Technische Universität München

Fakultät für Mathematik

Lehrstuhl für Informatik mit Schwerpunkt
Wissenschaftliches Rechnen

Data-Driven Surrogate Models for Dynamical Systems

Felix Dietrich

Vollständiger Abdruck der von der Fakultät für Mathematik der Technischen Universität München zur Erlangung des Akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende:		Prof. Dr. Barbara Wohlmuth
Prüfer der Dissertation:	1.	Prof. Dr. Hans-Joachim Bungartz
	2.	Prof. Dr. Gerta Köster

Die Dissertation wurde am 13.04.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Mathematik am 06.07.2017 angenommen.

Acknowledgements

Many people all over the world have supported me during the last years, and I sincerely thank all of you here. Prof. Dr. Hans-Joachim Bungartz, my primary thesis advisor, has been a true inspiration mainly through his ability to lead by example. Our numerous discussions have encouraged me to start the dissertation in the first place, and have also pushed me towards university politics, an experience that I am particularly grateful for. Prof. Dr. Gerta Köster has shown me how to gracefully join the joy of research with the desire to create something useful. Her curiosity inspires all around her, and I have been very lucky to be a part of her great team. There, I have had the great pleasure to work with my colleagues, Dr. Michael Seitz, Dr. Isabella von Sivers, and Benedikt Zönnchen, as well as many students. In the last two years of my dissertation, I have been incredibly lucky to meet and work with Professor Yannis Kevrekidis. With a seemingly endless number of ideas, great food, and the desire to go beyond what is known, he continuously surprises and amazes me.

In the TopMath program, Agnieszka Baumgärtel, Dr. Carl-Friedrich Kreiner, and Prof. Dr. Martin Brokate have been invaluable for their help and guidance, mostly with matters beyond mathematics.

The last years have been an exciting journey—into science, and also in my personal life. This is mostly due to Katharina, constantly providing love and happiness, and also many interesting insights into tradition and society. Maika, Benedikt, and Christoph have played a large role here, too, and I am honored to have them as friends. Finally, I want to thank my parents and my brother for their love and trust in me.

Abstract

Processes in nature occur on a multitude of temporal and spatial scales. Mathematical models can capture many important parts of these processes, and are hence one of the primary tools of understanding in the natural sciences. A model usually cannot represent the process on all scales. Granular matter, self-propelled particle systems, and multi-agent systems are often modeled on a fine scale, where grains, particles, or individuals can be distinguished. The challenge in this case is to set up the model on the fine scale correctly, such that behavior on a coarser, macroscopic scale emerges naturally from the individual interactions. Then, computer simulations of the model on the fine scale can be used to predict behavior on the macroscopic scale. After the model on the microscopic scale is complete, the new challenge is to find the model on the macroscopic scale that can reproduce the same observations. A large number of numerical methods addressing this challenge have gained attention recently. Many methods rely on a formulaic description of the process on the microscopic scale, and cannot cope with a description purely in the form of a simulation software. Data-driven methods are a good approach here. Most data-driven methods rely on the observations from the microscopic system to have enough information to compute future values. Even if this requirement is fulfilled, the methods suffer from the curse of dimension when many parameters are varied in the creation of the observation data.

In this dissertation, we introduce, analyze, and apply the concept of data-driven, dynamic surrogate models. These surrogate models capture the dynamical process on the macroscopic scale, but are computed from observations generated through models defined on the microscopic scale. We assume existence of a system on the macroscopic scale, which is able to generate the observations. Delay embedding of these observation values generates a diffeomorphic copy of the macroscopic system we assume to exist. We proof that the output of the data-driven surrogate model is approximating the observations of the original system, and show how the approximation error depends on the approximation method. We also show storage of data for the surrogate model is efficient if the number of intrinsic variables are less than or equal to the number of parameters we vary when generating the data. The concept of the data-driven, dynamic surrogate model is demonstrated in five applications, namely bottleneck and queuing behavior of crowds, uncertainty quantification for the evacuation of a train, car traffic on highways, and the flow of granular particles in a silo. For all applications, we discuss state of the art approaches without the surrogate model, and the benefits and caveats of using a surrogate. Benefits include efficient storage, fast computation of observations on the macroscopic scale, insights into the topology of the underlying macroscopic system, and a generic algorithm usable for many different systems and scenarios.

Zusammenfassung

Prozesse in der Natur finden auf einer Vielzahl von zeitlichen und räumlichen Skalen statt. Mathematische Modelle können viele wichtige Teile dieser Prozesse abbilden, und sind somit eins der Hauptwerkzeuge der Naturwissenschaften. Ein Modell kann normalerweise einen Prozess nicht auf allen Skalen abbilden. Körnige Materie, sich selbständig bewegende Partikelsysteme sowie Multiagentensysteme werden oft auf einer feinen Skala modelliert, auf der Körner, Partikel oder Individuen unterschieden werden können. Hierbei liegt die Herausforderung in der korrekten Modellierung auf der feinen Skala, so dass das Verhalten auf einer gröberen, makroskopischen Skala ohne zusätzliche Modellierung aus den einzelnen Interaktionen auf der feineren, mikroskopischen Skala entsteht. Gelingt dies, können Computersimulationen des Modells auf der feinen Skala zur Vorhersage des Verhaltens auf der makroskopischen Skala genutzt werden. Die neue Herausforderung ist dann, für die makroskopische Skala ebenfalls ein sinnvolles Modell zu finden, welches die Beobachtungen und Vorhersagen ebenfalls ermöglicht. Eine große Zahl moderner numerischer Methoden versucht dieser Herausforderung Herr zu werden. Methoden, die sich nur auf eine Formeldarstellung des mikroskopischen Modells stützen, können nicht mit einer Beschreibung umgehen, die nur über eine Simulationssoftware gegeben ist. In diesem Fall sind datengetriebene Methoden eine gute Wahl.

Viele datengetriebene Methoden wiederum setzen voraus, dass die makroskopischen Beobachtungen des mikroskopischen Modells genug Information enthalten, um zukünftige Werte zu berechen. Eine weitere Schwierigkeit datengetriebener Methoden ist der Fluch der Dimensionalität, also des exponentiellen Anstiegs des nötigen Datenvolumens beim linearen Anstieg der Anzahl Parametern.

In dieser Dissertation wird das Konzept der dynamischen, datengetriebenen Ersatzmodelle eingeführt, analysiert und auf mehrere Beispiele angewendet. Diese Ersatzmodelle erfassen den dynamischen Prozess auf der makroskopischen Skala, und werden über Beobachtungen erstellt, die vom mikroskopischen Modell stammen. Dazu wird die Existenz eines makroskopischen Modells angenommen, welches die gleichen Beobachtungsdaten generieren kann; dieses Modell wird aber nie explizit formal aufgestellt. Eine Einbettung von Beobachtungen über mehrere Zeitschritte erzeugt eine diffeomorphe Kopie des Zustandsraums des makroskopischen Modells. Diese Tatsache wurde im Satz von Takens schon vor längerer Zeit formuliert, und wird hier genutzt, um die korrekte Approximation des makroskopischen Systems durch das Ersatzmodell zu garantieren. Ebenso wird der numerische Approximationsfehler sowie der Speicherverbrauch durch das Modell analysiert. Das Ersatzmodellkonzept wird in fünf Anwendungen demonstriert, die alle einen Prozess in der Natur beschreiben: das Verhalten an Engstellen und in Anstehsituationen von Menschen, der Quantifizierung von Unsicherheiten bei einer Zugevakuierung, dem Autoverkehr auf einer Schnellstraße, sowie granularem Fluss in einem Silo. Bei allen Anwendungen werden die Vor- und Nachteile des Ersatzmodells im Vergleich mit Standardansätzen beschrieben. Zu den Vorteilen gehören vor Allem effiziente Speichernutzung, schnelle Berechnung von makroskopischen Beobachtungsgrößen, Einblicke in die Topologie des zugrundeliegenden makroskopischen Systems, sowie ein generisch einsetzbarer Algorithmus für viele verschiedene Systeme und Szenarios.

Contents

Chapter 1

Introduction

The main research question addressed in this thesis is:

How can a scale transition from microscopic to macroscopic models be achieved through a data-driven procedure, when the macroscopic equation is not known, but is assumed to exist?

All relevant terms are defined and explained in the next sections. The question is answered by the definition and analysis of data-driven, dynamic surrogate models. Applications include crowd dynamics and car traffic, as well as flow of granular particles in a silo. Expanding on the research question, the key contributions of this thesis are:

- A concise description of particle systems, integrating crowd dynamics, car traffic, and systems of granular matter into the framework of multiscale dynamical systems. This bridges a gap between problems in safety engineering and the mathematical sciences for dynamical systems and machine learning.

- The definition and analysis of a *dynamic surrogate model* on a manifold in time-lagged observation space. With this surrogate model, it is possible to generate observations of a microscopic system over time, on a coarser, macroscopic scale.

- Numerical analysis of the data-driven surrogate model, and application to several scenarios involving particle systems: a bottleneck and a queuing scenario as well as uncertainty quantification of crowd dynamics, car traffic on a highway, and the flow of granular particles in a silo.

1.1 For the hurried reader

The thesis is divided into sections with mathematical topics developing the theory for dynamic surrogate models, and sections on the applications of the theory. The start of each section in the main chapter (3) uses a specific aspect of two example systems, a spiral system and a bottleneck in crowd dynamics, to motivate the ideas and arguments that follow. The hurried reader should consider the introduction to multiscale systems (1.2), a short introduction to crowd dynamics and granular flow (1.3), the introduction to the section on scale transitions with surrogate models and its motivation (3.1), as well as the discussion of the results (5). The end of all chapters includes a summary. A reader interested only in the theoretical aspects of the thesis is referred to the section on multiscale systems (1.2), the construction process of the data-driven surrogate model as the main contribution of the thesis (3) and its analysis (3.2). Five applications of the data-driven surrogate models are described in Chapter 4, which can be read independently of the chapter on theory. Chapter 4 includes bottleneck and queuing scenarios in crowd dynamics, uncertainty quantification of the deboarding process at a train station, stop-and-go waves in car traffic on highways, and upscaling of a granular flow system in a silo.

1.2 Systems with multiple scales

Studying the multiscale nature of systems in science and engineering has led to many success stories, both in theory and application. The world seems inherently multiscale—at least to understand complex systems holistically, elements have to be grouped together on larger and larger scales. What is a scale? An informal definition is possible through consideration of spatial and temporal scales. A spatial scale is best described through the measurement tools needed to capture it adequately, namely the size of rulers, measuring rods—or scales used to measure distances in the given space, on the given scale. Measuring the width of an atom with a ruler for architectural drawings does not make sense, neither does looking at geography with the precision of an electron microscope. The same holds for temporal scales: switching parts of cellular molecules often occurs on temporal scales far below microseconds and needs very precise instruments to observe, whereas the current day of the year could already be measured with instruments thousands of years ago. All dynamic processes in nature happen on one or more temporal and spatial scales, and scientific descriptions, called models, must incorporate this.

Models are the basic tools to foster understanding of systems in the natural sciences. A model is an abstraction of reality such that key features of the system at hand can be reproduced well, whereas others are neglected to simplify the model's description (Frigg and Hartmann, 2012; Bungartz et al., 2014). When it comes to capturing a system with a hierarchy of interacting scales, models with the same structure are often found to be superior to models with only one scale. The work of Shalizi (2006) provides a broad overview about the methods and techniques of complex systems science, including multiscale systems.

If a multiscale model has two scales, the scales are often called *microscopic* (or fast) and *macroscopic* (or slow). Models with scales in between the two are sometimes called *mesoscopic* models. We choose the flow of water particles in a river as an example. This system could be captured by a purely microscopic model of individual water molecules, which interact with their neighbors and are subject to gravity. The same system can be described macroscopically, through the flow of a certain volume of water over time. A multiscale model would take into account both scales: the water molecules are modeled explicitly and give information about the local flow to the macroscopic scale, where it is integrated into the global flow of volume. Li et al. (2004) review multiscale methodology for complex systems, and E (2011) provides an overview of numerical multiscale methods.

There is another challenge to multiscale modeling: given a description of a system on a certain scale, how to find a description on another, coarser scale? This challenge is called homogenization (Stuart and Humphries, 1996; Givon, Kupferman, and Stuart, 2004) or upscaling (Farmer, 2002; Brandt, 2005), and is the main question addressed and answered in this thesis.

