gaussian with density $f(z, w) = C\exp(-\kappa_{\mathrm{D}}\omega/\gamma_{\mathrm{D}}^2[\omega^2 z^2 + w^2])$.

That is, zero-mean, zero covariance, and variances $\gamma_{\mathrm{D}}^2/(2\omega^3\kappa_{\mathrm{D}})$ and $\gamma_{\mathrm{D}}^2/(2\omega\kappa_{\mathrm{D}})$ for the response of Z and Υ respectively.

The question is whether this knowledge could be utilized to reduce both the grid resolution and the interpolation cost. A proposed solution is to consider the conditional densities $f(x, y|\phi)$ and $f(x, y|z, \upsilon)$ for these two cases respectively, and just interpolate linearly in the conditional variables. This will work well if the conditional density $f(\phi|x, y)$ for fixed x, y has a sufficiently similar shape to $f(\phi)$ in case B, and similarly for case D.

Let us go through the procedure for case B in detail. Assume that the PDF $f_{j\Delta t}(\mathbf{x})$ is already calculated in every grid point $\mathbf{x}_{k_1,k_2,k_3}$, and denote this for short by $f_j(\mathbf{x})$. Write $f_j(x_{k_1}, y_{k_2}|\phi_{k_3}) = f_j(\mathbf{x}_{k_1,k_2,k_3})/f(\phi_{k_3})$ for every grid point. For each fixed k_3, represent f_j by the cubic spline interpolant $\widehat{f}_{k_3}(x, y) = (I_{4,xy}f_j)(x, y|\phi_{k_3})$. For each point in the integration domain, the density is evaluated as $f_j(x, y, \phi) = \Big[(1 - \lambda_{k_3})\widehat{f}_{k_3}(x, y) + \lambda_{k_3}\widehat{f}_{k_3+1}(x, y)\Big]\, f(\phi)$, where $\lambda_{k_3} = \lambda_{k_3}(\phi) = (\phi - \phi_{k_3})/(\phi_{k_3+1} - \phi_{k_3})$, and k_3 is chosen such that $\phi_{k_3} \le \phi < \phi_{k_3+1}$.

In case D, the method is the same, except more indices, and the final linear interpolation becomes

$$\begin{aligned} f_i(x, y, z, \upsilon) = \big[&(1 - \lambda_{k_3})(1 - \lambda_{k_4})\widehat{f}_{k_3,k_4}(x, y) + \lambda_{k_3}(1 - \lambda_{k_4})\widehat{f}_{k_3+1,k_4}(x, y) \\ &+(1 - \lambda_{k_3})\lambda_{k_4}\widehat{f}_{k_3,k_4+1}(x, y) + \lambda_{k_3}\lambda_{k_4}\widehat{f}_{k_3+1,k_4+1}(x, y)\big] f(z, \upsilon). \end{aligned} \tag{6.13}$$

A challenge with this four-dimensional system is that the filter has a very long transient time for low noise levels. The main attempt to reduce this is by starting with the correct marginal density in Z and $\dot{Z}$.

A final remark on conditional PI is that the procedure could also work for systems where the marginal density of the filter variables is unknown or not stationary, as this PDF can also be found for each time step with fair accuracy using the PI method.

6.6 Numerical solution for response PDF by the PI method

A Poincaré section of the chaotic attractor obtained for the deterministic system, case A with $\gamma_{\mathrm{A}} = 0$ in equation (6.4) is shown in Figure 6.3. This is a well studied example, using $\omega = 0.32$, and this choice has been made for all the simulations with results shown in this paper. The sampling time is $2\pi k/\omega$, $k = 1, 2, \ldots$, i.e. after every full period of the harmonic forcing.

Snapshots of the PDF of the additive noise model A, equation (6.4), after 30 periods are shown in Figures 6.4 and 6.6, for experiments with varying amount of noise. As expected, the higher noise levels give less fine structure and more spreading of the PDF. There is hardly any difference between the estimated response for the two lowest noise levels. This could indicate that for these results, at least for the lowest noise level, the interpolation is the main source of smearing out the attractor, and the limit of structure detail obtainable has been reached. It is also interesting to see that the strongest peak at the upper left tip

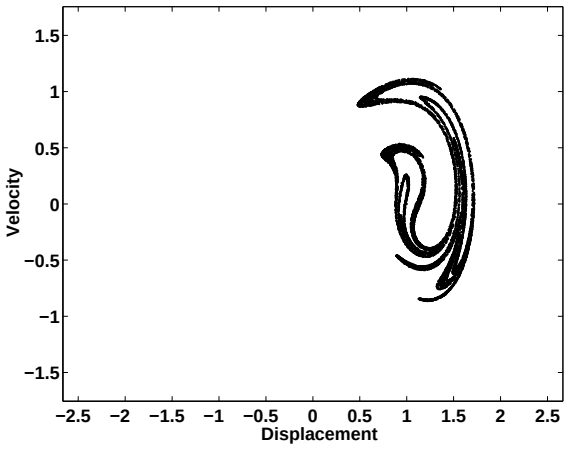


Figure 6.3: Poincaret section of the chaotic attractor.

of the structure gradually becomes lower with increasing noise, while the vertical structure around $x = 1.5$ remains strong. Similarly, one of two peaks near the center of the figures gradually dissappear with increasing noise, while the other remains a maxima.

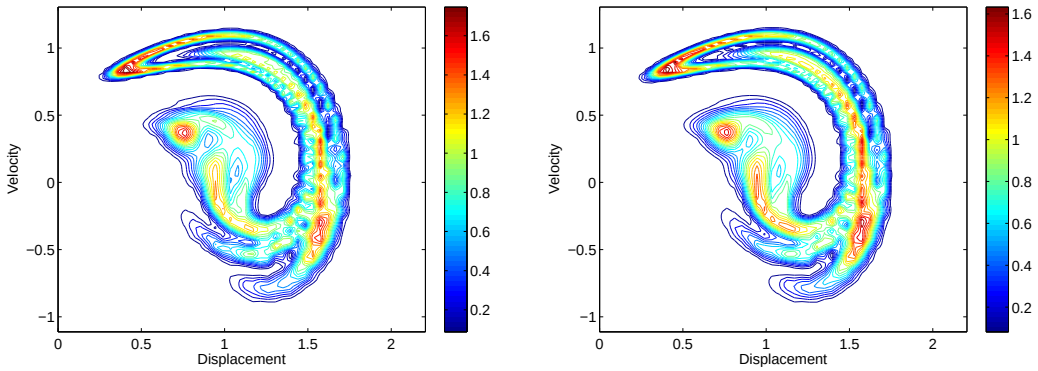


Figure 6.4: PDF of the response of the 2D system, additive noise, at time $60\pi/\omega$, i.e. 30 full periods from the initial distribution. The noise level is $\gamma_A = 0.005$ and $\gamma_A = 0.010$ in the left and right image, respectively.

Similar snapshots are shown for system B in Figures 6.7 and 6.8. In spite of the fact that the phase is varying and the amplitude is constant, while case A is opposite, the increasing noise seem to have the same effect on the system.

For the filtered noise version, case D, the results are shown in Figure 6.11.Even for the low damping, and hence low noise, one cannot see the same level of detail as in the previous results. This indicates that some averaging process is dominant for this system much more than for the previous cases. The location and size of the maxima is similar to what is seen for high noise levels in the other models. When studying the left and center images in Figure 6.11, it is quite surprising that the PDF doesn't seem to change when

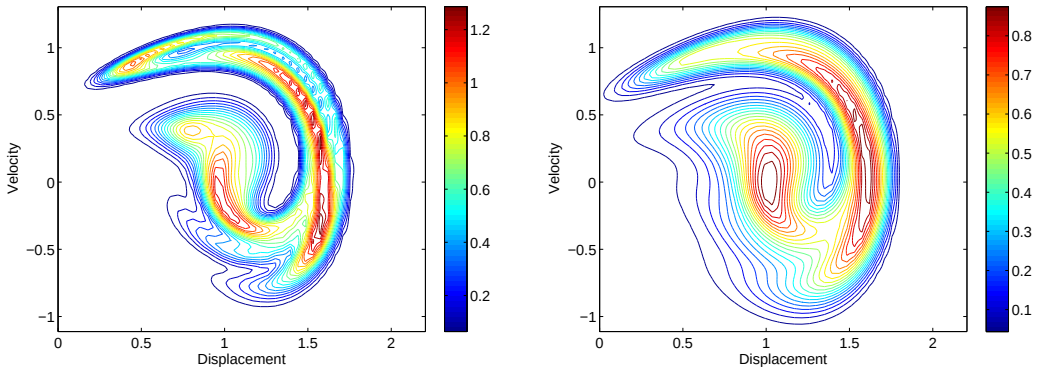


Figure 6.5: PDF of the response of the 2D system, additive noise, at time $60\pi/\omega$, i.e. 30 full periods from the initial distribution. The noise level is $\gamma_A = 0.025$ and $\gamma_A = 0.050$ in the left and right image, respectively.

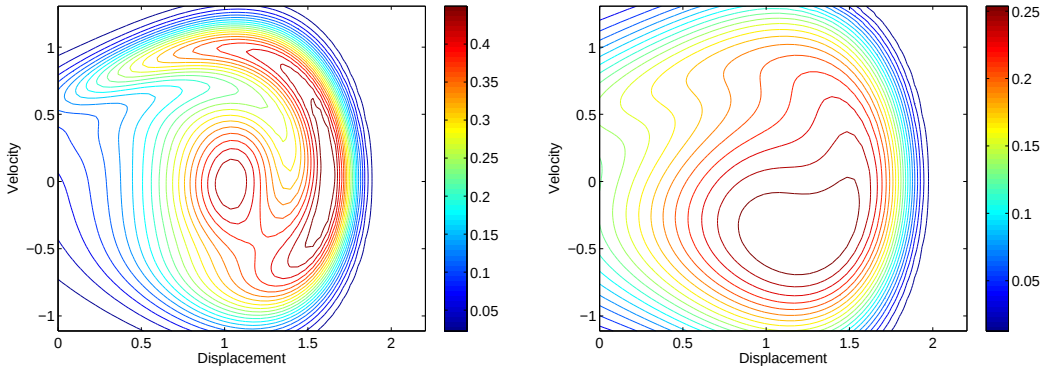


Figure 6.6: PDF of the response of the 2D system, additive noise, at time $60\pi/\omega$, i.e. 30 full periods from the initial distribution. The noise level is $\gamma_A = 0.100$ and $\gamma_A = 0.250$ in the left and right image, respectively.

the damping is increased from 10^{-4} to 10^{-1}, as the realizations of the forcing shown in Figure 6.2 are so different. Simulations were performed with a grid of $81 \times 81 \times 44 \times 45$ grid points over the state space. In the z-direction, the grid was from -0.8 to 0.8, i.e. twice the value of interest, 0.4. The grid for the w-direction was chosen correspondingly based on the maximal velocity (0.4ω). As discussed above, the deterministic harmonic forcing corresponds to an ellipse in the z, w-space. With the choice of computational grid, this ellipse crosses 22 gridlines in each direction - twice, so the interpolation was expected to be fairly good. It is possible that interpolation in four dimensions smears out the structure much more than in the previous cases, also since just two points on the ellipse actually intersects grid points. Also note that the separation between the two peaks is clearly decreased as the noise increases, while the main shape is preserved. For the case with highest damping and noise, the right image in Figure 6.11, very little shape is preserved,

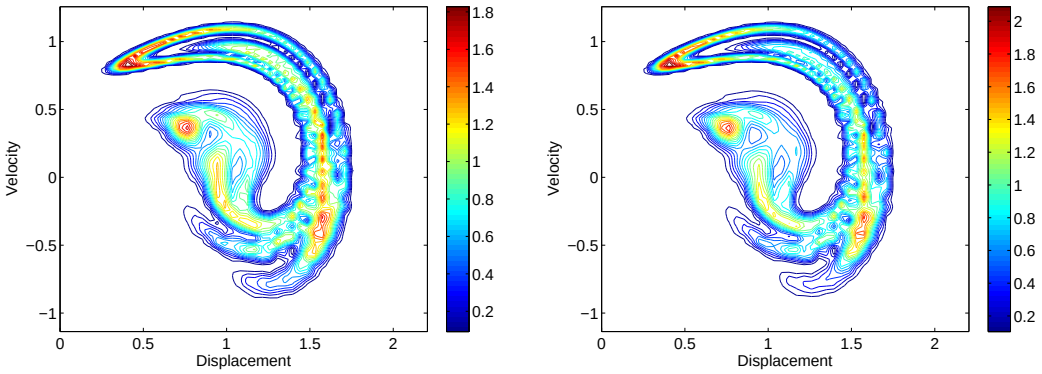


Figure 6.7: PDF of the response of the first 3D system, damped noise in the phase, at time $60\pi/\omega$, i.e. 30 full periods from the initial distribution. The noise level is $\gamma_B = 0.005$ and $\gamma_B = 0.010$ in the left and right image, respectively.

the maxima is almost circular, although the total variation is about the same as the bottom image in Figure 6.10 and the right image in Figure 6.6, and not much larger than in the bottom image in Figure 6.8, all of which has a clearly non-circular maximum. However, the added noise model A is much more similar to the 4D case D than the two 3D models B and C. This clearly indicates that the filtered noise model and the added noise model affect the oscillator in a very different way than the phase perturbation for high noise values.

6.7 The Lyapunov exponents – based on the flow of the system

The term Lyapunov exponents (LEs) is used both for local Lyapunov exponents (LLEs) and global LEs. Unless clearly stated otherwise, LEs here only refers to the global kind. This is further a finite set of numbers, i.e. a discrete spectrum, with size equal to the number of state-space variables (i.e. the dimension), and sometimes plus one for the time variable. An important statistic for dynamical systems is the maximal Lyapunov exponent (MLE), also denoted the largest LE (LLE), or the Lyapunov characteristic exponent (LCE). The MLE is said to be the averaged rate of divergence (or convergence) of two trajectories that are close to each other. As for chaos, the LE is defined for deterministic systems, and then extended to stochastic systems. This extension will be discussed below, but first assume that the system considered is deterministic.