In the example of flowing water, the challenge is to bridge spatial and temporal scales of many magnitudes. Models of continuous flow were formed successfully by upscaling systems analytically, usually in the limit of an infinite number of particles. This was in part possible by the particles being simple molecules, with the same form and "behavior". This allows for strong assumptions on averages, which ultimately yield the macroscopic dynamic (Legoll and Lelièvre, 2010).

Generally, the more complex the individual particle, the more complex the behavior of a system with more of these particles. The overall number of particles in the system can also matter greatly when considering homogenization. In granular flow, the sizes of individual particles can vary greatly, and the number of particles drops from approximately 1×10^{22} molecules in a gram of water to only a few millions or even thousands in a silo filled with grains. In this case, the assumption of equality of particles seizes to hold, and averaging produces results different from the actual process. The complexity of individuals can also increase when they are self-propelled, see the work of Helbing (2001) and Carrillo, Martin, and Panferov (2013) for reviews.

In car traffic, the individual particles are human drivers in their vehicles (Bellomo and Dogbe, 2011). They are very complex individually, but generally all abide the rules of traffic. Additionally, a car has very limited degrees of freedom when driven properly on a road. Cars are quite similar in speed and size compared to the size differences of rocks in avalanches, and could hence be studied using tools from granular flow. The complexity of human drivers, and the fact that cars are self-propelled and not force-driven particles, makes car traffic an active area of research on its own.

In crowd dynamics, the number of people moving is often comparable to the number of cars on a highway. However, pedestrians have more freedom to move than cars on roads, and change their direction of movement much quicker. At the same time, challenges introduced by individual complexity remain. Thus, the scale transition from micro to macro in crowd dynamics is even more challenging than for car traffic. Another major challenge, not only for upscaling, is the lack of a general purpose, microscopic model. This is due to the intrinsic complexity of humans, with current models incorporating not only physical but also psycho-social effects. Comparing recent microscopic and macroscopic models, Duives, Daamen, and Hoogendoorn (2013) even state that “for practical applications, that need both precision and speed, the current pedestrian simulation models are inadequate”. The multiscale nature of crowds has already been recognized in psychology, with the advance of theories such as social identity (Turner et al., 1987; Reicher, Spears, and Haslam, 2010). In short, and restricted to a crowd that is present physically, the theory explains crowd behavior macroscopically through the formation of different social groups. Individuals identify with one of the social groups, and act according to the norms of this group. The introduction of the social groups in addition to individuals makes social identity theory effectively a multiscale model of crowds. The complexity of humans results in a large number of parameters necessary to describe individuals. The number of parameters must then be multiplied thousands of times for large crowds. Most modeling attempts overcome this by using the same parameters for all individuals involved. This is in sharp contrast to homogeneous, molecular particle systems, where the repulsion and attraction potentials can be set with only a few parameters, and are the same for all particles. Due to the large number of challenges in crowd dynamics, many scientific fields are involved in its research: from mathematics (Francescoa et al., 2011; Degond et al., 2013), physics (Helbing and Molnár, 1995; Karamouzas, Skinner, and Guy, 2014), biology (Smith et al., 2007; Moussaïd et al., 2012), computer science (Richmond and Romano, 2008; Sud et al., 2008), engineering and safety science (Smith et al., 2009; Sivers et al., 2016), to psychology and sociology (Sime, 1995; Drury and Reicher, 2010).

Generally, observations of complex systems, followed by modeling, simulation, and analysis, can lead to predictions performed by computers (Sacks et al., 1989). Computational challenges in the context of crowd dynamics are (faster than) real time simulations (Richmond and Romano, 2008; Mroz, Was, and Topa, 2014), and uncertainty quantification (Iaccarino, 2008; Smith, 2014). The concept of data-driven surrogate models developed in this thesis helps to resolve the issue of real time simulations of macroscopic data, and also enable real time uncertainty quantification.

Finding the macroscopic equations of a system in closed form would ease the computational burden, and there are already great successes for classic physical systems with a large number of particles. Data-driven upscaling can help pave the way to understanding, and numerical algorithms already incorporate multiscale ideas very successfully.

The thesis combines results from manifold learning and the theory on dynamical systems with state-of-the-art microscopic models in granular systems, such as crowd dynamics, car traffic, and a generic granular flow model. The combination is a surrogate model approach to upscaling, where the microscopic model generates the data needed to learn the macroscopic model, effectively performing a scale transition through focusing on the observables that change the slowest.

1.3 Modeling crowd dynamics and granular flow

This section develops an understanding for the challenges in crowd dynamics compared to granular flow systems in physics. The state of the art for multiscale modeling and analysis is highlighted.

Individual human behavior is often difficult to understand. Why do we like certain people or things? Why do we choose to go left, instead of right? Why do we use the right foot to start walking, instead of the left? Often, a person cannot tell in hindsight why exactly they acted in a particular fashion. Understanding behavior as an outsider is even more challenging, mostly because limited or no information about the subject's thoughts is available. The challenge: understanding the internal processes and behavior of a system is the key to accurate predictions. How can human behavior be predicted, if it is so difficult to understand it on an individual level?

In physics, there is a similar problem with the prediction of motion for gases and fluids. Individual motion of atomic particles is impossible to predict exactly, since quantum mechanical effects introduce true randomness. Nevertheless, results from gas kinetics give hope for the case where many particles interact and only the collective, aggregated motion of the gas or fluid is of interest. In this case, the individual contributions of the particles to the global system are only important in an averaged or homogenized sense. In many cases, this averaging and homogenization produces systems that can be understood and even predicted on a much larger scale. Many systems studied in physics contain force-driven particles. Here, the motion of the particles is determined by intra- and inter-particle forces on the atomic level (weak interaction), the electro-magnetic level (electro-static and van der Waals forces), and even on the gravitational level (for example, the gravity field of the earth). Since the motion is not determined by an inner motivation as in humans, the motion of the particles can be determined completely by the forces acting between them.

This consideration of interactions has led to very deep mathematical theories. The theory of Hamiltonian systems, for example, explains and predicts motion solely by the position and velocity of all particles in the system. The theorem of Liouville relates the microscopic, individual and by their quantum nature also stochastic forces between particles to the deterministic motion of the probability density of their aggregated positions. This allows to ignore individual motion and to predict the motion of the system on a larger scale.

For an aggregation of humans—called physical crowd—similar assumptions on the importance of interactions are made, which form the basis for research of crowd dynamics. However, human interactions and behavior are much more complex than interactions between atoms and molecules. Therefore, the study of systems comprised of humans and animals does not yet have many successful theories such as the theorems of Liouville and Hamilton. In addition, swarms of fish, herds of sheep, or crowds of humans are systems of self-driven particles, and are therefore more difficult to study on an aggregated level. Aoki (1982) and Reynolds (1987) published the first results on computer simulations of self-propelled particles, and Vicsek et al. (1995), Czirók, Barabási, and Vicsek (1999), and Nagai et al. (2015) worked on upscaling these systems. Herbert-Read (2016) reviews work on collective motion in animals emerging from individual interactions, and Sumpter, Mann, and Perna (2012) study the modeling cycle of these systems. The considerations of humans as grains in granular flow is a crude approximation, and is able to predict crowd motion only in very dense situations, where persons cannot move against the flow. The self-propelled nature of individuals, and non-local, multiscale effects such as social behavior and psychology, add complexity on the microscopic scale that is not present in granular material. Still, many models in crowd dynamics are inspired by ideas from granular physics such as attraction and repulsion of particles. The most prominent examples are social force models (Helbing and Molnár, 1995; Helbing, Farkas, and Vicsek, 2000). Cristiani, Piccoli, and Tosin (2011) use a measure theoretic approach to combine ideas from granular flow and crowd dynamics in a multiscale model.

The mathematical theories for force-driven particles often assume that motion is determined by the interactions of the particles alone – which is reasonable if only the forces between them can change their paths. In contrast to that, self-propelled particles can change their paths completely independent of others. From an outside point of view, this independence breaks the first of the fundamental laws of Newtonian physics: the law of inertia, where a particle will keep its speed and direction constant if no external forces act on it. If we consider a self-propelled particle as seen by an observer, the particle can change its path without external influence. The key in this case is that the particle can exert a force on itself, which in turn changes its trajectory.

Many of the challenges in understanding granular flow media are easier when the objects of research are not self-propelled. One of these challenges is the performance of experiments. With molecular particles, most experiments can be done in vitro—“within the glass”, meaning in a laboratory—with many different experimental runs and precisely controlled experimental settings. In contrast to that, controlled experiments with humans are very time consuming and very costly. Even in case of a large number of experimental test subjects, the number of experimental runs is very limited (Boltes and Seyfried, 2013). When performing many experiments successively, extrinsic factors such as fatigue, hunger and adjustment to the experimental settings have to be taken into account, which also limits the number of experiments that can be run in total. Last but not least, the safety of the test subjects has to be ensured by the experimenter. This essentially forbids to perform experiments with high-pressure situations or fire, which on the other hand are very important aspects of safety research. In vivo experiments—“within the living”, meaning in a natural setting—that can be precisely controlled are even rare for molecular particles. Such experiments with humans are called “field experiments” or “field studies” (Zinke, Hofinger, and Künzer, 2013) and at the point of writing there is not much high-quality data available. Brŝĉić, Zanlungo, and Kanda (2014) provide an example of a valuable attempt, with trajectory data gathered over a whole year, over a small area of a stairway. In addition to the difficulty to control external factors, many countries have strict data privacy laws that forbid excessive video capturing of humans. This further complicates the gathering of data in the field, which actually would have to be much more excessive because of the numerous external factors. Nevertheless, qualitative information can be gathered even from small field studies, and is already being used to validate models for psycho-social behavior (Sivers et al., 2016).

In conclusion, studying humans experimentally requires a large effort, and trying to accurately predict the action a given human is performing next is often impossible, even for the next few seconds. Nevertheless, similar to the situation in physics, predictions are indeed possible in many scenarios for a whole crowd of people. Helbing, Keltsch, and Molnár (1997) predict optimal paths in an urban environment using a microscopic model, and Davidich and Köster (2013) show that real-life data is necessary for the calibration of the models to be predictive. The same seems to be true for an average of the behavior of many individuals over many scenarios (space average) or a longer time period (time average). In most cases, predictions in crowd dynamics are performed via models of human behavior.

Models for the dynamics of a crowd can be separated into classes, through the temporal and spatial scales the models operate on. The coarsest separation is often referred to as *microscopic*, *macroscopic*, and *multiscale* or *mesoscopic* models (Chraibi, 2012; Bellomo and Bellouquid, 2015). The scale differences between microscopic and macroscopic models are in the temporal and spatial domain, and often differ by two or three orders of magnitude. Table 1.1 shows the scales commonly used by the different model classes for crowd dynamics, as well as representative models for each class.

Table 1.1: Different model classes and representatives in crowd dynamics.

Class	**Temporal scale** $[s]$	**Spatial scale** $[m]$	**Representative model**
Microscopic	0.1–1	0.1–1	Social forces (Helbing and Molnár, 1995)
Macroscopic	1–100	1–10	Mass transport (Hughes, 2001)
Multiscale	0.1–100	0.1–10	Measure-theoretic (Cristiani, Piccoli, and Tosin, 2014)

Microscopic models describe the motion of the crowd by explicitly describing the behavior of each individual. A prominent example are social force models (Helbing and Molnár, 1995; Helbing et al., 2001; Chraibi et al., 2011), more recent approaches focus on the velocity of pedestrians (Dietrich and Köster, 2014), and the actual stepping process (Seitz and Köster, 2012; Sivers and Köster, 2015). Cellular automata usually operate on a slightly coarser spatial scale, but most automata still model each individual (Burstedde et al., 2001; Nishinari, Fukui, and Schadschneider, 2004) and therefore are also microscopic models. See the work of Dietrich et al. (2014) for a comparison of the two approaches.

Macroscopic models describe the crowd as a whole, where properties and features of individuals normally cannot be distinguished. In many cases, the density of the crowd is used as a macroscopic variable that changes over time. The evolution is then modeled as a transport or transport-diffusion process (Hughes, 2001; Bellomo, Piccoli, and Tosin, 2012).

The multiscale approach combines microscopic and macroscopic models into one model. This combination is often done by modeling individuals influenced by their neighbors and, in addition, all other individuals, the crowd, which is also modeled explicitly as in a macroscopic model. The motion of the macroscopic crowd is in turn also influenced by the behavior on the individual level (Cristiani, Piccoli, and Tosin (2011) and Cristiani, Piccoli, and Tosin (2014) discuss models in this category). A different approach is the combination of models with different scales in separate regions of the scenario (Biedermann et al., 2014) or for different navigational purposes (Kneidl, Hartmann, and Borrmann, 2013).

1.4 Summary

We discussed systems with multiple temporal and spatial scales. These systems are ubiquitous in nature, in particular, systems comprised of many particles often exhibit dynamics on a macroscopic scale very different from the microscopic, individual behavior. Expanding on particle systems, we discussed crowd dynamics, car traffic, and granular flow. These systems are composed of individuals or grains that are complex themselves. This inherent complexity leads to more intricate system behavior on the macroscopic scale, which makes the system a topic of active research.

We provided examples for applications where the macroscopic dynamic of the systems with complex particles are more important than individual behavior. This justified the search for a transition of scales from the microscopic to the macroscopic model. We argued that approaches from traditional physics are difficult to apply, because of core assumptions regarding uniformity or simplicity of the particles. An orthogonal challenge we discussed is the rapid development of the simulation software for systems with complex particles, because methods that rely on a formulaic description cannot cope with black-box simulator software. This led to the need for data-driven methods able to extract macroscopic models directly from data.

Chapter 2

Numerical analysis of dynamical systems

The numerical analysis of dynamical systems is a rich scientific field, both in mathematics and computer science. Easy and cheap access to computing power makes it possible to solve complex and large systems numerically. The solutions can then be studied via methods such as numerical bifurcation analysis (Theodoropoulos, Qian, and Kevrekidis, 2000; Gear, Kevrekidis, and Theodoropoulos, 2002; Marschler et al., 2013). Finally, the results can be visualized.

This chapter gives a brief introduction to the broad theory of dynamical systems and their numerical analysis, with a special focus on systems with more than one scale. In section 2.1, we begin with definitions of basic terms, such as *manifold* and *dynamical system*. We use a geometric view on dynamical systems in section 2.2 to introduce linear and nonlinear manifold learning techniques. The most intricate part of a transition from one scale to a coarser scale is the *closure problem*. It is defined in section 2.3, along with a definition of time delay embedding. The main contribution of this thesis solves the closure problem with the construction of a so-called *dynamic surrogate model* in the next chapter. This surrogate model is a dynamical system, constructed on a parametrization of a manifold embedded into a high-dimensional space.

2.1 Manifolds and dynamical systems

Definitions and notation used in most parts of the thesis are detailed here. Informally stated, we provide the mathematical description of a smooth surface that is locally flat, which means it locally resembles d-dimensional Euclidean space. This surface is called d-dimensional, smooth manifold. Then, we define directions or velocities for each point of the surface, which is possible due to the surfaces' smoothness. Given an initial point and its velocity on the surface, we can traverse the surface by following the directions of velocities, which constitutes the basic behavior of a dynamical system. The definitions given next are more precise, but essentially describe the same environment. Unless otherwise stated, the definitions are adapted from Perrault-Joncas and Meilă (2011) and Lee (2012). To understand manifolds and their resemblance to Euclidean space, it is necessary to define continuous deformations between spaces. Such a deformation is called homeomorphism. If the deformation and its inverse is differentiable, it is called diffeomorphism.

Definition 1. *A* ***homeomorphism*** *$\phi : A \to B$ between two topological spaces A and B is a continuous, bijective function with a continuous inverse ϕ^{-1}.*

Homeomorphisms can be understood as a continuous "stretching" and "bending" of the space A into the space B. While illustrative, this is an informal definition, and does not cover all homeomorphisms. Equipped with the concept of homeomorphisms, we can define a d-dimensional manifold. Then, we will use differentiability to define a special type of manifolds, namely smooth manifolds.

Definition 2. *A d-dimensional* ***manifold*** *M is a topological Hausdorff space such that every point has a neighborhood homeomorphic to an open subset of $\mathbb{R}^d$. A* ***coordinate chart*** *(U, ϕ) of the manifold M is an open set $U \subset M$ together with a homeomorphism $\phi : U \to V$ from U to an open subset $V \subset \mathbb{R}^d$. An* ***atlas*** *$\mathcal{A}$ is a set of charts*

$$\mathcal{A} = \cup_{\alpha \in I}\{(U_\alpha, \phi_\alpha)\}, \tag{2.1}$$

where I is an index set such that $M = \cup_{\alpha \in I} U_\alpha$. If for all $\alpha, \beta \in I$ the transition map between charts,

$$\phi_\alpha \circ \phi_\beta^{-1} : \phi_\alpha(U_\alpha \cap U_\beta) \to \mathbb{R}^d, \tag{2.2}$$

is continuously differentiable any number of times, the atlas is called ***smooth atlas*** *or C^∞-atlas. Finally, a* ***smooth manifold*** *M is a manifold with a smooth atlas.*

Figure 2.1 shows how the chart (U_α, ϕ) relates the Euclidean space $\mathbb{R}^d$ with the manifold M.

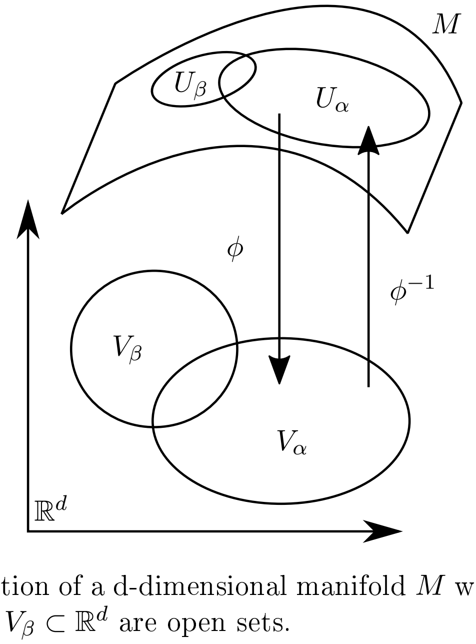

Figure 2.1: Visualization of a d-dimensional manifold M with chart (U_α, ϕ). The subsets $U_\alpha, U_\beta \subset M$ and $V_\alpha, V_\beta \subset \mathbb{R}^d$ are open sets.

Definition 3. *Given two manifolds N and M of the same dimension, a C^k-**diffeomorphism** $\phi : N \to M$ is a homeomorphism where both ϕ and ϕ^{-1} are k times continuously differentiable. In this case, N and M are **diffeomorphic**.*

In general, a manifold cannot be represented by a global coordinate chart. The textbook example for this is the 2-sphere, since there exists no homeomorphism between the 2-sphere and the Euclidean space $\mathbb{R}^2$—every atlas of the 2-sphere must contain at least two charts. As it is often preferable to represent a manifold by one homeomorphism to and from Euclidean space, the dimension of the Euclidean space can be chosen larger than that of the manifold. This concept, defined in the following text, is called **embedding**, and relates smooth manifolds by mapping between their **tangent spaces**. The concept of embedding will be essential in the numerical reconstruction of state spaces described below. The concept of a tangent space T_pM at a point p on a d-dimensional manifold M can be understood informally as being a d-dimensional linear subspace tangent to M at p. This informal definition requires an ambient space, while typically, tangent spaces are defined through intrinsic properties of the manifold. For the presentation here, the informal definition suffices. Isham (2002) gives a textbook treatment with several formal definitions.

Definition 4. *Let M and N be two manifolds, and let $H : M \to N$ be a smooth function between them. Then, at each point $p \in M$, the Jacobian dH_p of H at p defines a linear mapping between the tangent spaces T_pM and $T_{H(p)}N$. The map H has **rank** k if $dH_p : T_pM \to T_{H(p)}N$ has rank k for all $p \in M$. We write $rank(H) = k$.*

Definition 5. *Let M and N be two smooth manifolds, and let $H : M \to N$ be a smooth injective map with $rank(H) = \dim(M)$, then H is called an **immersion**. If H is a homeomorphism onto its image, then H is called an **embedding** of M into N.*

The concept of an embedding plays a crucial role in the theorem of Takens, which constructs an embedding of a manifold into Euclidean space through concatenation of observations of points on the manifold (see section 2.3 for a detailed discussion). The Strong Whitney Embedding Theorem states that any d-dimensional smooth manifold can be embedded into $\mathbb{R}^{2d}$ (Eells and Toledo, 1992; Lee, 2012). For example, the 2-sphere can be embedded into $\mathbb{R}^4$. On the other hand, it is clear that the 2-sphere can also be embedded into $\mathbb{R}^3$. This shows that the theorem is not sharp for all manifolds, however, the theorem is tight in the sense that there are manifolds, such as real projective spaces, that need all $2d$ dimensions for an embedding (see Perrault-Joncas and Meilă (2011) and Perrault-Joncas and Meilă (2013)). Embedded manifolds and Whitney's Theorem are important presets for the reconstruction of manifolds embedded in high-dimensional spaces with numerical algorithms, which will be discussed in section 2.2.

Equipped with the definitions for smooth manifolds, we can now define dynamical systems.

Definition 6. *A **dynamical system** is a manifold M called **state space**, equipped with a diffeomorphism $\phi : T \times M \to M$, where T is called **time**.*

If $T = \mathbb{R}$, we call the system **continuous** and the diffeomorphism ϕ is called **flow**. If $T = \mathbb{N}_0$, the system is called **discrete** and ϕ is called **map**. We denote $\phi_t(x) := \phi(t, x)$, to clarify that a flow can also be seen as a map acting on elements x in the state space, pushing them forward in time by t. To introduce the concept of **differential equations**, consider an n-dimensional manifold M, the time $T = \mathbb{R}$ and the flow $\phi : T \times M \to M$. Let f be a smooth vector field on M, in other words, f is a C^∞-diffeomorphism from M to M. Then, the following differential equation forms a dynamical system with $\phi_0(x) = x$:

$$\frac{d}{dt}\left(\phi_t(x)\right) = f(\phi_t(x)). \tag{2.3}$$

The flow $\phi_t(x)$ is the solution to the differential equation. Defining $x(t) := \phi_t(x)$, the short notation for equation (2.3) is $\frac{d}{dt}x(t) =: \dot{x} = f(x)$, and will be used throughout the thesis. The set $\{x(t)|t \in \mathbb{R}^+, x(0) = x_0 \in M\}$ is called **trajectory** of the dynamical system 2.3, starting in $x(0) = x_0$. If the function f in equation 2.3 only depends on the value $\phi_t(x) = x(t)$, the equation is called **ordinary differential equation** (ODE). If f also depends on derivatives of ϕ with respect to x, the equation is called **partial differential equation** (PDE).

2.2 A geometric view on dynamical systems

In this section, we focus on the question how geometric objects formed through trajectories can be described. We explain and relate the concepts *geometry*, *heat kernel* and *Laplace-Beltrami operator*. Building on these concepts, the idea of *time-lagged embedding* is introduced in section 2.3. Time-lagged embedding allows to reconstruct a diffeomorphic copy of the state space of a dynamical system when only time-discrete, generic observations of the true system states are available—such as in an experiment.

Dynamical systems can exhibit many different behaviors, which is reflected in the shape of the trajectories over time. The trajectories can tend towards a single point in the state space called *steady state*. They can return to previous points, forming a *closed orbit*, or tend to such an orbit, forming a *limit cycle*. Figure 2.2 shows the two-dimensional state space of a system with a steady state in the center, surrounded by a limit cycle where all trajectories tend to if they do not start at the steady state. The steady state in the center is *unstable*, as trajectories starting in its vicinity do not return, but move away from it over time. The limit cycle in this example is *stable* because the opposite is true here. All trajectories in the vicinity of the limit cycle, except for the steady state, will end up on the limit cycle in the limit of infinite time. The set of all points on the limit cycle is called the *attractor* of the system, and the set of points that end up on the attractor in infinite time is called *attracting set*.

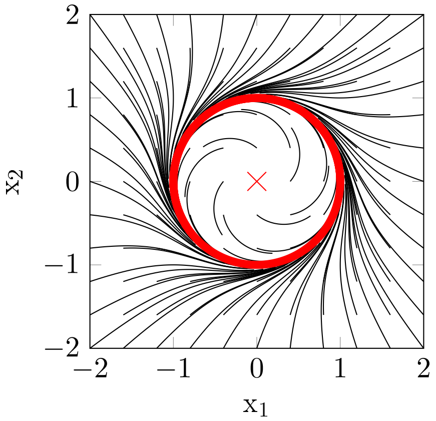

Figure 2.2: Trajectories (black curves) of a dynamical system with an unstable steady state (red cross) and a stable limit cycle (red circle). The trajectories are only shown for $t < 1$.