A system of ODEs determines a flow; $\mathbf{x}_t = \varphi(\mathbf{x}_0, t)\mathbf{x}_0$, when written formally as a (non-linear) matrix-vector product. The LE can then be defined through the eigenvalues $m_j(t)$, $j = 1, \ldots, n$, $m_1(t) \geq m_2(t) \geq \ldots \geq m_n(t)$ of the flow matrix $\varphi(\cdot, t)$;

$$\lambda_j = \lim_{t\to\infty} \frac{1}{t} \ln |m_j(t)|. \tag{6.14}$$

Often in the litterature, it is stated that to estimate the Lyapunov exponent, one should

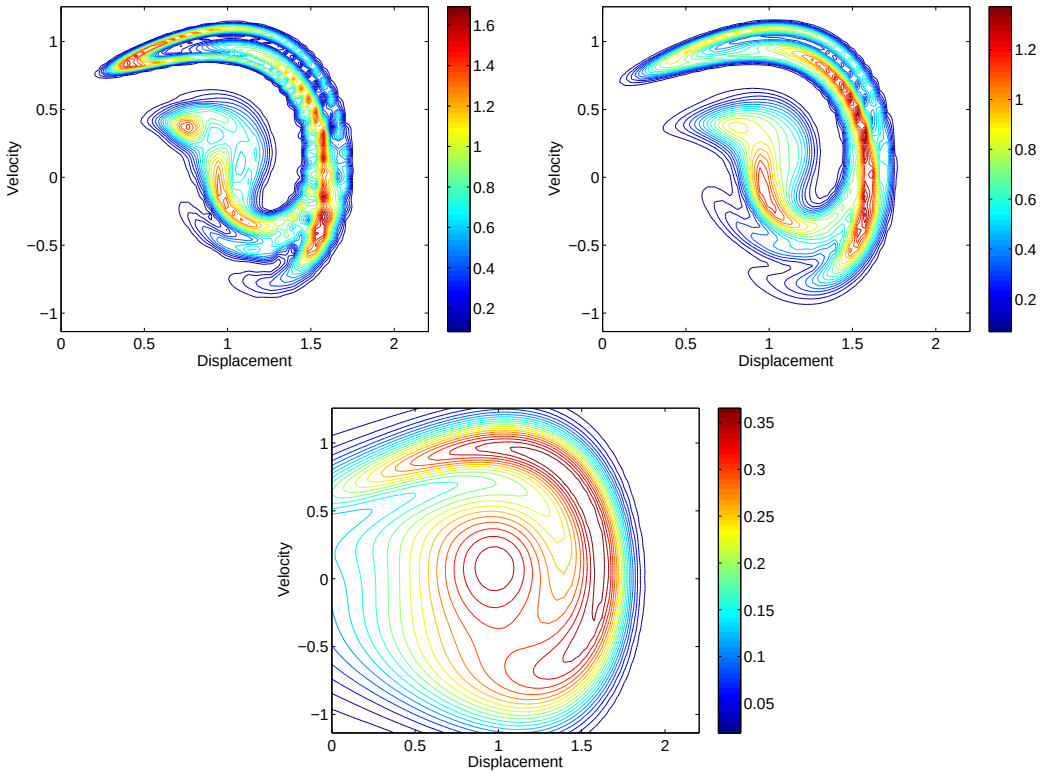


Figure 6.8: PDF of the response of the first 3D system, damped noise in the phase, at time $60\pi/\omega$, i.e. 30 full periods from the initial distribution. The left, right and bottom images have $\gamma_B = 0.050$, $\gamma_B = 0.100$, and $\gamma_B = 0.500$, respectively.

start with two points $\mathbf{x}_0$ and $\widetilde{\mathbf{x}}_0 = \mathbf{x}_0 + \Delta\mathbf{x}$ that are very close, follow their paths forward in time while measuring the distance between them. The definition should then be written

$$\lambda_1 = \lim_{t\to\infty, \Delta\mathbf{x}\to\mathbf{0}} \frac{1}{t} \ln\left(\frac{||\phi(\mathbf{x}_0 + \Delta\mathbf{x}, t) - \phi(\mathbf{x}_0, t)||}{||\Delta\mathbf{x}||}\right), \tag{6.15}$$

for almost every $\Delta\mathbf{x}$, where the limit in t and $\Delta\mathbf{x}$ should be taken so that for some fixed constants K_1 and K_2, $0 < K_1 \leq ||\phi(\mathbf{x}_0+\Delta\mathbf{x}, t)-\phi(\mathbf{x}_0, t)|| \leq K_2 < \infty$. Many presentations of the Lyapunov exponent use a more sloppy definition, i.e.

$$\lambda_1 = \lim_{t\to\infty} \frac{1}{t} \ln\left(\frac{||\mathbf{x}_t - \widetilde{\mathbf{x}}_t||}{||\mathbf{x}_0 - \widetilde{\mathbf{x}}_0||}\right). \tag{6.16}$$

This is correct only for systems that are linear, diverge to infinity, or converge to a fixed point. Equation 6.15 is not a good definition for all purposes, it's rather the first part of an approach to give meaning to equation (6.14).

First, the relative location of the initial points affects the result. If the two points are on the same trajectory, the distance will remain constant, and λ is zero. So there are many

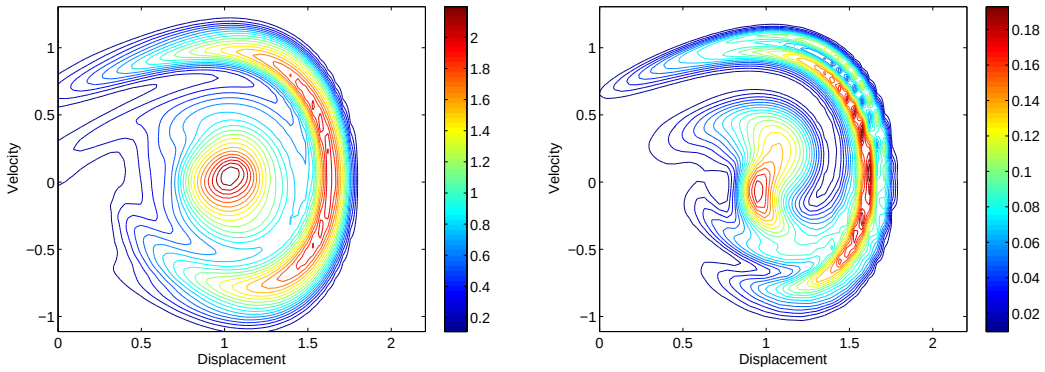


Figure 6.9: PDF of the response of the second 3D system, free noise in the phase, with $\gamma_C = 0.005$. The number of grid points in Θ is 22 in the left image, 44 in the right image, both have 121 grid points in x and y.

Lyapunov exponents, the same number as the dimension of the phase space and at least one will be zero.

Reconsider the Smale horseshoe map discussed in the beginning of this chapter, and illustrated in Figure 6.1. When applying the map several times, points aligned vertically will move closer to each other, while the distance between two points separated horisontally will be doubled every iteration, until one of them reaches the part of the structure that is broken. When the break is between the points, a longer vertical distance between the points is obtained, while the horisontal distance could be doubled, set to zero or anything in between, depending on the placement of the points before the iteration. With n iterates, the discance between two points increase by 2^n at most. In fact, one should measure the distance along the dotted line shown in the figure, and this line length asymptotically approaces 2^n. This means that the largest Lyapunov exponent is $n \ln 2$. [1] Because of the contraction in the other direction, the other Lyapunov exponent must be negative, in fact $-n \ln 2$ to make the system globally stable and area perserving.

Returning back to the continuous case, consider a small n-dimensional compact ball $B_\varrho(\mathbf{x}_0)$ centered in a point $\mathbf{x}_0$ with radius ϱ. Assume that every point in the ball is propagated forward in time by a flow, i.e. consider $\phi(B_\varrho(\mathbf{x}_0), t)$. The short-time dynamics will deform this ball into an ellipsoid. That is, after a finite short time t all orbits which have started in that ball will be in the ellipsoid. The jth Lyapunov exponent is then

$$\lambda_j = \lim_{t\to\infty} \frac{1}{t} \ln\left(\frac{\mathrm{d}l_j(t)}{\mathrm{d}\varrho}\right), \tag{6.17}$$

where $\mathrm{d}l_j(t)$ is the radius of the ellipsoid along its jth principal axis. If the largest principal axis typically is larger than ϱ, the MLE is positive.

[1] The Lyapunov exponent equals 1 if 2 is used as the cardinal number instead of e. This is just a different convention than what is used here, giving the LE spectrum in bits per second or bits per iteration.

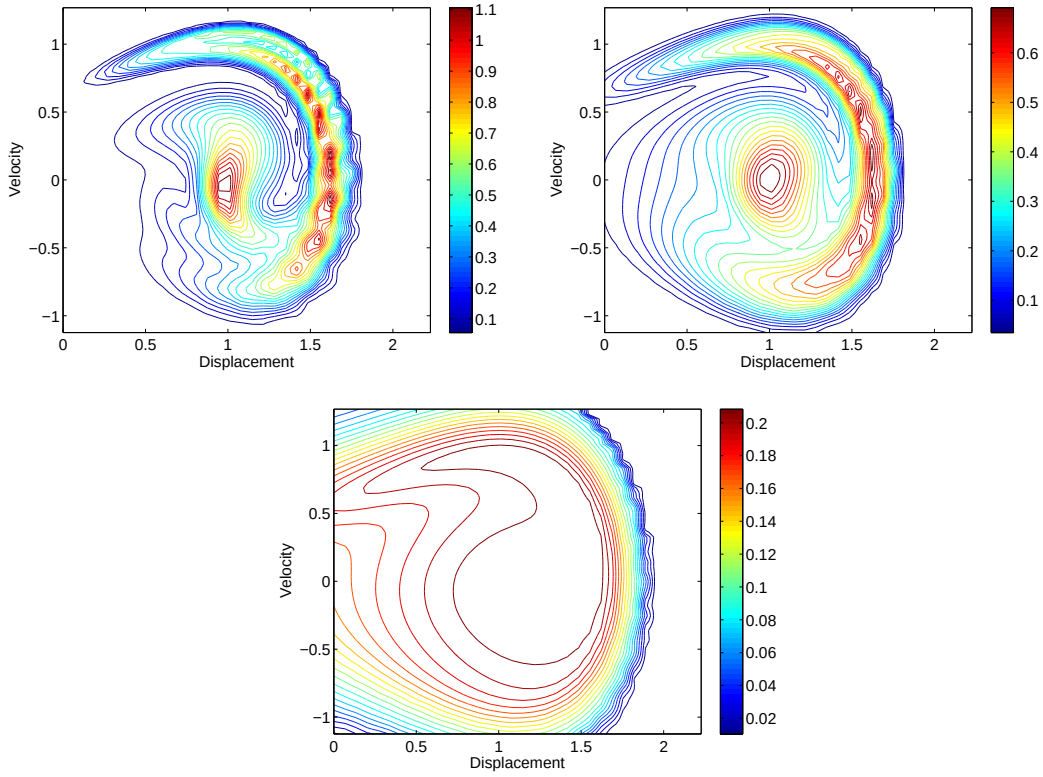


Figure 6.10: PDF of the response of the second 3D system, free noise in the phase, with $\gamma_C = 0.050$, 0.100, and 0.500 in the left, right, and bottom image, respectively. The number of grid points in Θ is 88, while the number of grid points in x and y is reduced by 2/3 to 81.

The limit $t \to \infty$ in equations (6.15) and (6.17) must then be interpreted correctly. If λ_1 is positive from those definitions, the distance between the paths must diverge exponentially to infinity, i.e. at least one of them must diverge. For chaotic systems, the definition is only valid as long as the points are sufficiently close, i.e. that one by time t in equation (6.17) has an ellipsoid. In general, this cannot be guaranteed. A practical interpretation of equations (6.15) and (6.17) is then to perform a rescaling sufficiently often, and use the products of the rescaling factors to compute the Lyapunov exponent. Even this method suffers from lack of accuracy, as one cannot even for a short time propagation ensure linear behaviour of the system. Therefore, this method is not discussed in detail here, but interested readers can find a further discussion in [47] that has many references to methods using rescaling and orhogonalization.

Generally, for a nonlinear system, the shape of $\phi(B_\varrho(\mathbf{x}_0), t)$ will not be an ellipsoid, or any other simple geometric figure. However, since the flow is continuous, compactness should be preserved, and basic properties such as (hyper)volume and surface area could then be computed. One could then define

$$\lambda_1 = \lim_{t\to\infty} \frac{1}{t} \ln\left(\frac{1}{2\varrho} \max_{p_0,p_1} \min_{\varsigma} ||\varsigma(p_0, p_1)||\right), \tag{6.18}$$

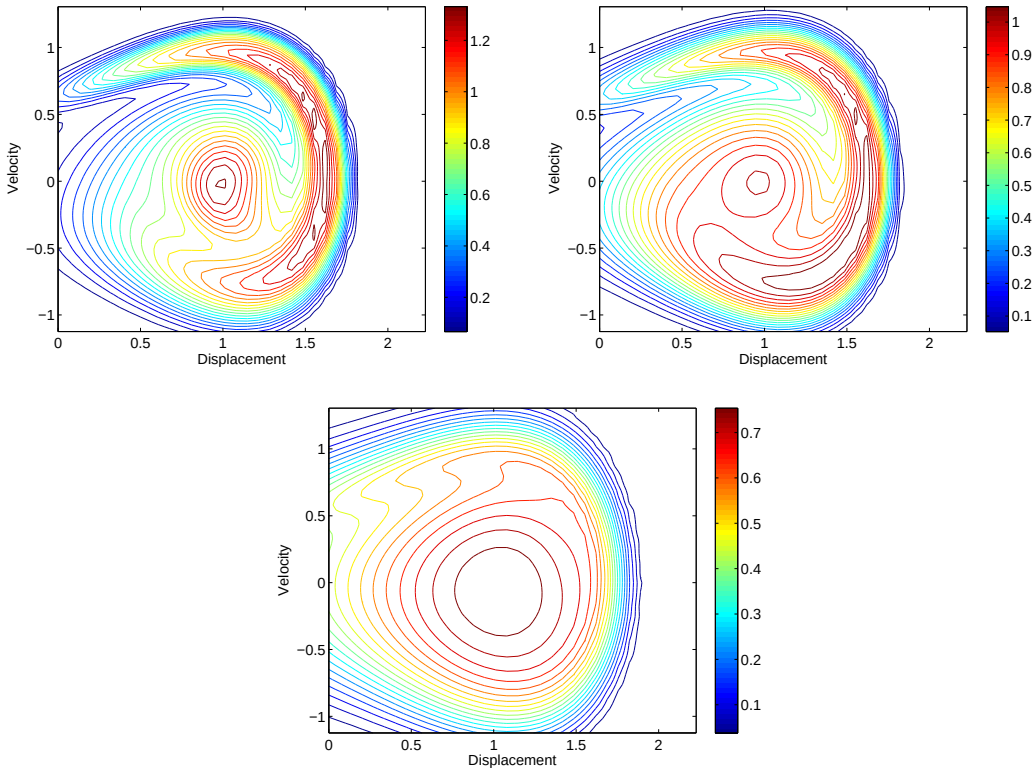


Figure 6.11: PDF of the response of the 4D system, filtered noise, with $\kappa_{\mathrm{D}} = 0.0001$, 0.10, and 1.0 in the left, right, and bottom image.

where the points $p_0, p_1 \in \phi(B_\varrho(\mathbf{x}_0), t)$, each curve $\varsigma \in \phi(B_\varrho(\mathbf{x}_0), t)$ is completely inside the deformed ball, and all the curves ς have p_0 and p_1 as end points. The factor $1/2\varrho$ can be neglected, as it dissappears in the limit. Similarly, the second largest exponent can be defined by

$$\lambda_2 = \lim_{t\to\infty} \frac{1}{\lambda_1 t} \ln \left(\max_{p_0,p_1,p_2} \min_{\delta A} ||A(p_0, p_1, p_2)|| \right), \tag{6.19}$$

where $p_0, p_1, p_2 \in \phi(B_\varrho(\mathbf{x}_0), t)$, and $A \in \phi(B_\varrho(\mathbf{x}_0), t)$ is some 2D surface with $p_0, p_1, p_2 \in A$, and the minimum is taken so that only such surfaces with minimal boundary length are considered.