We will return to this example system later, constructing a new dynamical system for the new variable $y(t) = \|x(t)\|$ with a data-driven surrogate model.

The geometry of a given set of points is defined through properties of and relations between individual points and tangent vectors of the set. Coifman and Lafon (2006) describe geometry generically as "a set of rules describing the relationships between data points". Common rules, properties, or relations are positions, angles, and lengths. We define the *intrinsic dimension* of a set of points as the minimal number of variables needed to parametrize it. If the geometric object is a manifold, its intrinsic dimension is the dimension of the manifold—that is, the dimension of Euclidean space locally homeomorphic to open sets around every point of the object. As an example, figure 2.3 shows three sets of points with different geometries. The first one, a line, has an intrinsic dimension of one, because we can identify all points on it through a single number, the arclength. The second set is a curved surface, which has an intrinsic dimension of two. The third set of points is a two-sphere, also with an intrinsic dimension of two. As discussed in the previous section on manifolds, the sphere is a special set of points, because it cannot be transformed into the second set through any homeomorphism. Even though the minimum number of dimensions for an embedding is three, its intrinsic dimension is two.

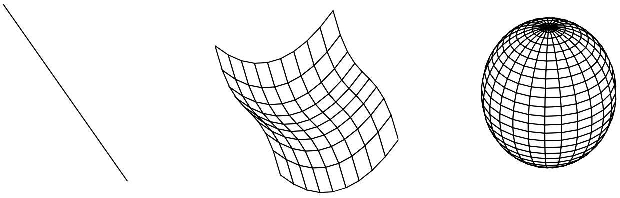

Figure 2.3: Objects with different geometries and intrinsic dimensions. Object (a) is a line segment of intrinsic dimension one, object (b) is an intrinsically two-dimensional surface, curved and embedded in three-dimensional space. Object (c) is a sphere, which also has intrinsic dimension two, but cannot be deformed through any homeomorphism into object (b).

Geometric objects can be embedded in a space with higher dimension than the shape itself. Let the set S contain the points of the shape, then an embedding $H : S \rightarrow E \supseteq S$ preserves the structure of S. In contrast to the intrinsic dimension of the object, we call the dimension of the ambient or embedding space E the *extrinsic dimension* of the object. Examples of embeddings are subgroups of groups, the real line in the complex numbers, and any object with an ambient space (also see Fig. 2.4). Depending on the application, the embedding space is considered extrinsic, without any special relation to the shape. In this case, dimension reduction is useful to project the geometric object onto a space with a dimension closer to the objects intrinsic dimension. In the process, the reduction methods often ignore the ambient space entirely.

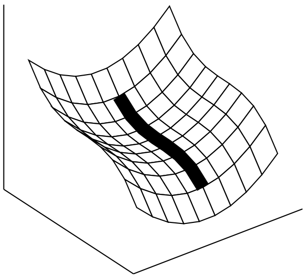

Figure 2.4: Embedding of a curved, one-dimensional object in a two-dimensional surface, which itself is embedded in three-dimensional, Euclidean space.

A common example used to demonstrate the power of dimension reduction is the manifold shown in figure 2.5, in the form of a Swiss roll (Lafon and Lee, 2006). It is a two-dimensional manifold with a hole in the middle (see figure 2.6), that is embedded in three dimensions (figure 2.5). Here, we use the embedding

$$H(x_1, x_2) = [\sqrt{x_1} cos(1.5\pi x_1), 2x_2 - 1, \sqrt{x_1} sin(1.5\pi x_2)] = (y_1, y_2, y_3) \qquad (2.4)$$

to map the two coordinates (x_1, x_2) of the surface into the three-dimensional ambient space coordinates (y_1, y_2, y_3). The goal of dimension reduction in this example would be to recover the intrinsic coordinates (x_1, x_2) of the surface, given the three-dimensional coordinates (y_1, y_2, y_3).

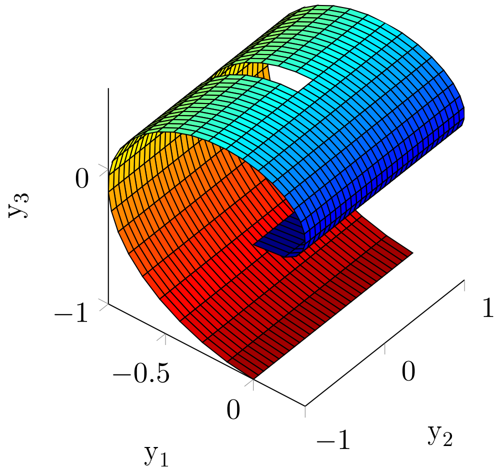

Figure 2.5: Embedding in three dimensions of a two-dimensional surface with a hole. The color indicates the values of the intrinsic coordinate x_1 of the surface.

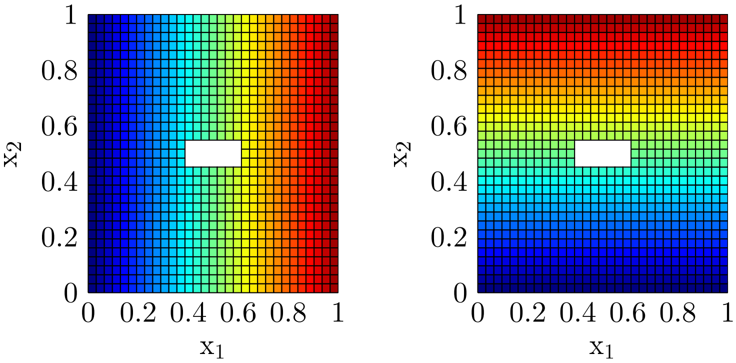

Figure 2.6: Intrinsic coordinates of the swiss-roll manifold. The colors plot show the values of coordinates x_1 (left) and x_2 (right).

2.2.1 Linear decomposition

A common tool in dimension reduction is the truncated singular value decomposition. The underlying idea of the reduction is the decomposition of a space into its relevant linear subspaces. This decomposition is achieved through a decomposition of a rectangular matrix, composed of columns as points of the given space, into two unitary matrices and a diagonal matrix of real values (equation 2.5), assigning each subspace a certain importance. The method is utilized under many names from different fields of science, such as linear principal component analysis (PCA) by Hotelling (1936), proper orthogonal decomposition (POD) by Pearson (1901), and Karhunen-Loeve decomposition by Karhunen (1946) and Loève (1946). Stewart (1993) gives a historical review, including a citation of Golub and Reinsch (1970) as "the algorithm [for the SVD] that has been the workhorse of the past two decades". To define the singular value decomposition, consider a matrix $A \in \mathbb{R}^{m \times n}$ with $m \geq n$ and $\text{rank}A = k$. Adapted from Golub and Reinsch (1970), the singular value decomposition of A is

$$A = U \Sigma V^T, \tag{2.5}$$

where U, V are unitary such that $U^T U = V^T V = I_n$, and $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)$. The matrix U is composed of the n orthonormalized eigenvectors of the matrix AA^T, associated with the largest n eigenvalues. The matrix V is composed analogously, but from the matrix $A^T A$. The diagonal matrix Σ contains $\sigma_1, \ldots, \sigma_n$, the non-negative square roots of the eigenvalues of $A^T A$. These values are called singular values of A. Assuming

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0,$$

we can approximate A by a truncated version $A_r = U \Sigma_r V^T$ of rank $r \leq k$, which ignores the columns of U and V associated with the smallest $k - r$ singular values by setting $\sigma_i = 0$ for $i > r$. The truncated matrix A_r is the best approximation of A in the Frobenius norm $\|A\| = \sqrt{\sum_{i=1}^m \sum_{j=1}^n A_{ij}^2}$, such that

$$\|A - A_r\| \leq \|A - B\| \tag{2.6}$$

for any matrix B with the same rank as A_r. This is the theorem of Eckart and Young, also called Schmidt's approximation theorem (see Stewart (1993)). The truncated singular value decomposition allows to ignore the columns of U and V associated with the smallest singular values, effectively reducing the dimensionality of A to its most important components. Figure 2.7 shows the decomposition of the surface in Fig. 2.3 (b) into two linear subspaces. The two intrinsic coordinates can be recovered successfully.

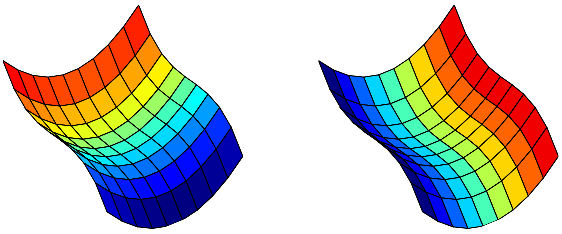

Figure 2.7: A surface decomposed into two linear subspaces. The new coordinate values are depicted by different colors.

A linear decomposition of a shape does not always produce a minimal number of coordinates for points of the shape. Considering the Swiss roll manifold, a linear decomposition yields a minimum of three basis vectors instead of two. In such a case, nonlinear dimensionality reduction must be employed.

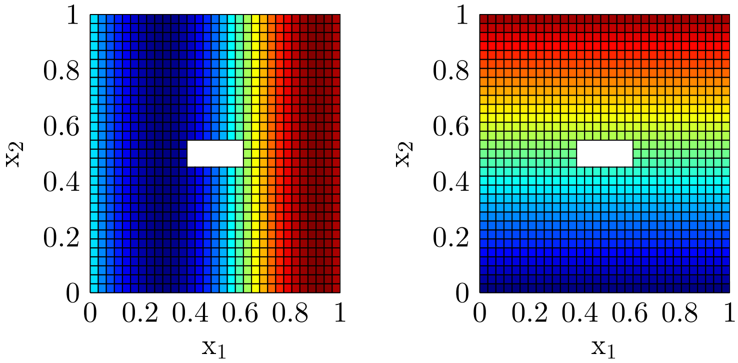

Figure 2.8: The reconstruction of the coordinates of the Swiss roll surface with a linear decomposition, performed through singular value decomposition. The x_2 coordinate is correctly identified, but x_1 is not, due to the nonlinearity of the embedding.

2.2.2 Non-linear decomposition

Many methods for nonlinear dimension reduction employ the kernel trick. Points of a manifold are embedded in a much higher-dimensional space where the shape is linear. Then, a linear decomposition yields the basis vectors of the space. In case a dimension reduction is possible, many of these will be redundant, and a small set is enough to span the reduced space (Hofmann, Schölkopf, and Smola, 2008). The kernel trick is evaluating the inner product in the high-dimensional space, without explicitly constructing the mapping into the space. The choice of the kernel is tied to the geometry of the high-dimensional space, and thus fundamentally important for a correct reduction. Kernels can be used to generate an arbitrary Riemannian geometry for the new, embedded manifold (Berry and Sauer, 2015). One choice used commonly is the heat kernel, as it weighs points close to each other higher than points far away. This property allows for a nonlinear decomposition which preserves local geometry but ignores the global structure of the shape.

The heat kernel is the solution to the heat equation, a partial differential equation on $\mathbb{R}^n$,

$$\partial_t u = \Delta u,\ \ u \in C^2(\mathbb{R} \times \mathbb{R}^n, \mathbb{R}), \tag{2.7}$$

with initial conditions

$$u(0, x) = \delta_0(x),\ \ x \in \mathbb{R}^n,$$

where $\Delta := \sum_{i=1}^{n} \frac{\partial^2}{\partial x_i^2}$ is the Laplace operator. The heat kernel has the form

$$K(t, x, y) = \frac{1}{(4\pi t)^{d/2}} \exp(-\|x - y\|^2/(4t)),\ \ x, y \in \mathbb{R}^d, t \in \mathbb{R}^+.$$

If the heat equation is solved on a smooth, compact, n-dimensional manifold M, the Laplace operator Δ in Euclidean space becomes the Laplace-Beltrami operator, which is also denoted Δ and defined through the Riemannian metric on the manifold (Coifman and Lafon, 2006). The heat kernel in this case is $T = e^{-t\Delta}$, which is associated to the integral transform of a solution u to equation 2.7, such that

$$Tu(x) = \int_{\mathcal{M}} K(t, x, y)u(y)dy,\ x, y \in M.$$

The Laplace-Beltrami operator on a manifold M contains fundamental, local properties of the geometry of the manifold, because the operator describes a diffusion process over M (see equation 2.7). If a—possibly nonlinear—coordinate transform from M into a lower-dimensional space preserves the diffusion properties of the operator, it preserves the local geometry, and is therefore suited for nonlinear dimension reduction. Such a coordinate transform is realized via *diffusion maps*, introduced by Coifman et al. (2005). They provide a numerical approximation of the eigenfunctions of the Laplace-Beltrami operator on a compact manifold.

The eigenfunctions of the Laplace-Beltrami operator are the solutions of the Helmholtz equation

$$-\Delta\psi = \lambda\psi,$$

where λ are the eigenvalues and ψ are the eigenfunctions. The Laplace operator on a compact manifold admits a discrete spectrum with a countable number of eigenfunctions $[\psi_1, \psi_2, \dots]$. A subset of eigenfunctions is the coordinate transform Ψ_t from the high- to the low-dimensional space, and is called *diffusion map* for a diffusion process of length t. The diffusion map can be written in vector form as

$$\Psi_t(x) = \left[\lambda_1^t\psi_1(x), \lambda_2^t\psi_2(x), \lambda_3^t\psi_3(x), \dots\right].$$

Coifman and Lafon (2006) prove that the diffusion distance on the original manifold is approximated by the distance between diffusion mapped coordinates. This allows to find the intrinsic distance on the m manifold M between two points $x_i, x_j \in M$, given their coordinates in the extrinsic space of M. After a time t of "heat propagation" of delta functions $\delta_{x_i}, \delta_{x_j}$ initialized at the points x_i, x_j, the diffusion distance D_t between the two points, defined by the L^2 distance between the propagated heat, is approximated by the distance between points in diffusion map space:

$$D_t^2(x_i, x_j) = \|e^{-t\Delta}\delta_{x_i} - e^{-t\Delta}\delta_{x_j}\|^2_{L^2(\mathcal{M})} \approx \|\Psi_t(x_i) - \Psi_t(x_j)\|_2^2$$

Coifman and Lafon (2006) also provide a short numerical algorithm to approximate the values of the eigenfunctions on given data points. In case the data is sampled uniformly, the algorithm is reduced to the following:

1. Compute a distance matrix $D_{ij} = d(x_i, x_j)$ between all pairs of data points x_i, x_j. The function d is a metric between points x_i, x_j on the manifold M, for example $d(x_i, x_j) = \|x_i - x_j\|$ on a normed space.

2. Define the weight matrix $W_{ij} = \exp(-D_{ij}/\epsilon)$, for a small $\epsilon > 0$ depending on the sampling density.

3. Compute the row sums of W by $S_{ii} = \sum_j W_{ij}$ in a diagonal matrix S.

4. Normalize W to $A = WS^{-1}$, so that A can be interpreted as a Markov matrix with jump probabilities A_{ij}.

5. The eigenvalues of A are the eigenvalues of the Laplace-Beltrami operator, and the eigenvectors contain the values of the eigenfunctions ψ_k, evaluated on the data points.

The algorithm is extended to work with non-uniformly sampled data, and to add numerical stability (see Algorithm 3.1.2 in section 3.1.2). When applied to the three-dimensional Swiss roll data from the previous section, the eigenfunctions provide a suitable parametrization of the surface, and an effective dimension reduction to the plane (see figure 2.9).

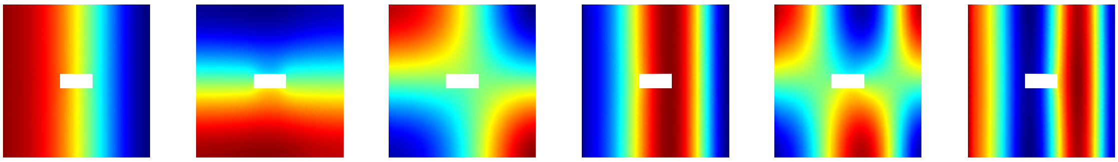

Figure 2.9: Coordinates of the Swiss roll surface with a nonlinear decomposition performed through diffusion maps, ordered by absolute value of the diffusion map eigenvalues. The first two nontrivial eigenfunctions of the diffusion map correctly identify the coordinates x_1 and x_2 of the surface, while the other eigenfunctions are harmonics of the first two.

2.3 Time-delay embedding and Takens' theorem

One of the main challenges for a successful transition of scales is that the model on the new scale is independent of the lower scale. Independence means that future states of the system can be computed through the current state alone, without additional information, for example, from a different scale. As this challenge occurs in many different disciplines, it has many different names, for example *closure* (Kevrekidis and Samaey, 2009), *Markov property* or *memory-less* (Ethier and Kurtz, 1986), and also appears in statistics as *dependent* or *explanatory variables*. Here, we call it the *closure* problem. Ultimately, the problem results from the choice of the wrong observables, or state variables, for a system.

Mathematically, the dynamic on the macroscopic scale is well-defined for every point in the state space of the macroscopic model. Figure 2.10 visualizes the case where the observation is a projection from a three-dimensional state space to a two-dimensional observation space. In three dimensions, given any point, the next step on the line is clear—whereas in two dimensions, given only the point at the crossing, it is impossible to define the next point uniquely.

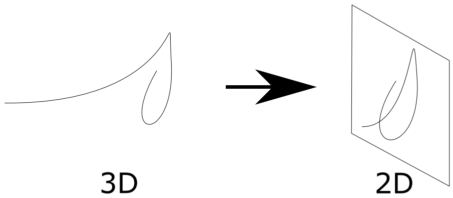

Figure 2.10: Visualization of the closure problem. After the observed values are projected onto a lower-dimensional space, their trajectories intersect.

A more elaborate example is the Lorenz system, known for its chaotic behavior and the wing-shape of its attractor. The Lorenz system is also used frequently in the introduction of time-lagged embedding theory, because it can be reconstructed from delays in time of either the first or second coordinate. Its attractor is the limit set of the dynamical system

$$\begin{aligned}\dot{x}_1 &= \sigma(x_2 - x_1),\\ \dot{x}_2 &= x_1(\rho - x_3) - x_2,\\ \dot{x}_3 &= x_1x_2 - \beta x_3,\end{aligned} \tag{2.8}$$

where σ, ρ, and β are parameters. For $\sigma = 10$, $\beta = 8/3$, and $\rho = 28$, the system exhibits chaotic behavior, shown in figure 2.11.

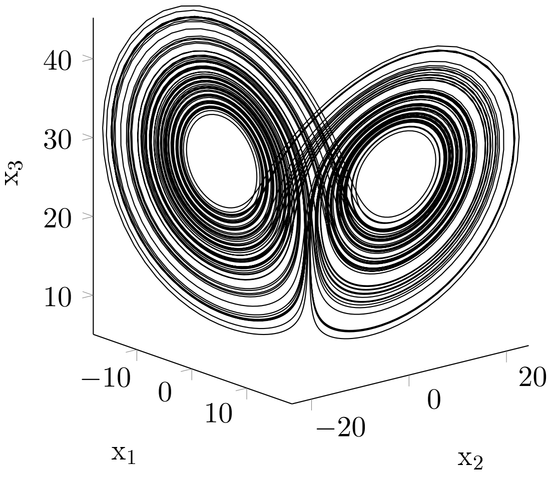

Figure 2.11: Lorenz attractor for parameters $\sigma = 10$, $\beta = 8/3$, and $\rho = 28$.

Figure 2.12 shows the observations of the first coordinate x_1 over time. If we only observe the coordinate x_1, and not all three coordinates, it is impossible to know the future states of x_1. In this particular example, due to the chaotic behavior of the Lorenz system, there are uncountably many future states for any given value $x_1(t)$ on the attractor.

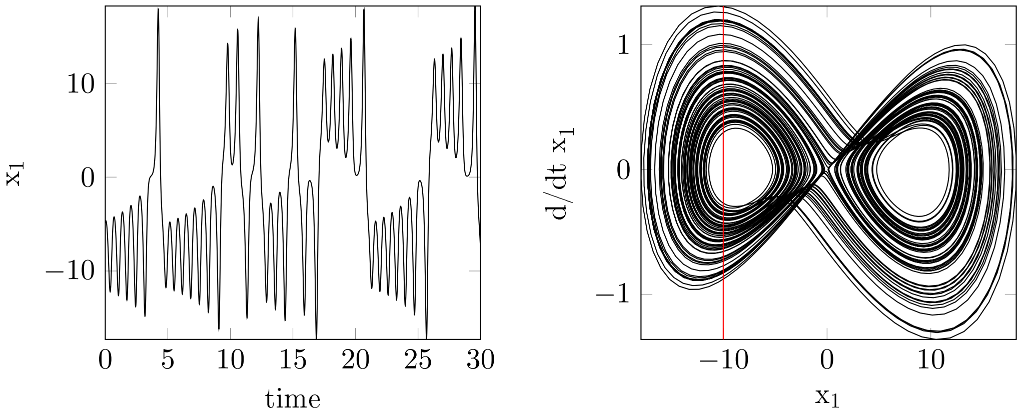

Figure 2.12: The left plot shows the first coordinate x_1 of the Lorenz system, over time. It is impossible to predict future states given only a single observation, which is obvious from the right plot, showing x_1 and its time derivative. The position $x_1 = -10$ is marked as a visual guide, to see that there are many possible values $\frac{d}{dt}x_1(-10)$.

The closure problem can be solved when more than the current point in state space is taken into account. This is precisely where the idea of time-lagged embedding enters (Ruelle and Takens, 1971; Takens, 1981). Theorem 1 is adapted from Takens (1981) to fit the notation used here, but otherwise it is left unchanged. The theorem provides an explicit embedding H between a manifold M and the Euclidean space $\mathbb{R}^{2n+1}$, by using time lagged versions of *observations* y of points $x \in M$:

Theorem 1. ***Time-delay embedding*** *Let M be a compact manifold of dimension m. For pairs (f, y), $f : M \to M$ a smooth diffeomorphism and $y : M \to \mathbb{R}$ a smooth function, it is a generic property that the map $\boldsymbol{H}_{f,y} : M \to \mathbb{R}^{2m+1}$, defined by*

$$\boldsymbol{H}_{f,y}(x) = [y(x), y(f(x)), \ldots, y(f^{2m}(x))] \tag{2.9}$$

is an embedding; by "smooth" we mean at least C^2.

The conditions on the observation function y are given in the proof. Quoted from Takens (1981):

> We may, and do, assume that if x is a point with period k of f, $k \leq 2m + 1$, all eigenvalues of $D_x f^k$ are different and different from 1. Also, we assume that no two different fixed points of f are in the same level of y.

The matrix $D_x f$ is the Jacobian matrix of the map f, such that $(D_x f)_{ij} = \frac{\partial f_i}{\partial x_j}$. Takens proved theorem 1 and two others, where

1. the map f is replaced by a flow ϕ_t with a constant $t > 0$, and
2. using the 0-th to the $2m$-th time derivative of the flow.

Takens' theorem and the resulting method of delays provide the means to construct a space diffeomorphic to the original state space M. We call this new space *delay space*. If the elements of M originate from a deterministic dynamical system with map f or flow ϕ_t, the space M is closed by definition, since being closed means a given state in M contains enough information for the system to predict future states. The diffeomorphism between the original space M and the delay space implies that the new space is also closed under the flow of the dynamical system. For many systems, including $2m + 1$ delays in the observation vector is enough to reconstruct a diffeomorphic copy of M. However, there are recent results from Berry et al. (2013) on including many more delays, improving smoothness of the new space, and also projecting onto a stable subspace of the dynamical system. Berry et al. (2013) employ diffusion maps for nonlinear dimensionality reduction of the embedded manifold, and show the relation to a Fourier analysis on the original manifold M. They also show that adding more delays has a smoothing effect on the dynamics, acting similar to a filter for noisy data. We will use the modified time delay embedding from Berry et al. (2013) to construct surrogate models later. The modified version needs the following preliminaries. Let $x \in M$, and define a dynamical system on M through

$$x(t + 1) = f(x(t)). \tag{2.10}$$

Also, let $\kappa \in \mathbb{R}^+$ and $H(x)$ be a vector of $T + 1$ observations through a vector-valued function $y : M \to \mathbb{R}^m$, $m \in \mathbb{N}$, such that

$$\mathbf{H}(x) = \left[e^{-T\kappa}y(x), e^{-(T-1)\kappa}y(f(x)), \ldots, e^{-\kappa}y(f^{T-1}(x)), y(f^T(x))\right] \in \mathbb{R}^{(T+1)m}. \tag{2.11}$$

Then, for sufficiently large T, theorem 1 states the existence of a function $g : \mathbb{R}^{t+1} \to \mathbb{R}^{(T+1)m}$, such that

$$\mathbf{H}(f(x)) = g(\mathbf{H}(x)), \tag{2.12}$$

and $H(x(t + 1)) = g(H(x(t))$ defines a discrete dynamical system on the delay space $H(M)$ with dynamic g. The manifold $H(M)$ is diffeomorphic to the state space manifold M of the original system. Berry et al. (2013) show that for large T and a suitable value $\kappa \in \mathbb{R}^+$ in the Lyapunov metric, the embedding $\mathbf{H}$ (Eq. 2.11) projects onto the most stable Oseledets subspace. This subspace is closed under the dynamics of the original system, which is the main motivation to construct a model on it. The choice of κ is discussed in general in (Berry et al., 2013), and we discuss its choice for specific applications in section 3.1.2. Reconsidering the Lorenz attractor, it is enough to include two delays in time to reconstruct a diffeomorphic copy of the original system (figure 2.13).