Chaos is often defined by λ_1. It is stated that if it is positive, the system is chaotic, and the LE is therefore also the number that determines how far into the future the system can be predicted, i.e. the prediction horizon. The nature of chaotic systems is the trajectories' sensitivity to initial conditions, and hence also to the numerically inevitable round-off errors. A high Lyapunov exponent means that the divergence of neighbouring points is high, and the system can only be predicted for a short time. However, these are dynamical systems with sufficient damping, so the trajectories will remain within certain bounds, and can therefore not diverge more than a certain degree.

The reason for this seemingly contradiction is that after some time, the two trajectories will not be close any more, and the basic assumption of linearity is no longer correct. I.e. the separation of the trajectories is no longer exponential. For the definition to work, the original distance between the points have to be chosen sufficiently small for a computation based on a certain length of time. Cartwright [6] defines a local Lyapunov exponent based on $l_j(t)$ the jth principal axis of the ellipsoidal ball evolving in time the state space, in a similar manner as in equation (6.17);

$$\lambda_j(\tau, t) = \lim_{l_j(\tau)\to 0} \frac{1}{\tau} \ln \frac{l_j(t+\tau)}{l_j(t)}. \tag{6.20}$$

Then, the global Lyapunov exponent is the limit as τ goes to infinity;

$$\lambda_j = \lim_{\tau\to\infty} \lambda_j(\tau, t). \tag{6.21}$$

Theoretically, the distance should be measured in some sense "along the attractor", like in equation (6.18), however this equation will be very difficult to implement numerically. If one for the horseshoe map imagine the horisontal distance between points is measured by a thin thread, the thread itself is also broken, and then it is continuously stretched (or shrunk) at a rate given by the LE. This is the same way the MLE should be interpreted for a dynamical system. The stretching of a line element is shown for the Duffing oscillator in Figure 6.15, and for the Lorenz attractor in the right of Figure 6.18, Mathematically, this could be expressed

$$\begin{aligned} \mathbf{x}_0(\tau) &= \mathbf{x}_0 + \tau\boldsymbol{\xi}_0, \quad \tau \in [0,1], \\ \lambda_1 &= \lim_{t\to\infty} \frac{1}{t} \ln \left(\frac{\int_0^1 \sqrt{\left(\frac{\partial\phi(\mathbf{x}_0(\tau),t)}{\partial\tau}\right)^\top \left(\frac{\partial\phi(\mathbf{x}_0(\tau),t)}{\partial\tau}\right)}}{\sqrt{\boldsymbol{\xi}_0^\top \boldsymbol{\xi}_0}} \,\mathrm{d}\tau \right) \\ &= \lim_{t\to\infty} \frac{1}{t} \ln \left(\int_0^1 \sqrt{\left(\frac{\partial\phi(\mathbf{x}_0(\tau),t)}{\partial\tau}\right)^\top \left(\frac{\partial\phi(\mathbf{x}_0(\tau),t)}{\partial\tau}\right)} \,\mathrm{d}\tau \right). \end{aligned} \tag{6.22}$$

Here, $\boldsymbol{\xi}_0$ is an initial perturbation vector of $\mathbf{x}_0$. So the difference from the above formulas is that the whole line segment is considered, rescaling is not strictly necessary, as long as the curve is well represented e.g. by an interpolation scheme, and the limit to infinity exists. Numerically, this method requires addition of points when the line grows longer to ensure constant accuracy. Otherwise, as the line segment curves, the loops between points on the cure will not be seen by the numerical algorithm, and hence the length of the curve will be under-estimated. Similarly to this line method, one could define the second largest LE from the evolution of an initial area, which should be proportional to $\lambda_1\lambda_2$, the third by considering a volume proportional to $\lambda_1\lambda_2\lambda_3$ etc. Also this will require a large number of points for a numerical method, even if the area or volume tends to zero. This is due to the complex structure of the propagated object, and the need for an accurate estimation of its

area or volume to get a reliable Lyapunov spectrum. Hence, this method will in practice be quite complicated.

In [30], the local Lyapunov exponent is defined as the jth largest eigenvalue of the iterated Jacobian

$$\lambda_j(\mathbf{x}) = \lim_{t\to\infty} \frac{1}{t} \ln(\mathrm{eig}_j(e^{J(\mathbf{x})})). \tag{6.23}$$

In [12], an even more simple, but not generally correct, form is presented;

$$\lambda_j(\mathbf{x}) = \ln(\mathrm{eig}_j(e^{J(\mathbf{x})})) = \mathrm{eig}_j(J(\mathbf{x})). \tag{6.24}$$

Here, one should let eig_j be the jth largest eigenvalue measured by the real part only. If a local LE $\lambda(\mathbf{x})$ can be computed for all $\mathbf{x}$ (or at least for all $\mathbf{x}$ along a sample path,) instead of the limit in equation (6.21), one could compute

$$\lambda_1 = \lim_{t\to\infty} \frac{1}{t} \ln\left(\int_0^t \lambda_j(\mathbf{x}_\tau)\,\mathrm{d}\tau\right), \tag{6.25}$$

for the moving trajectory $\mathbf{x}_t$. In one dimension equations (6.23) and (6.24) works. If the local LE is estimated taking m steps or iterations, this converges as $m \to \infty$. However, these are unfortunately incorrect definitions of a local Lyapunov exponent for this purpose, except for some very few exceptions. The problem is that each eigenvalue is associated to an eigenvector, and the composition above requires all eigenvectors to be parallell, which is only trivial in one dimension. I.e. equation (6.24) is valid for 1D systems. Secondly, a system could have a local LE larger than zero from the previous definitions without a positive real part of any eigenvalue of the Jacobian. Streching of line elements could also happen at boundaries between domains with different dynamics, e.g. by grazing bifurcation as in [31].

For the methods discussed above, one would in practice restart the simulation after some time, and estimate the Lyapunov exponent by averaging, which is basically the same as averaging the local Lyapunov exponents over the attractor, weighted by the time spent in each part of the space. Lim and Hamill [30] discusses renormalization and various numerical techniques to handle practical problems like discontinuities, numerical integration error, values below machine precision etc.

If the result of equation (6.25) is zero or negative, the result will be the same as for equations (6.15), and correspondingly for the stochastic versions. It is only if λ from equation (6.25) is positive, and if the system does not diverge exponentially to infinity, that there is a difference. The following argument is often used in discussing Lyapunov exponents;

$$\begin{aligned}
\lambda &= \lim_{t\to\infty} \frac{1}{t} \ln\left(\frac{||\mathbf{x}_t - \widetilde{\mathbf{x}}_t||}{||\mathbf{x}_0 - \widetilde{\mathbf{x}}_0||}\right) \\
&= \lim_{t\to\infty} \frac{1}{t} \left(\ln(||\mathbf{x}_t - \widetilde{\mathbf{x}}_t||) - \ln(||\mathbf{x}_0 - \widetilde{\mathbf{x}}_0||)\right) \\
&= \lim_{t\to\infty} \frac{1}{t} \ln(||\mathbf{x}_t - \widetilde{\mathbf{x}}_t||).
\end{aligned} \tag{6.26}$$

This requries a "true" limit to ∞ and a fixed $||\mathbf{x}_0 - \widetilde{\mathbf{x}}_0||$. For linear systems, exponential separation between trajectories is equivalent to exponential behaviour, so then

$$\lambda = \lim_{t\to\infty} \frac{1}{t} \ln(||\mathbf{x}_t||). \tag{6.27}$$

The largest Lyapunov exponent for a system can only be calculated from a probability density function in some special cases. One such example is given by Wedig [54], who separates amplitude and phase for a system with exponential behaviour (non-chaotic), and finds the exponential (non-stationary) divergence of the amplitude from the stationary distribution of the phase.

6.8 Other strange attractors

Strange Nonchaotic Attractors (SNAs) have complicated or fractal attractors, but the largest Lyapunov exponent is not positive. They are observed in dissipative systems driven by incommensurate frequencies [59]. Without the Lyapunov exponent, it is impossible to distinguish SNAs from true chaotic attractors. This also means that the PDF is itself not enough to classify a system as chaotic.

Indeed, a system could also have a strong sensitivity to the initial value and a complicated map of the basins of attraction without being chaotic. This feature has been described for the deterministic gear problem [56], which will also be discussed in section E. When noise is added to such systems, the periodic solutions could dissappear, and realizations could also be very sensitive to initial conditions and round-off errors. This phenomenon is related to the discussion in section 1.3.9. However, the attractor could still have a quite simple structure, as section E shows.

6.9 Stochastic Lyapunov Exponent

The definition of stochastic Lyapunov exponents SLEs could be given by taking an expectation value over all time realizations of the process. So equation (6.15) would read

$$\widetilde{\lambda} = \mathrm{E}\left[\lim_{t\to\infty} \frac{1}{t} \ln\left(\frac{||\mathbf{X}_t - \widetilde{\mathbf{X}}_t||}{||\mathbf{X}_0 - \widetilde{\mathbf{X}}_0||}\right)\right]. \tag{6.28}$$

As above, one should rather use the definition of a local Lyapunov exponent, as equation (6.23) and define the SLE

$$\widetilde{\lambda} = \mathrm{E}[\lambda(\mathbf{X})] = \int_{\mathbb{R}^n} \lambda(x) f(x)\, \mathrm{d}x, \tag{6.29}$$

if a stationary density $f(x)$ exist.

The interpretation is more difficult. There exist systems where the LE is negative, while the SLE is positive when noise is included [12], see section 6.17 for a brief discussion,

and opposite; addition of noise to some chaotic systems with a positive LE could result in a negative SLE, as is briefly discussed in section 6.18. In stochastic systems, periodic orbits doesn't really make sense, and the criterion of a topologically transitive attractor must be extended to the (possibly non-stationary) PDF and a probability measure.

However, when the noise level goes to zero and the response PDF narrows around the deterministic attractor, the SLE will converge to the LE. This way of estimating the LE could be valuable, as it provides information about the contributions of the local LEs and the sensitivity to perturbations, as the PDF indicates the amount of time spent in various parts of the attractor.

6.10 Lyapunov exponents from the tangent space

This subsection will basically follow Baxendale [5] and Wihstutz [55], which extensively builds on L. Arnold, 1998 [1] and R.Z. Khas'minskii, 1967 [24]. Although this work only will consider additive noise, Baxendale's more general formulation will be followed initially.

The linearization of equation (1.12) is written using the differential operator D, where Da also is referred to as the Jacobian of $a(x)$.

$$\mathrm{d}\mathbf{Y}_t = Da(\mathbf{X}_t)\mathbf{Y}_t\,\mathrm{d}t + \sum_{j=1}^{r} Db_j(\mathbf{X}_t)\mathbf{Y}_t\,\mathrm{d}W_t^j \quad \mathbf{Y}_0 = \mathbf{y}_0. \tag{6.30}$$

While Baxendale and others usually refers to this as linearization along trajectories, this text will also consider equation (6.30) as an extension of equation (1.12), so that an SDE system over $\mathbb{R}^{2n}$ is obtained.

Note that while it is assumed that equation (1.12) is bounded in the state space, the $\mathbf{Y}_t$ variable could diverge to infinity with probability 1, i.e. for any fixed $K < \infty$, $\lim_{t\to\infty} P(||\mathbf{Y}_t|| < K) = 0$.

Now, the MLE is defined

$$\lambda_1 = \lim_{t\to\infty} \mathrm{E} \ln\left(\frac{||\mathbf{Y}_t||}{||\mathbf{y}_0||}\right). \tag{6.31}$$

The requirements for the Lyapunov exponents to be globally well-defined will not be discussed in detail here, but are found in e.g. [5] and [1]. Basically, the system must be non-explosive, the Stratonovich-version of the SDE (1.12) must be full rank, and the projection of the solution of equation (1.12) onto $\mathbf{S}^{n-1}$ must be dense.

Khasminskii [24] studies a linear system, where $Da(\mathbf{x})$ and the $Db_j(\mathbf{x})$ are constant matrices in $\mathbf{x}$ for every j. For this case, the discussion is more simplified, and only $Da(\mathbf{0})$ and $Db_j(\mathbf{0})$ need to be evaluated.