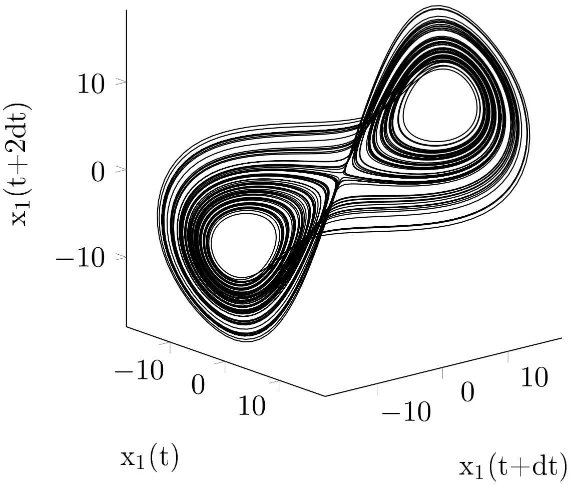

Figure 2.13: Diffeomorphic copy of the Lorenz attractor, constructed through points $[x_1(t), x_1(t + dt), x_1(t + 2dt)]$.

Takens' theorem assumes the observations are *generic*, so that there are no symmetries. For the Lorenz attractor, the coordinates x_1 and x_2 are generic, but the coordinate x_3 has a symmetry. This causes the reconstructed manifold to only have one “wing of the butterfly” (figure 2.14), as one cannot discern between wings when measuring x_3 only.

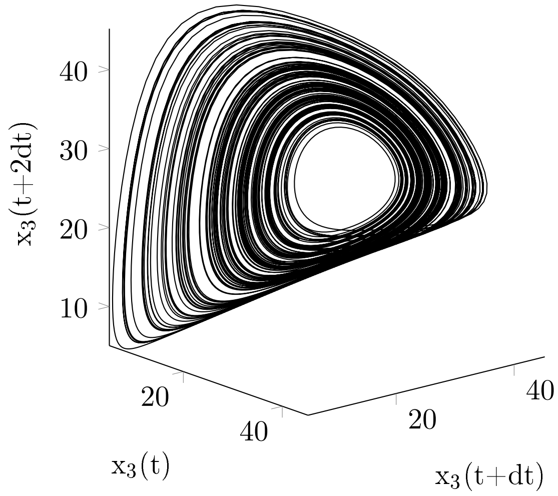

Figure 2.14: A degenerate copy of the Lorenz attractor, constructed through points $[x_3(t), x_3(t+dt), x_3(t+2dt)]$. Due to symmetry, the two circular regions of the attractor are merged into one.

From Takens (1981), we see that a time-delayed embedding creates a diffeomorphic copy of the state space manifold M in a higher-dimensional Euclidean space. A time-delayed embedding is produced by a smooth, nonlinear observation function $y \in C^\infty(M, \mathbb{R}^m)$ of the state space M, by combining a number $(T+1) \in \mathbb{N}$ observations *over time* into a new point in a high-dimensional Euclidean space. The derivative of the observation function y must have full rank, so that $\|Dy\| > 0$ on M.

If n is the dimension of the manifold M, then the number of lags T must be larger than $2n$ for $\mathbf{H}(M)$ to be an embedding of M. In theory, the precise choice of T does not matter, as long as it is larger than $2n$. For example, Berry et al. (2013) use thousands of delays to smooth the chaotic trajectories present in their examples, even though the intrinsic dimension of the reconstructed state space is quite low. The stability of the theoretical results regarding the choice of T is very important for applications, as the intrinsic dimension n of the state space manifold is usually unknown. For the data-driven surrogate models introduced in section 3.1.2, we also need to include the number of parameters in the intrinsic dimension (see theorem 2 in section 3.2.2). Small and Tse (2004) provide a short review on how to compute T in combination with the delay between individual observations, and Huke and Broomhead (2007) describe an approach where the time delays must not be equal for all observations. In the original form of the theorem, the observations $y(x)$ must be scalar and must also be generated without observational noise and without stochastic effects in f. The multivariate case is described by Deyle and Sugihara (2011), and the stochastic version by Stark et al. (1997).

Dynamical systems and the geometry of their state space are now defined. Time-delay embedding of observations from a given system yields a diffeomorphic copy of its state space. For multiscale dynamical systems, time-delay embedding yields the possibility to observe a microscopic system on a larger scale, and use the diffeomorphic copy of the state space to generate a model on this larger scale. The next section briefly reviews current methods to treat upscaling and analysis of multiscale systems.

2.4 Methods for multiscale modeling and analysis

Modeling and analysis of systems with multiple scales is a very broad field. Still, as far as literature is concerned, modeling and analysis of systems with one scale far precedes multiscale systems. Here, we focus on systems with more than one scale. We review the state of the art, along with remarks on current challenges. We also introduce many of the concepts later referred to in the analysis of the dynamic surrogate (section 3.2) and its application (section 4).

There is an important difference between finding a dynamical system on a coarser scale, and reducing the dimension of a state space. Coarsening the scale of a system might need an even higher-dimensional state space. An example is the transition from a particle system to the probability distribution in the limit of an infinite number of particles with zero size. Usually, a system of ordinary differential equations is used as a description of the particle system on the microscopic scale, whereas a partial differential equation describes the evolution of the probability distribution on the macroscopic scale.

In this case, the particle system has a finite-dimensional state space, enumerated by the particle index and spatial dimension. The probability distribution is a function, and as such an element of an extrinsically infinite-dimensional space. Hence, the coarser system has an infinite-dimensional state space, compared to the finite-dimensional microscopic system. The output observed from the coarse system is on a coarser scale, either spatially, temporally, or both.

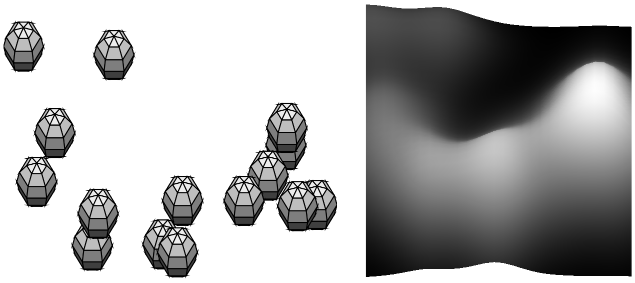

Figure 2.15: On the left, individual particles (spheres) are distinguishable, and the system state contains discrete positions. On the right, the same state is shown on a coarser scale, the density of the particles, using a superposition of slowly decaying exponential functions at each particle position. The particle positions are no longer distinguishable, and the new state is a function.

If coarsening the system is not an option, the information needed by the system to advance in time can normally be reduced by a reduction of the dimension of the state space. This reduction is called *model order reduction*, and hence results in *reduced order models*. It does not change the scale of the output observed from the system. Compared to the observations of the full model, some methods reduce the accuracy to achieve an even greater reduction, but the scale of the observations is still the same. Figure 2.16 demonstrates the process of model order reduction geometrically. The sphere is approximated with less and less points, or *to a lower order*, but it is still resembling a sphere, and a dynamical system acting on this sphere is defined at the same spatial scale. For a similar example and a comprehensive review of the field including its history, see the book from Schilders, Rommes, and van der Vorst (2008).

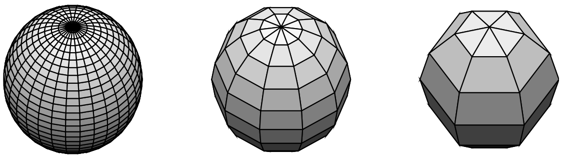

Figure 2.16: Illustration of model order reduction. A detailed state space (left sphere) is approximated by less detailed versions, while important features are kept: the right object has the same size and is still a spheroid object. The approximation might reduce the accuracy of the solution of a system, but does not change the scale the system is solved at.

One of the most prominent examples of model order reduction is the method of adaptive mesh refinement (Garcia et al., 1999). Adaptive mesh refinement is a common tool used in computational fluid dynamics, where the solution to a flow system is represented with different accuracy at different points in space. The adaptive refinement in regions of space where the solution needs to be more accurate is a form of model order reduction, since without adaptive refinement, the accuracy would have to be high over all of space.

Here, we discern methods for multiscale systems through their need for either data or formulaic models. We call a method *data-driven* if it primarily functions by manipulating numerical data generated by the system under study. *Equation-driven* methods manipulate the formulaic description of the system, with little or no need for data. Figure 2.17 shows our classification of methods and applications for multiscale systems. The two dimensions of complexity relate equation-driven and data-driven approaches. The equation-driven dimension shows how difficult it would be to treat the problem primarily with analytic tools. The data-driven dimension shows the difficulty to treat the problem numerically. This includes complexity of implementation of the algorithm, as well as the problem of sampling high-dimensional spaces. The sampling problem is often called *curse of dimensionality*, an expression originally coined by Bellman (1957) (also see Bellman (2003)). Essentially, the curse of dimensionality expresses the challenge that the amount of data needed to equally sample a certain volume of space increases exponentially with the dimension of that space. We will discuss this further in section 3.2.5, concerning the data needed to store the surrogate model.

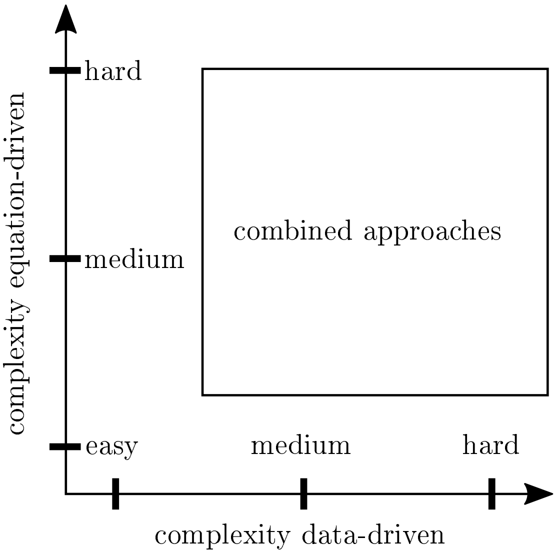

Figure 2.17: Levels of complexity for data-driven and equation-driven methods. If a method only modifies the formulaic description of a model, it lies on the vertical axis. Analogously, a purely data-driven method would lie on the horizontal axis. Most methods combine both a reformulation and a numerical approach, and would be placed off the axis.

Table 2.1 adapted from E et al. (2007) contains general and special purpose solvers for multiscale problems. A special purpose solver utilizes special properties of the system to solve it, such as scale separation. The solution does not always have to be on the microscopic scale. A general purpose solver aims to solve the system on the microscopic scale as accurately as possible, and is using information from other scales. We discuss some of the methods in table 2.1 in sections 2.4.1 and 2.4.2, and for a more detailed review refer to the book of E (2011). The complexity one faces when applying the methods is shown in figure 2.18. The level of complexity is assessed by counting the number of steps needed to successfully solve a problem. For example, a purely equation-driven method could have a very low data-driven complexity, because no data is needed. However, the method could still be very complex to apply for a specific problem. We assess the complexity level of seven example problems at the beginning of chapter 4. Here, we do not assess the methods complexity by the breadth of problems they are able to solve. This implies that a method could be quite simple to apply, but solves very few problems—compared to a complex method that solve many different problems. Nevertheless, all methods shown here can solve an abundance of different problems successfully.

Table 2.1: General and special purpose multiscale solvers.