Define the operator $Q(\mathbf{x}, \boldsymbol{\theta}) : \mathbb{R}^n \times S^{n-1} \to \mathbb{R}$ by

$$Q(\mathbf{x}, \boldsymbol{\theta}) = \langle Da(\mathbf{x})\boldsymbol{\theta}, \boldsymbol{\theta}\rangle + \frac{1}{2}\sum_{j=1}^{r} \left\{||Db_j(\mathbf{x})\boldsymbol{\theta}||^2 - 2\langle Db_j(\mathbf{x})\boldsymbol{\theta}, \boldsymbol{\theta}\rangle^2\right\}. \tag{6.32}$$

Theorem 3. *Assume that the martingale*

$$M_t = \sum_{j=1}^{r} \int_0^t \langle Db_j(\mathbf{X}_s)\theta_s, \theta_s \rangle \, \mathrm{d}W_s^j$$

is an L^2 martingale, and $(1/t)\mathrm{E}\langle M \rangle_t$ is bounded. Assume further that $Q \in L^1(\nu)$, i.e. that Q is integrable with respect to the probability measure. Then the MLE λ_1 can be computed by

$$\lambda_1 = \int_{\mathbb{R}^n \times S^{n-1}} Q \, \mathrm{d}\nu \qquad \textit{almost surely,} \tag{6.33}$$

with ν as the invariant probability measure for $(\mathbf{x}, \boldsymbol{\theta})$.

Proof. Dividing equation (6.30) by $||\mathbf{Y}_t||$ and integrating by Itô's formula gives

$$\ln(||\mathbf{Y}_t||) - \ln(||\mathbf{y}_0||) = \int_0^t Q(\mathbf{x}_\tau, \boldsymbol{\theta}_\tau) \, \mathrm{d}\tau + M_t. \tag{6.34}$$

In the limit $t \to \infty$, $M_t \to 0$ almost surely. When the function Q is integrable, and the system is ergodic, the long-term time integral of the process can be exchanged with a spatial integral with respect to the stationary probability measure almost surely according to the ergodicity. In the limit $t \to \infty$, the lhs of equation (6.34) equals λ_1 almost surely. □

Khasminskii's result is similar, except that the dependence on $\mathbf{x}$ dissappears since $Da(\mathbf{x})$ and $Db_j(\mathbf{x})$ are constant. This again makes it possible to only consider the probability measure over S^{n-1} instead of $\mathbb{R}^n \times S^{n-1}$. However, numerically, a method like PI would still need to calculate the higher-dimensional PDF unless the SDE also could be separated. A practical or visual interpretation of the Q operator in equation (6.33), is that this describes the rate of change in the radial variable over a time step Δt. That is, with $\rho_t = ||Y_t||$,

$$\lambda_1 \approx \lim_{t \to \infty} \frac{1}{\Delta t} \mathrm{E}\left[\frac{\rho_{t+\Delta t}}{\rho_t} \right]. \tag{6.35}$$

The dynamics, and hence $\rho_{t+\Delta t}$ is determined from the position in the state space at time t, so it makes sense to write it as a function $\rho(\mathbf{x}', \boldsymbol{\theta}', \Delta t)$. Since this is a linearized system with $\mathrm{d}Y_t = Da(\mathbf{X}_t)Y_t \, \mathrm{d}t$, it can be scaled arbitrarily, which basically means that one can choose $\rho_t = 1$, which implies

$$\lambda_1 \approx \frac{1}{\Delta t} \mathrm{E}\left[\rho(\boldsymbol{x}, \boldsymbol{\theta}', \Delta t)\right] = \frac{1}{\Delta t} \int\int_{\mathbb{R}^n \times S^{n-1}} f(\mathbf{x}', \boldsymbol{\theta}', t') \ln(\rho(\boldsymbol{x}, \boldsymbol{\theta}', \Delta t) \, \mathrm{d}\mathbf{x}' \, \mathrm{d}\boldsymbol{\theta}'. \tag{6.36}$$

Note that when integrating this over many time steps, only the expectation operator separates this from equation (6.17). The expectation could also be taken with respect to the probability measure over θ' and $\mathbf{x}$. As stated earlier, one will in the limit $t \to \infty$ observe ϱ as the largest principal axis with probability one. The probability measure over $\boldsymbol{\theta}$ captures this behaviour. For additive white noise processes $\mathrm{E}(M_t) = 0$, $Db_j(\mathbf{x}) \equiv 0$ for $j > 0$, and one obtains

$$\lambda_1 = \int_{\mathbb{R}^n \times \mathbf{S}^{n-1}} \langle Da(\mathbf{x})\boldsymbol{\theta}, \boldsymbol{\theta} \rangle \, \mathrm{d}\nu \qquad \text{almost surely.} \tag{6.37}$$

6.11 Time pattern of MLE for time-dependent SDEs

The Lyapunov theory has been based on the assumption that the divergence of nearby trajectories and change of length for the curve discussed above follows an exponential law $\exp(\lambda_1 t)$. For an autonomous system, this will be true in the limit. However, if there is time dependence, the long term observation could also be time dependent, typically $\exp(\lambda_1(t)t)$, where $\lambda_1(t)$ now varies with time. If the time-dependent term is periodic, $\lambda_1(t)$ will have the same periodicity. When $\exp(\lambda_1(t)t)$ is observed, $\lambda_1(t)$ is recovered by

$$\lambda_1(t) \approx \frac{1}{\Delta t}(\log(\exp[\lambda_1(t+\Delta t)(t+\Delta t)]) - \log(\exp[\lambda_1(t)t])). \tag{6.38}$$

This should then be compared to $\mathrm{E}(Q(t))$ where $Q(t)$ is the now time-dependent operator from equation (6.32), and the expectation follows equation (6.33), where the measure is also time-dependent. In figures, the function $\lambda_1(t)$ is, in lack of a proper term, denoted the *time pattern* of the MLE. The long term averaged MLE will equal the time average of the time pattern, and it suffices to average over one period after reaching steady state behaviour. When using this method to classify a system as chaotic or non-chaotic, one must determine if this time average is positive or negative, respectively.

One could also eliminate the time variable by replacing it by a state-space variable z with $\dot{z} \equiv 1$. It is trivial that two starting points in this space placed on the same trajectory, will follow each other with constant distance, hence one Lyapunov exponent must be zero. Further, assume that the system is chaotic or diverging. Consider two trajectories starting in nearby points $\mathbf{p}$ and $\mathbf{q}$, and that these points are also separated by Δt in the auxillary z variable. Then $\phi(\mathbf{p}, t)$ cannot diverge more per time unit from $\phi(\mathbf{q}, t)$ than from $\phi(\mathbf{q}, t + \Delta t)$. This is shown by the triangle inequality together with the knowledge that the separation in z is constantly equal to Δt. Hence, there is no need for introducing the auxillary variable z for time-dependent systems to find the MLE, unless the MLE one would like to find is zero.

6.12 Calculating Lyapunov exponents through response PDF

Using the extended system, equation (1.12) together with equation (6.30), the response PDF in $2n$ dimensions can be computed with the PI method described earlier. However, since $||\mathbf{Y}_t|| \to \infty$ with probability one if $\lambda_1 > 0$, the direct method requires a grid in $\mathbf{y}$ that grows as $\exp(\lambda_1 t)$ with sufficient accuracy. To avoid this, and to reduce the number of state-space variables and save CPU time, $\mathbf{y}$ is replaced with $\boldsymbol{\theta}$, and the radial component $\varrho = ||\mathbf{y}||$ is ignored. This change of variables is possible by adjusting the time stepping method to end up with angle and radius, assuming the initial value $\varrho_t = 1$ everywhere at the beginning of a time step. For a deterministic system, there is no path integral, just a propagation of points, and then this assumption would not cause any errors. Therefore, it is assumed to hold fairly well also for small noise levels.

In fact, since this implementation of PI calculates the time steps backwards, one should rather use the assumption $\varrho_{t+\Delta t} = 1$, calculate $\boldsymbol{\theta}'$ from grid points in $\boldsymbol{\theta}$ and interpolate the PDF over $\mathbb{R}^n \times S^{n-1}$. The problem is that this surface is very hard to interpolate due to singular points and closed loops where the angle always increases in one direction (due to the modulo 2π). To avoid these problems, use the coarse nearest neighbour interpolation method on S^{n-1}.

To calculate the MLE numerically, any of equations (6.33), (6.35), (6.36), or (6.37) can be chosen. Numerical experiments gave practically identical results, which just indicates that the time step Δt is sufficiently small and the time stepping method is sufficiently accurate that the approximations in the latter equations hold.

In this project, the full PDF method has only been implemented for two-dimensional systems, due to the complexity of the code. For the Duffing equation (6.1), which will be discussed more in section 6.14, the result was that the conditional PDFs $f(\boldsymbol{\theta}|\mathbf{x})$ at almost every point $\mathbf{x}$ were ellipses with high eccentricity. This is illustrated in Figure 6.12. Again, the symmetry is no surprise since the system is linear, so $-\mathbf{Y}_t$ should have the same statistical properties as $\mathbf{Y}_t$. With small noise level, it is no surprise that the angle giving the MLE dominates.

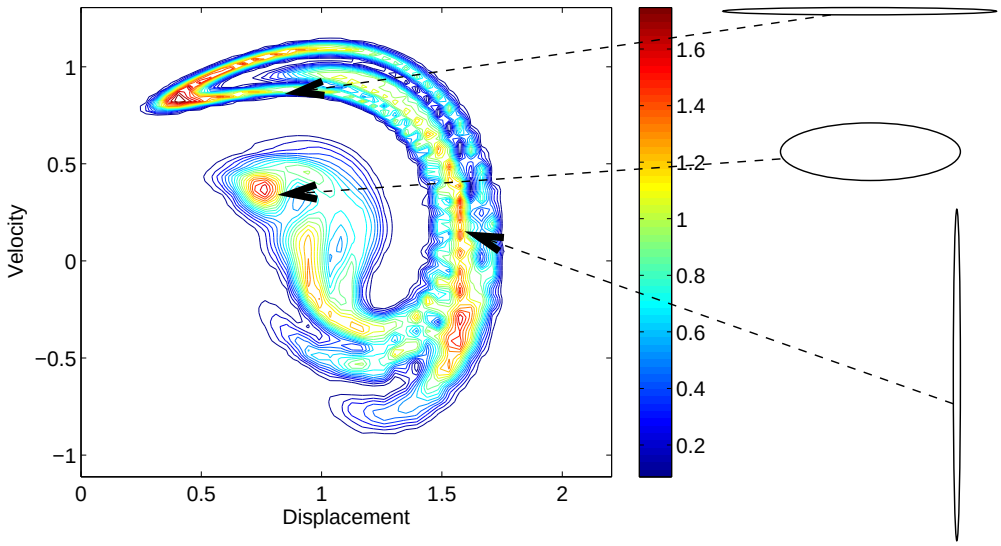


Figure 6.12: Snapshot of the response PDF for the chaotic Duffing equation. The angular probability distribution of the perturbation variables $f(\theta|x, y)$ is indicated at three points by aligned ellipses.

This observation leads to at least two new approximations. One is to assume that there is only one dominant angle $\boldsymbol{\theta}$, try to compute this and ignore working with S^{n-1}. This "dominant angle" method would again be exact for a deterministic system, except for the spatial discretization and time stepping error, and can be used if one chooses some value for the initial angles and angles for points $\mathbf{x}$ outside the grid. Here, the dominant angle

is stored separately from the probability density. For each point in the state space, one approximates the angle at $\mathbf{x}'$ by the nearest neighbour method, and uses a simple time-stepping algorithm to propagate this angle, represented by an n-dimensional point on S^{n-1} forward in time to $\mathbf{x}$. The idea will be explained more in detail in sections 6.14 and 6.15. Figure 6.12 illustrates how the "dominant angle"changes with the dynamics over the first period.

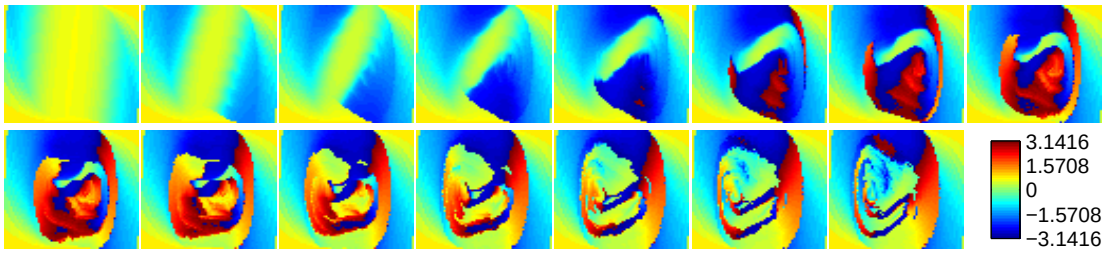


Figure 6.13: Estimated dominant angle through the first period of the Duffing oscillator.

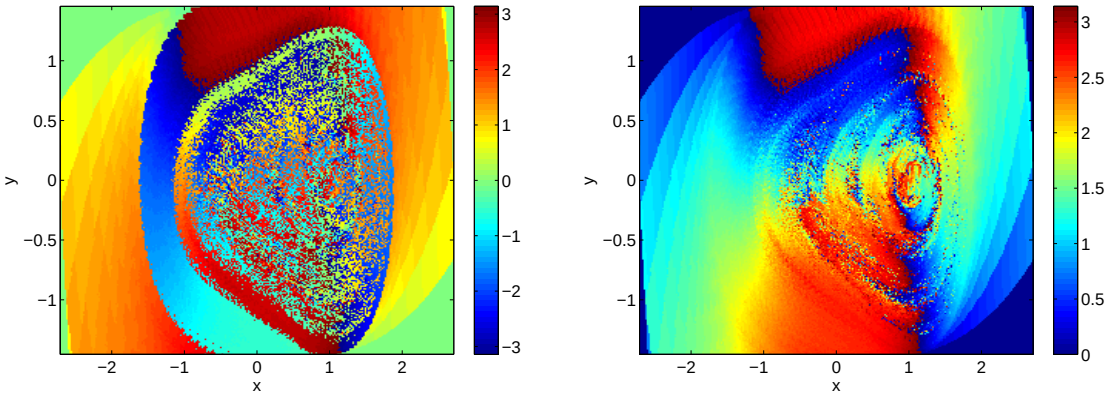


Figure 6.14: Estimated dominant angle after ten full periods of the Duffing oscillator. In the left image plotted with scale from $-\pi$ to π, as calculated by the numerical routine in parallell with PI. To the right, the same plot except that every angle θ is identified with $\theta \pm \pi$, so a scale from 0 to π is sufficient.

The second, more sophisticated version, is to parametrically represent the ellipses or ellipsoids by the length and direction of the principal axes, which still should reduce the need of computer memory, and possibly also the CPU-time. This is a direct application of the idea from section 5.6. Here, one just evaluates the ellipse at each step backwards. The challenge is to find the principal axes, this has here been done in a crude manner by just propagating the angle of the axes forwards separately, and then re-orthogonalize. Here, the probability density values are found implicitly by integrating the values over all the ellipsoids. However, for scaling of the PDF and control, it makes sense to store the total probability at each state-space point separately. The implementation of this idea worked

well, but the programming code became extremely complicated. So additional ideas are needed to make this a practical approach for future work.

6.13 Projective space

The projective space, Π^{n-1}, is obtained by associating all opposite angles in the hypersphere S^{n-1}.

Using the dominant angle idea, the angular distribution quickly becomes entangled and very complicated, as shown for the Duffing equation in Figure 6.12 for the first period, and in the left of Figure 6.14 after ten full periods. Remembering that $\boldsymbol{\theta}$ and $-\boldsymbol{\theta}$ could be identified through the linearity of the extended half part of the SDE system, one could change S^{n-1} into the projective space Π^{n-1}, the left and right image of Figure 6.14 are identical, where the latter is a remarkable smooth picture in comparizon. Remember that the nearest neighbour interpolation method does not introduce any smoothing of the PDF, but many possibilities of numerical errors. This result clearly indicates that Π^{n-1} is the correct space to consider for this study.

6.14 Example: MLE in smooth two-dimensional SDEs

A system with two variables x and y would be extended to

$$\begin{aligned}
\dot{x} &= a_1(x, y, t), \\
\dot{y} &= a_2(x, y, t), \\
\dot{\xi} &= \frac{\partial a_1}{\partial x}\xi + \frac{\partial a_1}{\partial y}\eta = a_3(x, y, \xi, \eta, t), \\
\dot{\eta} &= \frac{\partial a_2}{\partial x}\xi + \frac{\partial a_2}{\partial y}\eta = a_4(x, y, \xi, \eta, t).
\end{aligned}$$

The original system does not depend on the perturbations ξ and η. Therefore, these arguments will not be introduced in the functions a_1 and a_2 or the discretized schemes $\widetilde{a}_1$ and $\widetilde{a}_2$. On the other hand, a_3 and a_4 and the discretized approximations will generally depend on all the variables in the extended state space, as shown above.

When propagating this system forward in time, or if possible, solving the system for some appropriate initial condition, one can calculate the MLE by

$$\lambda_1 = \lim_{t\to\infty} \frac{1}{t} \ln\left(\sqrt{\frac{\xi_t^2 + \eta_t^2}{\xi_0^2 + \eta_0^2}}\right). \tag{6.39}$$

This limit should again be correct for almost every initial value (ξ_0, η_0). Again, the problem with this approach is that the system is generally not bounded in ξ and η. If $\lambda_1 > 0$ these variables can become arbitrarily large in absolute value as time increses. For the PI method,

some problems of this kind can be treated, but generally a bounded behaviour would be simpler.

The angle $\theta_t = \tan^{-1}(\eta_t/\xi_t)$ is clearly bounded to the circle. For a deterministic system, the following holds. Given the angle θ' at time t', the angle θ after a time t can be found by

$$\tan(\theta) = \frac{\phi_3(x', y', \cos\theta', \sin\theta', \Delta t)}{\phi_4(x', y', \cos\theta', \sin\theta_{t'}, \Delta t)}. \tag{6.40}$$

This means that the state space can be extended from two (x, y) to three (x, y, θ) variables, instead of the four above. Given a stationary PDF for the system over this increased state-space, the MLE can be found through either

$$\lambda_1 = \int_{\mathbb{R}^2\times\mathbb{S}} f(x, y, \theta)\rho(x, y, \cos\theta, \sin\theta, \Delta t)\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}\theta \tag{6.41}$$

with a numerical estimation of ρ, or with equation (6.33) inserted for Q,

$$\lambda_1 = \int_{\mathbb{R}^2\times\mathbb{S}} f(x, y, \theta)\left(\frac{\partial a_1}{\partial x}\cos^2\theta + \left(\frac{\partial a_1}{\partial y}\frac{\partial a_2}{\partial x}\right)\cos\theta\sin\theta + \frac{\partial a_2}{\partial y}\sin^2\theta\right)\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}\theta. \tag{6.42}$$

In the PI method, the extended system is written on the standard form with TPD

$$p(x, y, \theta|x', y', \theta') = \delta(x - \widetilde{a}_1(x', y', t')\Delta t)\frac{\exp\left(-\frac{1}{2}\frac{(y-\widetilde{a}_2(x',y',t')\Delta t}{\gamma^2\Delta t}\right)}{\sqrt{2\pi\gamma^2\Delta t}}$$
$$\delta\left(\theta - \tan^{-1}\left(\frac{\widetilde{a}_3(x', y', \cos\theta', \sin\theta', t')}{\widetilde{a}_4(x', y', \cos\theta', \sin\theta', t')}\right)\right),$$

where the final δ-function is based on expression (6.40).

First, an arbitrary line segment is propagated forward in time by the Duffing oscillator. The effect is illustrated in Figure 6.15 for the first iterations, slightly more than a period.

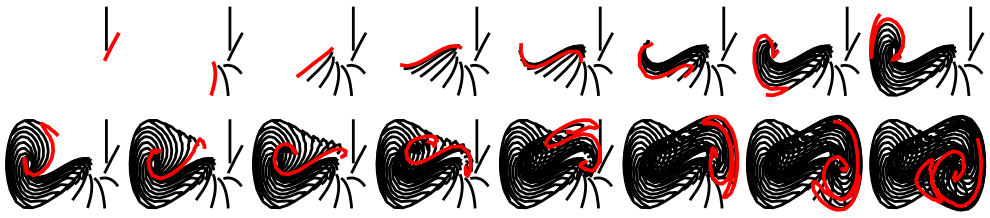


Figure 6.15: Propagation of a line segment by the Duffing oscillator. The scale in each image is the same as in Figure 6.14. The red curve is in each plot the last to emphasize the dynamical behaviour, the black curves are from the previous time steps.

The next figure, Figure 6.16, shows log of the length of a line segment propagated by the Duffing oscillator, and the according time pattern calculated using equation (6.38). As

the attractor for this system is known, one could start with a small line segment placed on the attractor, which seems to give the same long-term effect, but with a slightly shorter transient, also shown in Figure 6.16. The full PI method with the Q operator and based on the dominant angle gives, as stated above, similar results. These are shown in Figure 6.17.

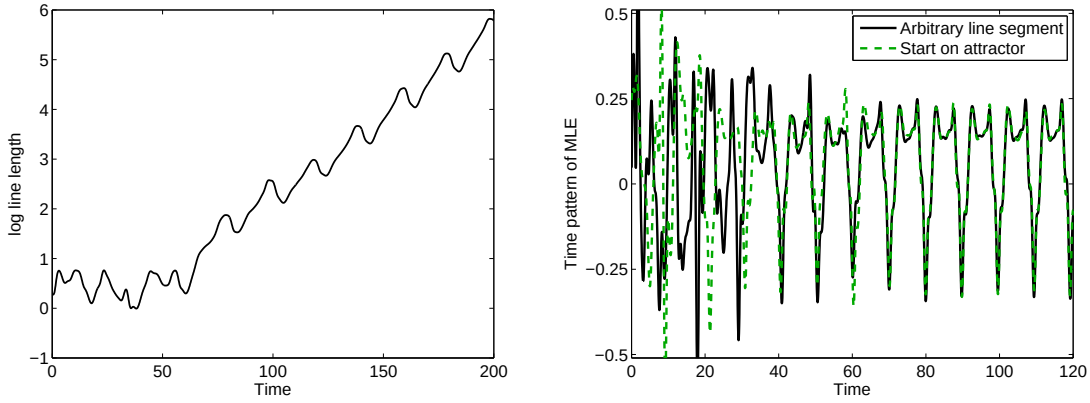


Figure 6.16: Log of the length of a line propagated in time by the Duffing oscillator to the left, and the time pattern for the MLE it generates to the right. The line segments position at $t = 0$ was chosen to be vertical, parallell to the y-axis. The right plot also shows the time pattern for a line segment initially placed vertically at $x = 1.5$ at $t = 0$, which should be on the attractor.

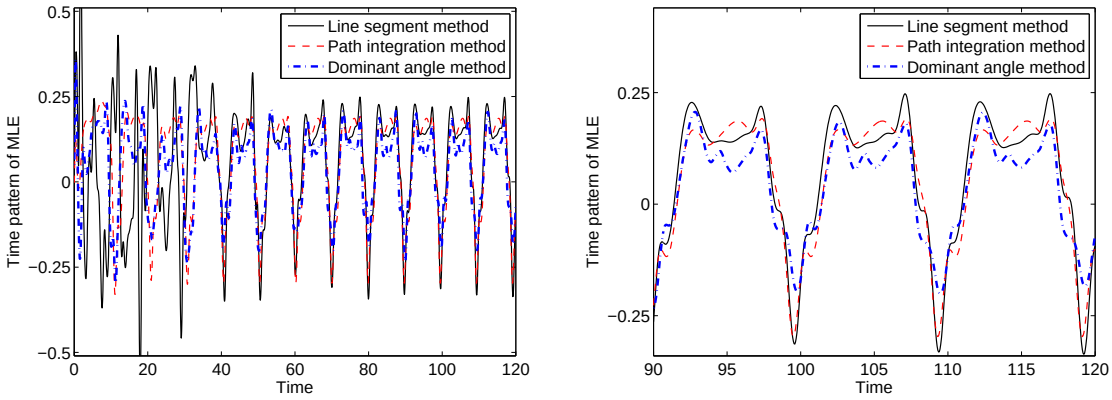


Figure 6.17: Time pattern of the MLE for the Duffing oscillator, computed with three different methods. The left plot shows the transient, illustrating the number of time steps required for convergence, the right plot zooms in on the more steady state behaviour.

6.15 Example: MLE in smooth three-dimensional SDEs

In a general case, the extended system could be expressed by $2n-1$ state-space variables, where the last $n-1$ corresponds to a space angle. This again introduces new challenges, as

spheres and hyper-spheres are more difficult to represent by an interpolated grid. Although a uniform rectangular grid is not applicable, uniform triangularizations exist. It should also be possible to represent the PDF as ellipsoids, storing the length and direction for the n principal axis. This has not been investigated for this project, basically because even $n = 3$ gives a five-dimensional system that currently cannot be solved in short time and high accuracy. Instead, the dominant angle idea has been persued a bit.

Consider the following Lorenz system

$$\begin{aligned} \mathrm{d}X_t &= (-10X_t + 10Y_t)\,\mathrm{d}t, \\ \mathrm{d}Y_t &= (28X_t - Y_t - X_tZ_t)\,\mathrm{d}t, \\ \mathrm{d}Z_t &= (-\frac{8}{3}Z_t + X_tY_t)\,\mathrm{d}t + \gamma\,\mathrm{d}W_t, \end{aligned} \tag{6.43}$$

and its linearization

$$\begin{aligned} \mathrm{d}\xi_1 &= (-10\xi_1 + 10\xi_2)\,\mathrm{d}t, \\ \mathrm{d}\xi_2 &= (28\xi_1 - \xi_2 - Z_t\xi_1 - X_t\xi_3)\,\mathrm{d}t, \\ \mathrm{d}\xi_3 &= (-\frac{8}{3}\xi_3 + Y_t\xi_1 + X_t\xi_2)\,\mathrm{d}t. \end{aligned} \tag{6.44}$$

This system is autonomous, so one should expect that the time pattern of the MLE converges to one value. Running the MC simulation with the program from [57] with $\gamma = 0$ gave $\lambda_1 = 0.906 \ln 2 = 0.628$. The results of the propagated line segment method and the dominant angle method with a noise level $\gamma = 0.005$ are shown in the left plot of Figure 6.18. The line segment method takes a long time to converge. In fact, the simulation tool stopped when the curve was represented by more than two hundred thousand points, since each time step then becomes very calculation intensive with the simple implementation. The dominant angle method on the other hand, converged very rapidly. Here the grid size was $128 \times 128 \times 128$, the interpolation was performed with cubic splines with a truncation for negative values, and FFT was used for the path integral. The space angle is represented by two variables, say θ and ϑ, such that

$$\begin{aligned} x &= \cos\theta \sin\vartheta, \\ y &= \sin\theta \sin\vartheta, \\ z &= \cos\vartheta. \end{aligned}$$

Initially θ and ϑ was set to an arbitrary value, constant for the whole grid. The propagation of the angles then basically followed and the MLE was computed to be 0.58. The three-dimensional attractor for this system has a very thin structure, with two butterfly-like wings. This structure also appears directly from the line length method, illustrated in the right hand image in Figure 6.18. Such plots from different realizations clearly indicate that the angle the line traverses at specific points will be almost deterministic. The angle also follows the attractor, which is also shown in [57]. This is not generally the case, [57] also shows a Rössler system where the perturbation direction is nearly orthogonal to the flow.

The main reason for the discrepancy in results is assumed bo be the grid resolution. The thin structure of the attractor is hard to represent well, even with 128 grid points in each direction.