General	Multigrid method (Brandt, 2005; Brandt et al., 2011) Domain decomposition (Quarteroni and Valli, 1999) Wavelet-based (Daubechies, 1992) Adaptive mesh refinement (Ainsworth and Oden, 1997) Fast multipole method (Greengard and Rokhlin, 1997) Conjugate gradient method (Hestenes and Stiefel, 1952; Golub and Van Loan, 1996)
Special	Car-Parinello method (Car and Parrinello, 1985) Quasi-continuum method (Tadmor, Ortiz, and Phillips, 1996) Optimal prediction (Chorin, Kast, and Kupferman, 1998) Heterogeneous multiscale methods (E and Engquist, 2003) Gap-tooth scheme (Kevrekidis et al., 2003) Adaptive mesh and algorithm refinement (Garcia et al., 1999)

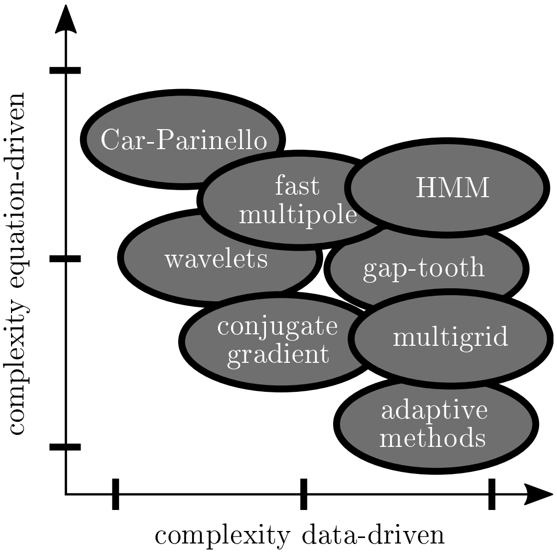

Figure 2.18: Levels of complexity for the methods.

2.4.1 Equation-driven methods for multiscale systems

The thesis' contribution, a dynamic surrogate modeling method, is ultimately a data-driven method for upscaling. There are also equation-driven methods available that can homogenize a given dynamical system, and yield a new one on a coarser scale. If feasible, this is preferable to data-driven methods, because the formal description of the new system is available, and enables further analysis. Nevertheless, as discussed in the last section, feasibility is often an issue—which makes data-driven methods a viable alternative. After all, data-driven methods can readily be applied on a new, analytically homogenized system, until it is understood enough to start with equation-driven methods.

Many properties of multiscale systems can be presented through systems with two scales (Pavliotis and Stuart, 2008; Kuehn, 2015), so called *slow-fast systems*. Such a slow-fast system of ordinary differential equations (ODEs) with two scales has *fast variables* x and *slow variables* y. The fast and slow dynamics are governed by functions f and g, respectively. The scales are separated by a small constant ϵ with $0 < \epsilon \ll 1$, called the *scale parameter*, such that

$$\begin{aligned} \epsilon\dot{x} &= f(x, y, \epsilon), \\ \dot{y} &= g(x, y, \epsilon). \end{aligned} \tag{2.13}$$

If the system is described through partial differential equations (PDEs), the scale parameter ϵ often appears as coefficient of the highest-order derivatives (E, 2011). One example are the Navier-Stokes equations for incompressible flows at large Reynolds numbers Re—which in this case serves as the scale parameter $\epsilon = \frac{1}{\mathrm{Re}}$ (adapted from E, 2011, p.4):

$$\begin{aligned} \rho_0(\partial_t u + (u \cdot \nabla)u) + \nabla p &= \frac{1}{\mathrm{Re}}\Delta u \\ \nabla \cdot u &= 0. \end{aligned}$$

Slow-fast ODE and PDE systems similar to equations 2.13 and 2.4.1 can be treated by singular perturbation theory (Kevorkian and Cole, 1996; Verhulst and Bakri, 2007), with Fenichel theory as its basis.

Fenichel theory provides theorems to treat slow-fast systems (equation 2.13, see the original work of Fenichel (Fenichel, 1972; Fenichel, 1979), and a modern treatment in the book of Kuehn (2015)). Figure 2.19 illustrates the general setting described by the theory. A dynamical system with a small scale parameter ϵ generates trajectories on a manifold $\mathcal{M}_\epsilon$. In the limit of $\epsilon = 0$, the system is reduced to trajectories on $\mathcal{M}_0$, which is typically easier to handle than $\mathcal{M}_\epsilon$, because the fast components of the system are removed completely.

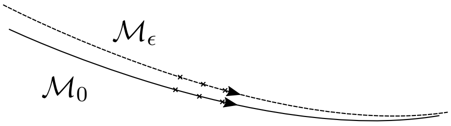

Figure 2.19: Manifold $\mathcal{M}_0$ for the limit case $\epsilon = 0$ (line) and approximating manifold $\mathcal{M}_\epsilon$ (dashed line). Fenichel theory uncovers the properties of $\mathcal{M}_0$ that ensure the existence of the approximating manifold. The theory also defines when the behavior of dynamical systems on $\mathcal{M}_0$ is similar to the behavior on $\mathcal{M}_\epsilon$, which is shown by three consecutive time steps here (crosses).

The theorems provide the tools for geometric singular perturbation theory, and the theoretical basis for methods relying on numerical approximations of manifolds. Essentially, the theorems assure existence of manifolds $\mathcal{M}_\epsilon$ close to $\mathcal{M}_0$, and state that the behavior of the system on $\mathcal{M}_\epsilon$ is not very different from the behavior on $\mathcal{M}_0$. Numerical approximations of slow-fast systems (2.13) are thus accurate even in the vincinity of $\mathcal{M}_0$.

There are also systems with more than two scales. The number of scales can be finite, but also countably infinite, or even continuous. In the latter case, no differentiation between scales is possible. Examples of systems with a continuum of scales are turbulent flow systems. These systems are often very difficult to solve numerically, because the whole continuum of scales interacts and there is no possibility to cut off scales after a certain, finite number. Usually, a physically inspired closure relation is used to treat the missing scales after a cutoff. An example of a closure relation is the assumption of linear dependence of the viscous stress on the local strain in Newtonian fluids (Panton, 2013). The closure is valid in a certain area of the system's state space, and approximates the influence of the scales not treated in the system of equations.

The book of Pavliotis and Stuart (2008) treats the theory of averaging and homogenization for ordinary, partial, and stochastic differential equations, as well as Markov chains (a special type of discrete dynamical systems). They consider linear dynamical systems of the form

$$\frac{d}{dt}u_\epsilon = \mathcal{L}_\epsilon u_\epsilon, \tag{2.14}$$

where u_ϵ is the solution, and $\mathcal{L}_\epsilon$ is a linear operator. Both depend on a small scale parameter ϵ. Since u_ϵ can be a function, both partial and ordinary linear differential equations are treated simultaneously. The operator $\mathcal{L}_\epsilon$ is assumed to have one of the two forms

$$\mathcal{L}_\epsilon = \frac{1}{\epsilon}\mathcal{L}_0 + \mathcal{L}_1 \tag{2.15}$$

or

$$\mathcal{L}_\epsilon = \frac{1}{\epsilon^2}\mathcal{L}_0 + \frac{1}{\epsilon}\mathcal{L}_1 + \mathcal{L}_2. \tag{2.16}$$

The first case is referred to as *averaging* or *first-order perturbation theory*, and the second case *homogenization* or *second-order perturbation theory*.

2.4.2 Data-driven methods for multiscale systems

Data-driven methods for multiscale systems can be separated into methods for analysis and methods for simulation. Model order reduction is traditionally an equation-driven method, but recently received considerable attention when performed data-driven.

It is possible to reduce the dimension of the state space of a dynamical system through a combination of equation- and data-driven techniques. The new, lower-dimensional state space is found by a change of basis from the original space to the new space. New basis functions allow for a more compact description of the state variables. An equation-driven method would find or choose the basis functions analytically, for example, a Fourier basis instead of the canonical basis in Euclidean space. However, for many systems, it is advantageous to choose basis functions adapted to the problem, which is possible through data-driven techniques. In this case, the basis functions are computed from data, and then used to represent the states of the system.

The *method of snapshots* (Sirovich, 1987) is a prominent example. Given a number of points in state space that the dynamical system visits over time (called snapshots), a new, reduced basis for the space of solutions is computed by principal component analysis. The dynamical system then can be solved with the principal components as a basis for the state space (see section 2.2.1 on linear decomposition, and example 4.1.1 for an application). If the number of principal components is significantly less than the original number of dimensions of the state space, a significant dimensionality reduction is performed. If the original dynamic is given through a linear operator, it can even be defined analytically on the new basis, which results in very accurate reduced order models. The book of Schilders, Rommes, and van der Vorst (2008) reviews the subject of model order reduction in general. Carlberg et al. (2013) discuss and advance the state of the art of non-linear, data-driven model order reduction in fluid dynamics.

Numerous methods are used to analyze systems with multiple scales. The analysis is performed through computation of quantities interesting for the given application. In most cases, a computationally efficient numerical simulation over time or space is necessary. If the methods are able to extrapolate from given data, the simulation is also called *forecasting*.

Many methods build on the basic idea of a decomposition of the state space into principal components, and almost all of the methods employ some form of spectral decomposition of matrices. For inherently linear systems, the theory and numerical applications are already developed very far, see (Moore, 1981) for an assessment of principal component analysis in linear systems for control and model reduction.

Barrault et al. (2004) describe the empirical interpolation method (EIM) for model reduction in nonlinear systems. Peherstorfer et al. (2014) build on their results with the Localized Discrete Empirical Interpolation Method (LDEIM). Romijn et al. (2008) combine a known physics model as an approximation to a given process, and add a black box model in form of neural nets to capture unknown nonlinearity. Coifman et al. (2008) use diffusion maps with dynamical systems and the equation-free approach, and Singer et al. (2009) use a special kernel in the diffusion map to find intrinsic, slow variables in highly stochastic systems.

Through the recent availability of large data sources, and the fast generation of numerical solutions to complex systems, the approximation of the *Koopman operator* for dynamcial systems has risen in importance and coverage in the literature (Rowley et al., 2009; Budišić, Mohr, and Mezić, 2012). The Koopman operator is defined in the following setting. Consider a nonlinear dynamical system

$$x(k+1) = f(x(k)),$$

with $x \in \mathbb{R}^n$, $k \in \mathbb{N}$, and observation functions $y : \mathbb{R}^n \to \mathbb{R}$. The Koopman operator $\mathcal{K}$ is a *linear* functional, acting on the observables y by projecting them further in time, such that

$$(\mathcal{K}y)(x_k) = (y \circ f) = y(x_{k+1}). \tag{2.17}$$

Budišić, Mohr, and Mezić (2012) introduce the operator in a setting that includes both ODEs and PDEs, and the work of Brunton et al. (2016) combines Takens delay embedding and the Koopman operator in an effort to predict chaotic systems over a certain time span using a linear, forced dynamical system. The composition property of $\mathcal{K}$ of the observation and the dynamic of the system is the reason for the alternative name *composition operator*. The linearity of the Koopman operator makes it an interesting object for numerical approximation as a matrix. Dynamic Mode Decomposition (Schmid, 2010) and its extensions (Williams, Kevrekidis, and Rowley, 2015) leverage this linear structure to approximate the operator in finite dimensions. The eigenvalues and eigenfunctions of the operator provide a decomposition into spatio-temporal features of the dynamical system under study (Giannakis, Slawinska, and Zhao, 2015). A diffusion map of time-delayed observations provides a non-linear and typically low-dimensional embedding of dynamical data. The Koopman operator provides a complementary picture, with an infinite-dimensional state space, but linear dynamic and observation function.

The multigrid algorithm and its extensions to systematic upscaling (Brandt, 2005; Brandt et al., 2011), equation-free methods (Theodoropoulos, Qian, and Kevrekidis, 2000; Kevrekidis and Samaey, 2009), and heterogeneous multiscale methods (E and Engquist, 2003; E et al., 2007) are methods for *on-demand*, data-driven upscaling. The general idea is to assume a model on the macroscopic scale exists, but only its state space can be accessed through observation of the state of a microscopic model on a finer scale. To advance in the macroscopic domain, the microscopic model—most of the time, a simulator of the model—is queried on demand, to compute the next state of the macroscopic model (see figure 2.20).

$$\begin{tikzcd}
\mathrm{U} \arrow[r, dashed, "\text{macroscopic}"] \arrow[d] & \mathrm{F(U)} \\
\mathrm{u} \arrow[r, "\text{microscopic}"] & \mathrm{f(u)} \arrow[u]
\end{tikzcd}$$

Figure 2.20: General concept of using a microscopic simulation to advance on the macroscopic scale. Starting at the macroscopic scale, the current value of U is mapped to a microscopic state u, advanced with a microscopic simulation to $f(u)$, and then mapped back to the macroscopic scale. This procedure thus computes $F(U)$, a simulation step on the macroscopic scale.