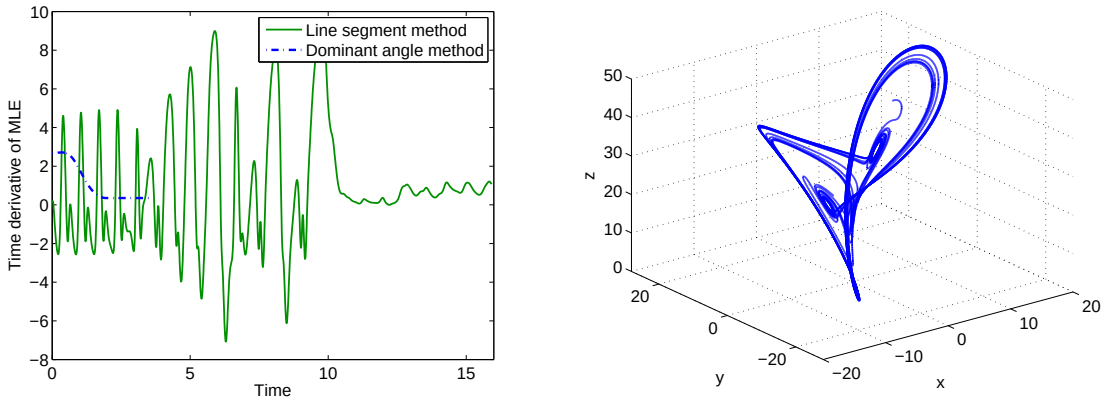


Figure 6.18: Left: Time pattern of the MLE for the Lorenz attractor, computed by length of a propagated line segment and by PI and dominant angle. Right: The line segment after the first few seconds indicates the spatial structure of the Lorenz attractor.

6.16 MLE in nonsmooth systems

First, chaotic behaviour can be found in nonsmooth systems, e.g. shown by Kahraman, [22], and studied phenomenologically in appendix F and [33]. For such systems, it is already mentioned that one might need to take special care of time stepping and interpolaton, see section 5.8.

For such systems, the propagated line segment method should work to compute the MLE, or one could try running simulations with two points, rescaling and orthogonalization. Other methods fail, because a nonsmooth system cannot be differentiated everywhere. It might be tempting to study an approximated system where the shifts between the domains are smoothened out, e.g. by a function like equation (6.47) used in the next two sections. However, such systems exhibit an important phenomenon called *grazing bifurcations* that is not preserved under smoothening. At the points where the vector field is not smooth, trajectories that come in with a suitable angle can either cross the boundary or be reflected. Thus, an instantaneous, nonlinear deviation of nearby trajectories is obtained. This was also strongly pointed out by John Hogan in [17].

6.17 Positive stochastic Lyapunov exponent due to noise

This is a continued discussion of a simple analytical example from Dennis et al 2003 [12]. Consider the system

$$\mathrm{d}X_t = (\alpha X_t - \beta X_t^2)\,\mathrm{d}t, \quad \alpha, \beta, X_0 > 0. \tag{6.45}$$

The Jacobian is the derivative, and also its own local LE, and equals $\alpha - 2\beta X_t$. So for $X < \alpha/(2\beta)$, the local LE is positive, and trajectories diverge in this area. It is shown that all trajectories end up in the stable point $X_\infty = \alpha/\beta$, so this is the stable attractor, and the LE is $\lambda = -\alpha$.

Now add a multiplicative noise term to equation (6.45)

$$\mathrm{d}X_t = (\alpha X_t - \beta X_t^2)\,\mathrm{d}t + \gamma X_t\,\mathrm{d}W_t, \quad \gamma^2 < 2\alpha. \tag{6.46}$$

The PDF for this system is known to be a gamma distribution

$$f(x) = \frac{p^q}{\Gamma(p)} x^{p-1} e^{-qx}, \quad p = \frac{2\alpha}{\gamma^2} - 1, \quad q = \frac{2\beta}{\gamma^2}.$$

The expectation value is $\mathrm{E}(X) = p/q$, which gives the SLE

$$\widetilde{\lambda} = \mathrm{E}(\alpha - 2\beta X) = \gamma^2 - \alpha.$$

If $\gamma^2 < \alpha$, there is no confusion that the system is stable and non-chaotic, and as $\gamma \to 0$, $\widetilde{\lambda} \to \lambda$ as stated above. If $\alpha < \gamma^2 < 2\alpha$, the SLE is positive. Most of the time the response has values close to zero, and there is an overall or averaged sensitivity to initial conditions. However, it is not clear if this behaviour really qualifies to be denoted chaotic. If a noise level $\gamma^2 > 2\alpha$ is used, the system will end up in the absorbing state zero in finite time with probability one, and the only reasonable value for the SLE is 2α. So two nearby trajectories subjected to the same realization of the noise process will diverge from each other, and almost surely spend very different amount of time before being absorbed at zero. This kind of behaviour cannot be seen in deterministic systems.

Theoretically, it should be possible to construct a vector field $a(\mathbf{x}, t)$ such that a region with large positive LLEs is close to a globally stable deterministic attractor with negative LLEs on its support. In such a case, noise could push the system off the attractor and into the region with diverging paths. With sufficiently high noise and sufficiently small damping towards the deterministic attractor, the system could become chaotic. A trivial example would be to implement a weakly damped oscillator inside a chaotic system

$$\begin{aligned}
&\mathrm{d}X = Y\,\mathrm{d}t, \\
&\mathrm{d}Y = \gamma\,\mathrm{d}W + \\
&\quad \begin{cases}
-(\omega^2 X + \alpha Y)\,\mathrm{d}t, & X^2 + Y^2 < \rho^2, \\
(\zeta(X,Y)a_2(X,Y,t) - (1-\zeta(X,Y))(\omega^2 X + \alpha Y))\,\mathrm{d}t, & \rho^2 \le X^2 + Y^2 < R^2, \\
a_2(X,Y,t)\,\mathrm{d}t & R^2 \le X^2 + Y^2.
\end{cases}
\end{aligned}$$

Here, $\mathrm{d}Y = a_2(X, Y, t)\,\mathrm{d}t$ would make the system chaotic with an attractor that overlaps the ball centered in the origin with radius ρ. This way, the deterministic system will spiral down to the origin in finite time with probability one. The weighting function $\zeta(x, y, t) \in [0, 1]$ is constructed to smooth the vector field, e.g. linearly

$$\zeta(x, y) = \frac{x^2 + y^2 - \rho^2}{R^2 - \rho^2} \tag{6.47}$$

The parameters should be chosen such that $R > \rho > 0$ with R and ρ small, and $\alpha > 0$ sufficiently small so that noise easily can throw a realization out of the balls with radius ρ and R. However, this would be a very special construction, and the author has not seen such examples in other places.

The system from equation (6.17) deserves to be studied by MC and PI, since it illustrates some of the considerations one should make for such special cases. Using the chaotic Duffing equation,

$$a_2(x, y, t) = \frac{1}{5}x - \frac{8}{15}x^3 - \frac{1}{25}y + \frac{2}{5}\cos(0.32t),$$

and parameters $\alpha = 1/2$, $\omega^2 = 1/5$, $\rho^2 = 0.007$, $R^2 = 0.01$, MC simulations show that a noise level $\gamma = 0.005$ seemingly after a few cycles pushes any realization to stay within the ball with radius ρ, while $\gamma = 0.05$ does not seem to have this behaviour. There should always be a small probability of leaving the inner radius ρ ball and spend a cycle or more in the chaotic region, but this phenomenon was not seen for the lowest noise level for the MC simulations, in spite of many trials and fairly long runs. One realization with MC for each of the two noise levels are shown in Figures 6.19 and 6.20, respectively. The initial condition was $(x_0, y_0) = (1.03, -0.041)$. For the MC and PI simulations, a time step $\Delta t = 2\pi/(400 \cdot 0.32)$ was used. The PI simulation used FFT on a 256×256 grid, using cubic B-splines for the interpolation. The high grid resolution was needed to ensure that a few grid points were inside the ball of radius ρ. At the same time, this high grid resolution partly eliminates the need of removing artificial numerical oscillations, it suffices to truncating interpolated values at zero to avoid the PDF of being negative some places. A snapshot of the response PDFs for the two simulations after 12 periods are shown in Figure 6.21. Qualitatively, it would be easy to conclude that the noise level just reduces the image resolution by smearing out the finest structure. However, the system with $\gamma = 0.005$ has not reached stationarity. It has an extremely long transient, where probability mass gradually accumulates at the origin like the black holes in the universe. The (almost) stationary PDF after 30 periods is basically a spiked distribution in the origin with a small variance, but looking carefully, the structure seen in the left side of Figure 6.21 does not dissappear. The long transient is a challenge with PI. As the final result is generally unknown, only the difference between iterations can be studied. When convergence is slow, it is difficult to determine if parts of the structure is just part of the transient, or if the structure seen will ultimately be stable. Another warning is that determining the grid based on the MC simulation in this case could cause a grid just covering e.g. the ball of radius R. Then the possibility of entering the chaotic region, although this is very small, is ignored.

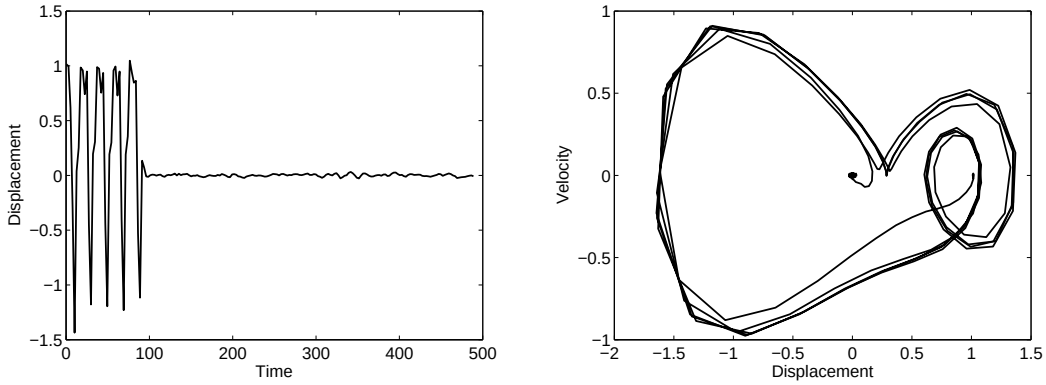


Figure 6.19: Monte Carlo realization of the system in equation (6.17) with noise level $\gamma = 0.005$. The left plot shows just the response in x, plotted against time, the second illustrates the realization's movement in the state space within the same timeframe.

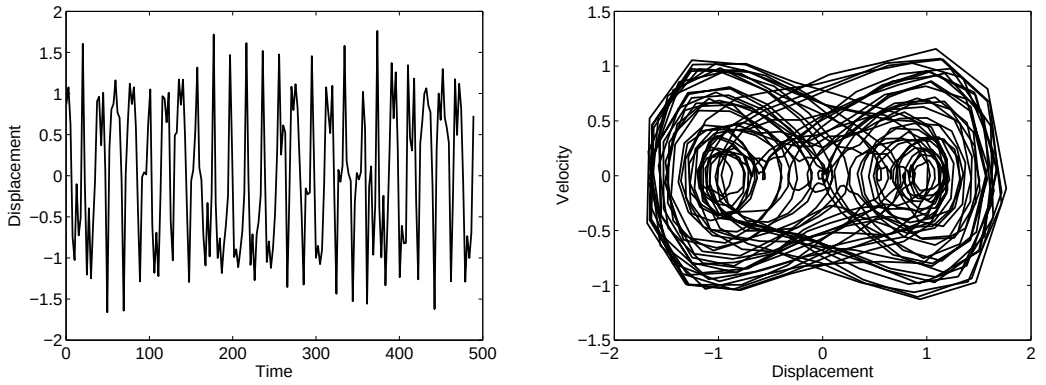


Figure 6.20: Monte Carlo realization of the system in equation (6.17) with noise level $\gamma = 0.05$, all other parameters are the same as in the previous figure.

As a final remark to this system, the correction factor in the integral from section 4.2 must be found. Due to the ζ weighting function from equation (6.47), this is not a constant, but becomes

$$|J_g| = \begin{cases} \exp(\alpha \Delta t), & \text{for } x^2 + y^2 < \rho^2, \\ \exp\left(\frac{\left[2y(\omega^2 x + \alpha y - a_2(x,y,t)) + 0.04(x^2+y^2-\rho^2) + \alpha(R^2 - x^2 - y^2)\right]\Delta t}{R^2 - \rho^2} \right), & \text{for } \rho^2 \leq x^2 + y^2 < R^2, \\ \exp(0.04 \Delta t), & \text{for } R^2 \leq x^2 + y^2. \end{cases} \tag{6.48}$$

Since the vector field is constructed to be continuous, there is no need for corrections at the boundaries of the inserted circles. With a more smooth weighting function than ζ from equation (6.47), equation (6.48) would become even more complicated and nonlinear.

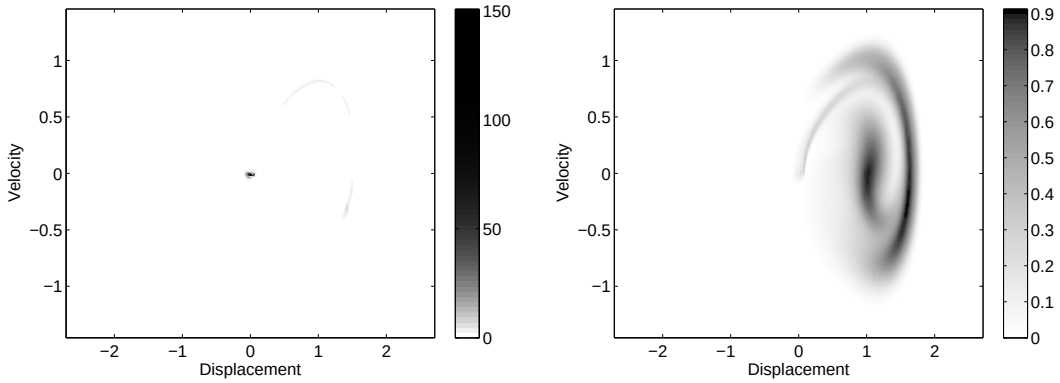


Figure 6.21: Snapshots of path integration solution of the system in equation (6.17). The left image for noise level $\gamma = 0.005$, the right image for $\gamma = 0.05$. First, the system was run for five periods with $\dot{Y} = a_2(X, Y, t)$, i.e. without the damped system around the origin, then 12 periods with the combined system.

6.18 Negative stocastic Lyapunov exponent caused by added noise

It is generally assumed that sufficient amounts of noise should cause the MLE to become negative for many chaotic systems. However, the author has not seen any general proof or requirements, and it does not seem to be intuitive. Such discussions is also a bit beyond the scope of this work. Instead, like in the previous section, an example is constructed where high noise levels should give a negative SLE, while the unperturbed system is chaotic.

Assume a system where the sensitivity to small perturbations is confined to some domain around the origin in the coordinate system of the state-space variables, and outside this domain, the system has the form of a damped linear oscillator. Such a system can simply be constructed by gluing together two different systems to an arbitrary smoothness. Say we use the chaotic Duffing equation discussed earlier, choose two radiuses $R > \rho$ such that the deterministic system never passes outside a ball in the state space centered in the origin with radius ρ, ($\rho = 10$ is sufficient), and write:

$$\begin{aligned}
\mathrm{d}X &= Y\,\mathrm{d}t, \\
\mathrm{d}Y &= \gamma\,\mathrm{d}W + \begin{cases} a_2(X,Y,t), & \text{for } X^2+Y^2 < \rho^2, \\ \zeta(X,Y)a_2(X,Y) - (1-\zeta(X,Y))(\omega^2 X + \alpha Y), & \text{for } \rho^2 \le X^2+Y^2 < R^2, \\ -(\omega^2 X + \alpha Y), & \text{for } R^2 \le X^2+Y^2, \end{cases} \\
a_2(x,y) &= \frac{1}{5}x - \frac{8}{15}x^3 - \frac{1}{25}y + \frac{2}{5}\cos(0.32t),
\end{aligned} \tag{6.49}$$

This again uses a crude linear overlap between the chaotic system and the damped oscillator, with the same weight function $\zeta(x, y)$ from equation (6.47) in the previous section.

However, the overlap can be constructed to be k times differentiable for any $k \in \mathbb{N}$.

As $\gamma \to \infty$, the noise will eventually push a significant part of the probability density out to the damped oscillator where Q is negative. At some point, the total variation due to noise will be much larger than R, much time will be spent oscillating around the chaotic domain without entering it, and the behaviour inside the radius R ball can be neglected. Then the PDF will approach a Gaussian shape with a standard deviation proportional to the noise level. As the PDF of the stochastic system increases its support in all directions, the increasing contributions from Q outside R must at some point make λ_1 in equation (6.33) negative.

If the individual realizations are easier to visualize, the following argument might help. Trajectories will diverge in the chaotic region. However, due to the noise, they are sometimes diverted out to the non-chaotic region due to the random perturbations, where they are brought closer together before re-entering to the chaotic region. If the time spent in the chaotic region typically is short, and they will on average be pushed together.

For the MC and PI simulations, $\alpha = 0.04$, $\omega^2 = 1/5$, $\rho^2 = 3.25$, $R^2 = 3.5$, and $\Delta t = 2\pi/(400 \cdot 0.32)$. With a sufficiently small noise level, e.g. $\gamma = 0.005$, the system in equation (6.49) has a behaviour indistinguishable from the pure chaotic Duffing oscillator. With $\gamma = 1.0$, the system is not chaotic any longer. MC realizations for this system are shown in Figures 6.22 and Figure 6.23 for $\gamma = 0.005$ and $\gamma = 1.0$, accordingly. The corresponding PI results are shown in Figure 6.24. For this system, it is not obvious that the system indeed is non-chaotic for the high noise level, but this is verified by calculating the MLE. Because the vector field is not differentiable everywhere and to save CPU-time, the curve length method is used. Results for the time derivative of the MLE, averaged over four different noise realizations, but with the same initial line segment, are shown in Figure 6.25. Careful investigation of the plot should convince that the time average for the highest noise is negative, which is also verified numerically.

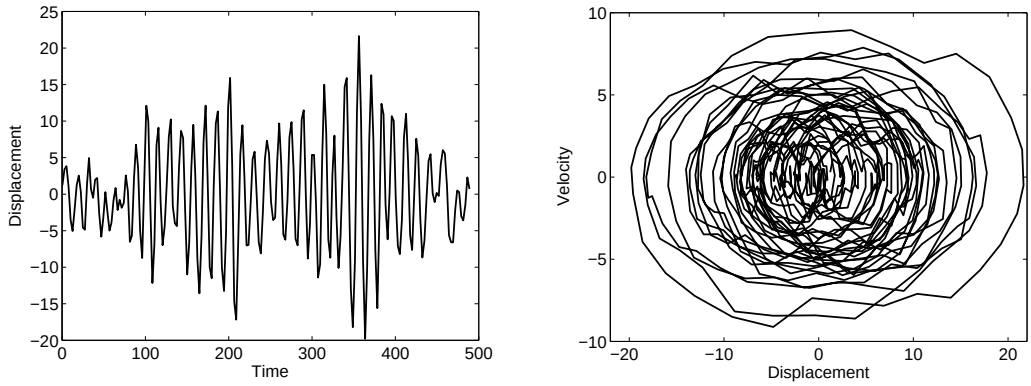


Figure 6.22: Monte Carlo realization of the system in equation (6.49) with noise level $\gamma = 0.005$. The left plot shows just the response in x, plotted against time, the right image illustrates the realization's movement in the state space within the same timeframe.

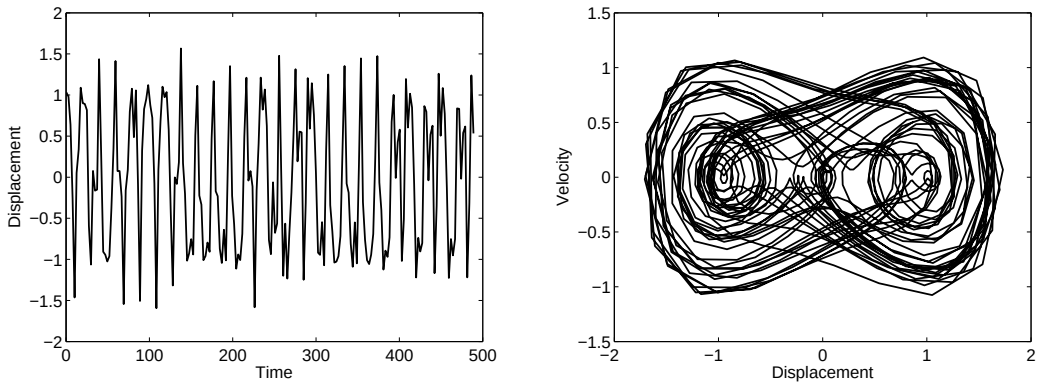


Figure 6.23: Monte Carlo realization of the system in equation (6.49) with noise level $\gamma = 1.0$, all other parameters are the same as in the previous figure.

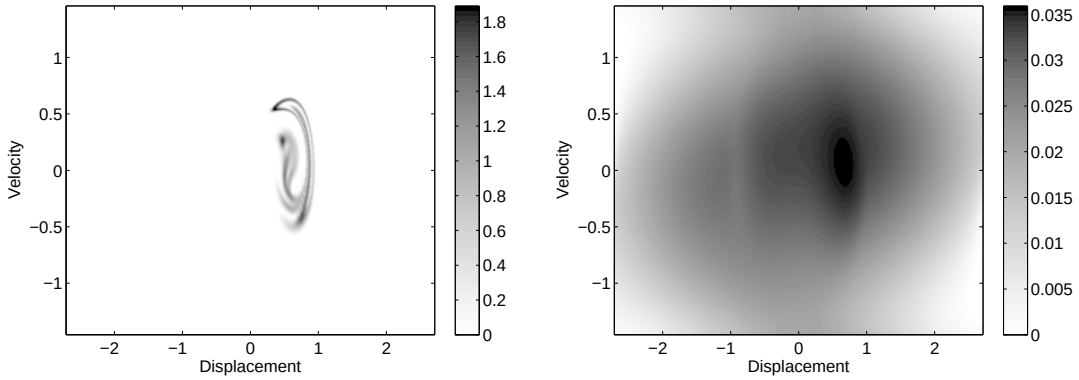


Figure 6.24: Snapshots of path integration solution of the system in equation (6.49). The left image for noise level $\gamma = 0.005$, the right image for $\gamma = 1.0$. First, the system was run for five periods with $\dot{Y} = a_2(X, Y, t)$, i.e. without the damped system around the origin, then 12 periods with the combined system.

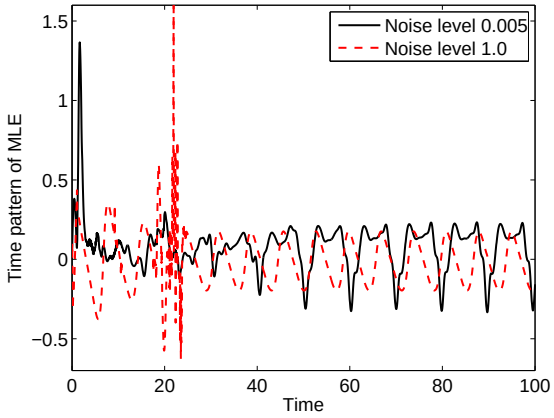


Figure 6.25: The time pattern for the MLE for equation (6.49) for the two noise levels $\gamma = 0.005$ and $\gamma = 1.0$. The average over one period for these two curves gives the MLE, which is positive for the first case, negative for the second. The curves are found using the propagated line segment approach.

Chapter 7

Summary and concluding remarks

The objective of this thesis has been twofold. One is to present some resent developments in PI and tries to give a more structured presentation of the method than earlier work, the other is to answer some specific questions regarding the possibility of studying highly nonlinear systems, and chaos in particular, with PI. The following two sections summarize these two perspectives in brief.

7.1 New developments

7.1.1 Improving the path integral

In cooperation with Professor Arvid Naess, the PI method has been refined and more properly described as a Markov operator.

Implementing a variable transformation to properly perform the path integral might not be new, but it has not been described earlier. This thesis explores this from a phenomenological, an algebraical, and a numerical perspective. The implementation of PI differs from earlier work in that the variable transformation is performed on all the state-space variables, which also proved to be important for the development of a more time-efficient implementation by FFT.

Through the implementation of PI by FFT, the CPU-time has been reduced by a factor of about 50^{N-r}, where N is the number of dimensions, and r the number of equations where noise does not enter directly. However, one must observe that there are additional requirements for this FFT-based method to work well within PI.

7.1.2 Improved interpolation

Numerical stability of PI has been a challenge mentioned by Johnsen [20], Moe [34], and Skaug [50], and probably seen by anyone trying to implement PI numerically. There are basically two different problems.

- One could see convergence to an incorrect, but seemingly fairly reasonable PDF.

- The PDF would attain very high or very small (negative) values in just a few grid points, typically near the boundaries of the domain where the PDF should be close to zero.

Linear interpolation does not possess the latter error, but typically under-estimates peaks, giving too high variance for mono-spaced PDFs. Cubic splines work well for some systems, but for other systems show the second behaviour. These problems have been observed and explained in this work, and several different approaches to reduce these erraneous effects have been proposed and tested. Earlier, the main stabilizing method was to exchange all negative interpolated values by zeroes, and filter the PDF after every iteration to remove the unlikely high delta-shaped spikes. This was unsatisfactory, especially since the cause of the problem was not well known, and neither the problem nor this work-around was documented. The method that most consistently avoids this problem is a new kind of interpolation scheme which combines Bézier curves and linear interpolation. This method retains the positive stability and non-negativity of the linear interpolant, while highly reducing the problem of systematic error in peaks and tails. However, the drawbacks are longer CPU-time and that this method cannot be represented as a linear matrix multiplication so convergence is not guaranteed and optimization is difficult or impossible. Another well-functioning method is based on using linear interpolation in just some intervals where the cubic or rectangular spline could be unstable. This also gives good results for most systems, it is faster, and in spite of not being a complete linear matrix interpolation, it allows for efficient programming. One could probably find many indicators for when the higher-order interpolant should be substituted by a straight line. Doing this for grid points where the spline coefficient is negative, and all its neighbours seems to work well, so this is recommended for future work.

A few other methods developed aiming to improve interpolation has been presented, namely conditional PI, tension splines, and rectangular B-splines. All of these are slightly less likely to cause the most dramatic interpolation error, but the improvement is not convincing. The downside is that the (iterative and total) interpolation error is higher than for cubic splines where the cubic spline is stable. Conditional PI and/or rectangular B-splines is recommended for high dimensional systems to reduce CPU-time, otherwise these methods have too little effect to be recommendable.

7.1.3 Extended usability

To adapt to some specific systems, PI has been implemented to handle signum-kind non-linearities, vibro-impact problems, and angular state-space variables. Further, PI can now handle noise in several variables, and solve the extended system obtained by including the Jacobian matrix. Finally, by implementing a programming tool for PI, different methods for time-stepping and interpolation can be combined arbitrarily, and efficient high-dimensional code can be generated, which should be of help for future work. This tool also includes binomial white noise to approximate the compound Poisson noise process with uniform or Gaussian increments, and multiplicative/parametric noise.

One proposal for future work to improve the PI method further has also been mentioned, denoted parametric PI. For at least some problems, it should be possible to partly represent the PDF by parametrized functions rather than just a discretized grid. Some simple trials with representing angular densities as ellipsoids look promising.

7.2 Studying chaos and nonlinear systems by PI

Basically three questions was raised when starting this project, based on a short study of a chaotic Duffing system by Naess and Skaug in 2002 [42]. One was simply if chaotic systems could be studied by adding noise, the second if, and in what sense, the results by PI could give insight in the system's behaviour, especially if a system could be classified as chaotic based on its PDF. This classification question was later modified to ask how far the Lyapunov spectrum, or especially the largest Lyapunov exponent, could be found with PI. These questions haven't been fully answered, but the main points are summarized below.

7.2.1 Adding noise to study a deterministic systems

With a range of different references to other work, it has been shown that adding noise could introduce qualitatively new effects to a system. This also includes transitions from chaotic to non-chaotic and vice versa. However, it seems that these transitions occur in very special systems, constructed to obtain this effect, or very gradually with increasing noise level. Hence, for most systems and most parameter values a low-level added noise should not cause problems.

Chaotic systems, and also some other systems, are sensitive to initial values. For a specific realization, the added noise would then cause a radical change in the response at a specific time instant. However, in the probability density setting, the added noise has little effect on the dynamics. So for a specific problem, one should not expect such transitions.

7.2.2 Qualitatively studying chaos by PI and resulting PDF

Both early and later simulations showed that the PDF does preserve a lot of the fine-level structure that many chaotic systems exhibit. Further, it was clear that the addition of noise had the effect of "smearing out" the structure, popularly illustrated as "throwing a stochastic blanket" over the deterministic response structure seen in Poincaré maps and averaged densities. Another effect that should be stressed, is that this "smearing out" also is caused by the smoothening of the interpolation method. For a specific study, one would have to use a very high grid resolution to avoid this effect, and the high resolution is needed in all the state-space variables. With the improvements of PI discussed above and the increase of computer speed and memory, high resolution simulation is now possible within acceptable time.

Simulations from 2003 [41], included in section 6.3, showed that the structure from the Poincaré maps of the deterministic system could also be found in the stationary PDF

of the SDE obtained by exchanging the harmonic forcing function by a stochastic linear filter equation. It was also shown how random phase could be implemented numerically, and that the structure is more resistant to high noise in the phase than high noise in the amplitude. Filter equations, noisy phase, and more complicated systems require more dimensions. Again, code up to six dimensions is now available, and with improved computer capacity also advanced systems should be possible to solve up to six dimensions with higher resolution.

In later studies, especially while working on the paper [40], included in appendix E, it has been shown that PI has a great advantage in calculating transient behaviour and periodic PDFs, especially compared to Monte-Carlo simulations.

7.2.3 Classification of chaotic SDEs by PI

Due to the existence of strange non-chaotic attractors, and because the PDF itself does not carry sufficient information about the underlying process, the top Lyapunov exponent is needed to classify chaos. This requires an extension of the state space, and currently also some additional numerical approximations. For the few systems studied here, PI seems to be able to estimate the MLE. However, simulation time is long, and the computer code is quite complicated, especially compared to standard methods based on realizations, especially Monte-Carlo methods. Therefore, using PI for calculating the Lyapunov exponent can only be recommended if the joint PDF in all state-space variables and space angle is of particular interest, e.g. for control problems or further analysis.

The extension of the state space will also, presumably for many years, be limited to six dimensions, unless the proposed parametric PI method or similar techniques could lift the curse of dimensionality.

Therefore, the main recommendation is that PI should be used for qualitative studies mainly, especially non-autonomous systems where the gain is especially high compared to Monte-Carlo methods, and that a lot of care needs to be put on the interpolation method and the path integral, in addition to ensuring a stable time-stepping method.

7.3 Future work

Many of the ideas and techniques described here are developed at the end of the PhD work. The studies are hence just preliminary, and many examples can be persued further. This especially includes high-dimensional path integration and calculation of the MLE, and detailed studies of specific systems. Some of the ideas presented here, like parametric PI, have not even been implemented - except for the very special case of ellipse shaped perturbation angles.

After this project, PI code is available in a general form through an internet interface, and some specialized code can be distributed by request. This should be of help for future research.

For a specific SDE, the choice of grid, time step, and interpolation method requires some insight in how the numerical algorithm works. So PI cannot be run completely as a black-box algorithm. However, with the work to develop very stable interpolation routines and automatic code generation, the author would think that also under-graduate students could do exercises with PI. There are a lot of possible improvements for making the code available, e.g. as library functions in numerical software.

Appendix A

Efficient path integration by FFT

Eirik Mo and Arvid Naess

Presented at the 10th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP10), Tokyo, 2007

Appendix B

Probability density and excursions of structural response to imperfectly periodic excitation

Mikhail F. Dimentberg and Eirik Mo and Arvid Naess

Journal of Engineering Mechanics 133 (2007), 1037–1041

Appendix C

Response probability density functions of strongly nonlinear systems by the path integration method

Daniil V. Iourtchenko and Eirik Mo and Arvid Naess

Int. J. Non-linear Mechanics 41 (2006), 693–705

Appendix D

Reliability of strongly nonlinear systems by the path integration method

Daniil V. Iourtchenko, Eirik Mo, and Arvid Naess

Submitted paper

Appendix E

Stochastic spur gear dynamics by numerical path integration

Arvid Naess, Finn Erik Kolnes, and Eirik Mo

Journal of Sound and Vibration 302 (2007), 936–950

Appendix F

Nonsmooth dynamics by path integration - an example of chaotic response of a meshing gear pair

Eirik Mo and Arvid Naess

Proceedings of the fifth conference on Computational Stochastic Mechanics, CSM-5
(2007), 481 – 485

Bibliography

[1] L. Arnold. *Random Dynamical Systems*. Springer, 1998.

[2] J. Banks, J. Brooks, G. Cairns, G. Davis, and P. Stacey. On Devaney's definition of chaos. *American Math. Monthly*, 99:332–334, 1992.

[3] G. Barone, G. Navarra, and A. Pirrotta. Stochastic dynamics of linear structures with nonlinear damper devices (PIS method). In *Proceedings of the fifth international conference on computational stochastic mechanics*, June 2006.

[4] R.H. Bartels, J.C. Beatty, and B.A. Barsky. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann Publishers, INC., 1987.

[5] P. Baxendale. Stability along trajectories at a stochastic bifurcation point. In H. Crauel and M. Gunlach, editors, *Stochastic Dynamics*, chapter 1, pages 1–25. Springer, 1999.

[6] J.H.E. Cartwright. Nonlinear stiffness, Lyapunov exponents, and attractor dimension. *Physics Letters A*, 264:298–304, 1999.

[7] J.H.E. Cartwright and O. Piro. The dynamics of Runge–Kutta methods. *Int. J. Bifurcation and Chaos*, 2:427–449, 1992.

[8] M. Grigoriu Ç. Kafali, E. Mo. Seismic fragility of linear systems subjected to non-Gaussian ground acceleration processes. In *Proceedings of the Eight U.S. National Conference on Earthquake Engineering, 8NCEE*, 2006.

[9] Y.F. Chang. *ATOMFT User Manual ver. 3.11*. Lake of the Woods, California, March 1993.

[10] C. de Boor. *A Practical Guide to Splines*, volume 27. Springer-Verlag New York INC., 1978.

[11] C. de Boor. Convergence of cubic spline interpolation with the not-a-knot condition. Technical report, University of Wisconsin-Madison, Mathematics Research Center, October 1985.

[12] B. Dennis, R.A. Desharnais, J.M. Cushing, S.M. Henson, and R.F. Costantino. Can noise induce chaos? *OIKOS*, 102:329–339, 2003.

[13] T.C. Gard. *Introduction to stochastic differential equations*. Monographs and textbooks in pure and applied mathematics 114. Marcel Dekker, New York, 1988.

[14] R. Gilmore and J.W.L. McCallum. Structure in the bifurcation diagram of the Duffing oscillator. *Physical Review E*, 51(2):935–956, 1995.

[15] J. Gleick. *Chaos*. Minerva, 1997.

[16] G.H. Golub and C.F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 3 edition, 1996.

[17] J. Hogan. Nonsmooth systems: Challenges and unsolved problems. Presented at the SIAM Conference on Applications of Dynamical Systems, Snowbird, Utah, May 2007.

[18] Ph. Holmes. A nonlinear oscillator with a strange attractor. *Phil. Trans. of the Royal Society, London*, A292:419–448, 1979.

[19] N. Ikeda and S. Watanabe. *Stochastic Differential Equations and Diffusion Processes*. North-Holland/Kodansha, 1981.

[20] J.M. Johnsen. *Response statistics of nonlinear dynamic systems*. PhD thesis, NTH, August 1992.

[21] À. Jorba and M. Zou. A software package for the numerical integration of odes by means of high-order taylor methods. Technical report, Universitat de Barcelona and The University of Texas at Austin, http://www.maia.ub.es/˜angel/taylor/, January 2004. Accessed June 2007.

[22] A. Kahraman and R. Singh. Non-linear dynamics of a spur gear pair. *Journal of Sound and Vibration*, 142(1):49–75, 1990.

[23] H.C. Karlsen. *Statistics of wave induced nonlinear loads and responses*. PhD thesis, Norwegian University of Science and Technology, September 2006.

[24] R.Z. Khas'minskii. Necessary and sufficient conditions for the asymptotic stability of linear stochastic systems. *Theory Probab. Appl.*, 12:144–147, 1967.

[25] Y. Kifer. *Random perturbations of dynamical systems*. Progress in probability and statistics 16. Birkhäuser, Boston, 1988.

[26] T.S. Kleppe. Numerical path integration for Lévy driven stochastic differential equations. Master's thesis, Department of Mathematical Sciences, Norwegian University of Science and Technology, June 2006.

[27] T. Klinker, W. Meyer-Ilse, and W. Lauterborn. Period doubling and chaotic behaviour in a driven Toda oscillator. *Physics Letters*, 101A(8):371–375, 1984.

[28] P.E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag, 1992.

[29] A. Lasota and M.C. Mackey. *Chaos, Fractals, and Noise: Stochastic aspects of dynamics*. Applied mathematical sciences. Springer-Verlag, New York, 2 edition, 1994.

[30] Y.H. Lim and D.C. Hamill. Problems of computing Lyapunov exponents in power electronics. *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems*, 5:297–301, 1999.

[31] A.C.J. Luo. Grazing and chaos in a periodically forced piecewise linear system. *Journal of Vibration and Acoustics*, 128:28–34, 2006.

[32] E. Mo and A. Naess. Efficient path integration by FFT. In *Proceedings 10th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP10)*, Tokyo, Japan, 2007. Included in the book as appendix A.

[33] E. Mo and A. Naess. Nonsmooth dynamics by path integration - an example of chaotic response of a meshing gear pair. In G. Deodatis and P.D. Spanos, editors, *Computational Stochastic Mechanics*, pages 481 – 485. CSM - 5, 2006, 2007. Included in the book as appendix F.

[34] V. Moe. *Nonlinear random vibrations - numerical analysis by path integration methods*. PhD thesis, Norwegian University of Science and Technology, January 1997.

[35] D.M. Monro. Algorithm AS 83. complex discrete fast Fourier transform (in Statistical Algorithms). *Appl. Statist.*, 24(1):268–272, 1975.

[36] D.M. Monro. Algorithm AS 97. real discrete fast Fourier transform. *Appl. Statist.*, 25(2):166–172, 1976.

[37] R.E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, N.J., 1966.

[38] D.E. Musielak, Z.E. Musielak, and J.W. Benner. Chaos and routes to chaos in coupled Duffing oscillators with multiple degrees of freedom. *Chaos, Solitons and Fractals*, 24:907–922, 2005.

[39] A. Naess. Chaos and nonlinear stochastic dynamics. *Probabilistc Engineering Mechanics*, 15:37–47, 2000.

[40] A. Naess, F.E. Kolnes, and E. Mo. Stochastic spur gear dynamics by numerical path integration. *Journal of Sound and Vibration*, 302:936–950, 2007. Included in the book as appendix E.

[41] A. Naess and E. Mo. A numerical study of the existence and stability of some chaotic attractors by path integration. In *Proceedings*. GAMM, 2004.

[42] A. Naess and C. Skaug. The study of chaotic attractors of nonlinear systems by path integration. In N.S. Namachicivaya and Y.K. Lin, editors, *Proceedings of the IUTAM Symposium on Nonlinear Stochastic Dynamics, Monticello, Illinois, USA, August 2002*, pages 445–454. Kluwer Academic Publishers, Dordrecht, 2002.

[43] M. Noori, H. Davoodi, and A. Saffar. An Itô-based general approximation method for random vibration of hysteretic systems. part i: Gaussian analysis. *Journal of Sound and Vibration*, 127(2):331–342, 1988.

[44] B.K. Øksendal. *Stochastic Differential Equations; an introduction with applications*. Springer-Verlag, New York, 6th edition, 2003.

[45] J.C. Sommerer E. Ott and C. Grebogi. Scaling law for characteristic times of noise-induced crises. *Physical Review A*, 43(4):1754–1769, 1991.

[46] L.B. Rall. *Automatic Differentiation: Techniques and Applications*, volume 120 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, 1981.

[47] G. Rangarajan, S. Habib, and R.D. Ryne. Lyapunov exponents without rescaling and reorthogonalization. *Physical Review Letters*, 80:3747–3750, 1998.

[48] I. Scheuring and G. Domokos. Only noise can induce chaos in discrete populations. *Oikos*, 116(3):361 – 366, 2007.

[49] R. Seydel. *From Equilibrium to chaos - Practical bifurcation and stability analysis*. Elsevier Science Publishing Co., Inc., 1988.

[50] C. Skaug. *Random Vibration and the Path Integration Method*. PhD thesis, Norwegian University of Science and Technology, February 2000.

[51] J.Q. Sun and C.S. Hsu. The generalized cell mapping method in nonlinear random vibration based upon short-time Gaussian approximation. *Transactions of the ASME*, 57:1018–1025, december 1990.

[52] J.M.T. Thompson and H.B. Stewart. *Nonlinear Dynamics and Chaos*. Chichester : Wiley, 2 edition, 2002.

[53] S. Ulam. *Problems in modern mathematics*. Interscience Publishers, 1960.

[54] W.V. Wedig. Invariant measures and Lyapunov exponents for generalized parameter fluctuations. *Struct. Safety*, 8:13–25, 1990.

[55] V. Wihstutz. Perturbation methods for Lyapunov exponents. In H. Crauel and M. Gunlach, editors, *Stochastic Dynamics*, chapter 9, pages 209–239. Springer, 1999.

[56] E. Wilson, M. Homer, P.T. Piiroinen, and J.F. Mason. An analysis of attractors in mathematical models of vacuum pumps. Presented at the SIAM Conference on Applications of Dynamical Systems, Snowbird, Utah, May 2007.

[57] A. Wolf, J.B. Swift, H.L. Swinney, and J.A Vastano. Determining Lyapunov exponents from a time series. *Physica D*, 16(3):285 – 317, 1985.

[58] W.X. Xie, W. Xu, and L. Cai. Path integration of the Duffing-Rayleigh oscillator subject to harmonic and stochastic excitations. *Applied Mathematics and Computation*, 171:870–884, 2005.

[59] T. Yalçınkaya and Y.-C. Lai. Bifurcation to strange nonchaotic attractors. *Physical Review E*, 56(2):1623–1630, August 1997.

[60] J.S. Yu, G.Q. Cai, and Y.K. Lin. A new path integration procedure based on Gauss-Legendre scheme. *International Journal of Non-Linear Mechanics*, 32(4):759–768, 1997.

[61] J.S. Yu and Y.K. Lin. Numerical path integration of a non-homogeneous Markov process. *International Journal of Non-Linear Mechanics*, 39:1493–1500, 2004.