Table 2.2: Names of the steps of systematic upscaling, equation-free methods, and heterogeneous multiscale method.

Method	step 1	step 2	step 3
Systematic upscaling, multigrid	prolongate	simulate	project
Equation-free methods	lift	simulate	restrict
Heterogeneous multiscale method	reconstruct	simulate	compress

All of the equation-driven and data-driven methods discussed above are used to generate models that exactly reproduce, or approximate, the output of a given, more complex model. Although not explicitly stated in most cases, the newly generated models can be thought of as surrogates for the original.

2.5 Surrogate models

The concept of a surrogate model is to replace a given model by another one, which is cheaper to evaluate or easier to analyze (Qian et al., 2006). Actually, every model of a process in nature can be studied instead of that process and is hence also a surrogate. A surrogate model, however, replaces a given model, not a real process. Figure 2.21 shows the process of numerical computation of surrogate models from observation data of an original model.

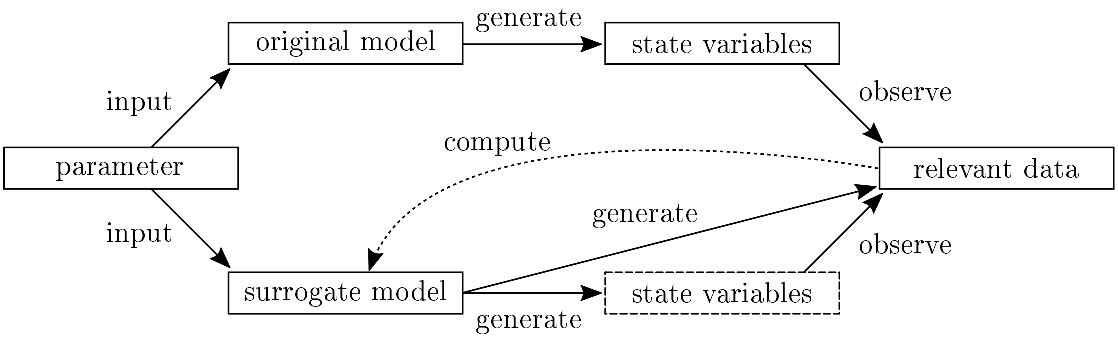

Figure 2.21: The concept of a surrogate model, compared to an original model. From given input parameters, the original model generates values of state variables over time, which are then measured to yield the relevant data for the application. From this data, a surrogate model can be computed, which can reproduce the relevant data either also through internal state variables, or directly. The thesis discusses the bottom-most path *input–generate–observe*, and the approximation of the functions involved.

A simple example for a surrogate model is a response surface. It does not capture dynamic information, and thus also does not need internal state variables, but yields observations independent of time, which is enough for many scenarios. Consider the scenario shown in figure 2.22, where 100 pedestrians move through a bottleneck (Dietrich, Albrecht, and Köster, 2016). The microscopic model is a dynamical system, the Gradient Navigation Model (Dietrich and Köster, 2014). The observed value is the total evacuation time - that means one number as an output of a complete simulation. The parameters are mean and standard deviation of the distribution of desired speeds of all pedestrians. Combined with the quantity of interest, the surrogate is a surface from the parameters to the evacuation time (figure 2.23).

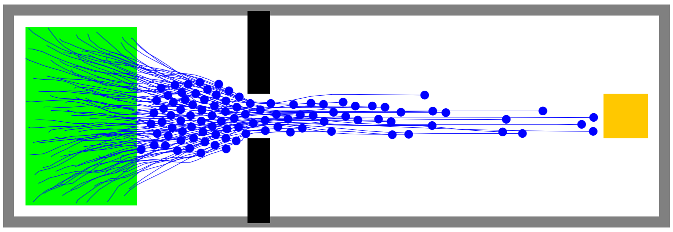

Figure 2.22: Example scenario where a surrogate can help estimate the evacuation time for given parameters. One hundred pedestrians move from the starting area on the left through the bottleneck, and to the target on the right. Their desired speeds are distributed randomly with a normal distribution $N(v, \sigma)$ with mean speed $v = 1.3ms^{-1}$ and standard deviation $\sigma = 0.26ms^{-1}$. Resulting desired speeds are accepted if they lie between 0.3 and 3.0, and redrawn otherwise.

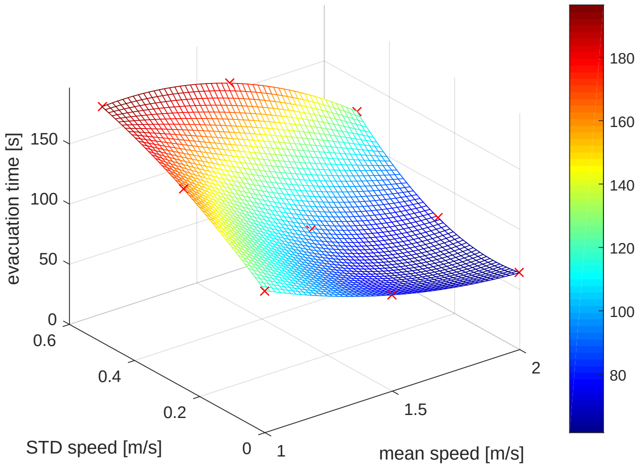

Figure 2.23: Response surface of the evacuation time through the bottleneck, given mean and standard deviation of the pedestrians desired speed. The nine red crosses indicate where microscopic simulations were used to estimate the evacuation time. The response surface is a quadratic interpolation of the results.

If the original model is not known, but only the observation data is available, the computation of surrogate models can be used to understand the dynamics and internal structure of the original model. This is one of the main objectives of machine learning. A response surface does not capture the dynamical system producing the observed values. Scenarios where the observed values depend on time require dynamical systems, which can be replaced by dynamic surrogates. Romijn et al. (2008) employ a grey-box modeling approach for the reduction of nonlinear systems, combining a known physics model as approximation and add a surrogate model, a neural network, to capture unknown nonlinearity. When the observed values are on a different scale than the dynamical system that produces them, constructing the dynamical surrogate model transitions these scales.

2.6 Summary

We introduced the concept of *manifold*, *embedding*, and *flow*. Using these concepts, we then defined the term *dynamical system*. Dynamical systems have trajectories, the set of states over time given a specific initial state.

Trajectories can be analyzed geometrically. For this, we introduced the geometric concepts of intrinsic and extrinsic dimension of an object. Using the example of a line, a curved surface, and a two-sphere, we explained the intrinsic dimension of a geometric object through parametrization. The extrinsic dimension was explained through the introduction of embedding, or ambient, space. This space surrounds the geometric object under study, and is vital to methods for dimension reduction. We discerned linear and non-linear methods for dimension reduction, and demonstrated their capabilities on the Swiss roll manifold, a two-dimensional surface rolled up in three dimensions. Linear methods perform poorly, as there is no linear subspace able to unfold the Swiss roll back to its two-dimensional state. We chose diffusion maps as an example of a non-linear method, and showed that the method can indeed reconstruct the two-dimensional nature of the Swiss roll.

Having introduced basic concepts and methods, we showed how they can be utilized in the more complex setting of Takens' theorem. This theorem concerns delay embedding, and is the theoretical basis for the reconstruction of dynamical systems from observation data. We will utilize the reconstruction through Takens' theorem in the next chapter, when building a surrogate model.

There are already many methods to treat multiscale, and, in particular, particle systems. In this section, we classified them as equation-driven or data-driven. We defined an equation-driven method to mostly use the formulaic description of the system, modifying equations to achieve dimension reduction or homogenization. A data-driven method was defined to work primarily on the data generated by the system under study. Some methods shown here utilize both equation-driven and data-driven aspects.

After the state of the art on multiscale methods, we focused on the concept of surrogate models. Data-driven surrogate models enable an efficient way to store information gained through microscopic models, or through multiscale methods. Capturing the dynamic of a process with a data-driven procedure is a challenge, which we solve as the main contribution of the thesis, and the next section.

Chapter 3

Dynamic, data-driven surrogate models

3.1 Scale transition with dynamic surrogate models

Dynamic, data-driven surrogate models can be on a different scale than the system creating the data. This section describes the construction process of the surrogate model in detail. We first give an informal description of the general idea. A precise mathematical definition follows, where tools such as manifolds, dynamical systems, and time-lagged embedding are employed. Figure 3.1 compares the original model generating observations on the left with the surrogate model on the right. Starting at the top, we have to choose a set of parameters to initialize the model. This includes the initial conditions of the state space. The state is advanced by the functions f and g, so that time series of the intrinsic state variables $x(t)$ and $z(t)$ are generated. The construction of the surrogate model encapsulates the parameters p into the new state variables z, such that unlike f, g does not depend on p anymore. At each point in time, the state can be measured by the observation functions y and $\tilde{y}$, creating the observations from the original and the surrogate model. The output of the surrogate model approximates the output of the original when a finite amount of data is used. In the limit of infinite data, the output is reproduced exactly. Both the approximation and the limit case are made precise by two theorems in the next section (theorems 2 and 3).

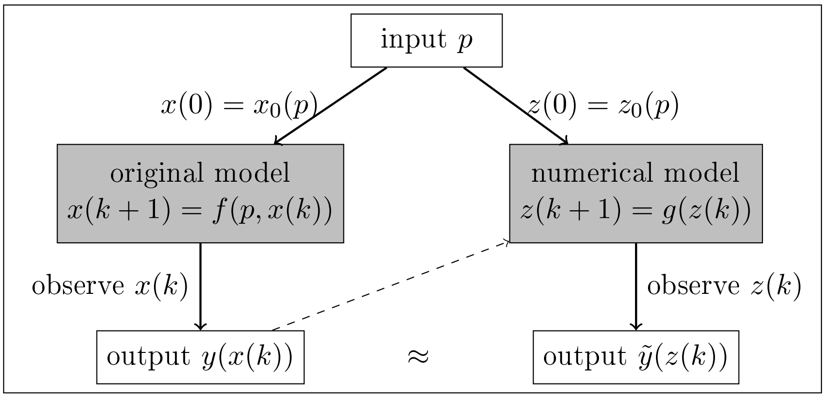

Figure 3.1: General structure of the original model (left) compared to the surrogate model (right).

The construction process of the surrogate is outlined here, while all steps are explained in detail in the following sections. Two dynamical systems are chosen to demonstrate the numerical procedures: a spiral system with a limit cycle, and a more applied example, where virtual passengers leave a train.

1. Construction
   (a) Choose relevant parameters.
   (b) Choose relevant observables.
   (c) Generate observation data from microscopic model by sampling the parameter space.
   (d) Embedding of the data through time lags.
   (e) Manifold learning with diffusion maps.
   (f) Interpolation of the new variables, which forms the surrogate model.
2. Simulation
   (a) Input the parameters.
   (b) Map into the space of closed observables.
   (c) Generate the state of the closed observables over time.
   (d) Map into original space by observing the generated states.

Figure 3.2: Construction and simulation process of the data-driven surrogate model.

3.1.1 Motivating examples: spiral and bottleneck scenario

We demonstrate the scale transition with surrogate models developed in this section on two examples. For a mathematically clear presentation, we use a spiral system in two dimensions. As a more applied example, we show the scale transition from micro to macro in a bottleneck scenario from crowd dynamics.

Spiral system with limit cycle

Consider a slow-fast system in two dimensions, written in angular coordinates as

$$\begin{aligned} \epsilon\dot{\alpha} &= 1 \\ \dot{r} &= r^3 - r, \end{aligned} \tag{3.1}$$

with $0 < \epsilon \ll 1$ and initial conditions $r(0) = r_0 \geq 0$, $\alpha(0) = \alpha_0 \in [0, 2\pi)$. This system has an unstable steady state for $r = 0$, and a stable limit cycle for $r = 1$. The radius r is the slow variable, the angle α is the fast variable. The scale parameter ϵ enters as the inverse of the angular speed—the smaller epsilon is, the faster the system spirals around the center at $r = 0$. Figure 3.3 shows the state space in Cartesian coordinates, for two different angular speeds $\frac{1}{\epsilon}$.

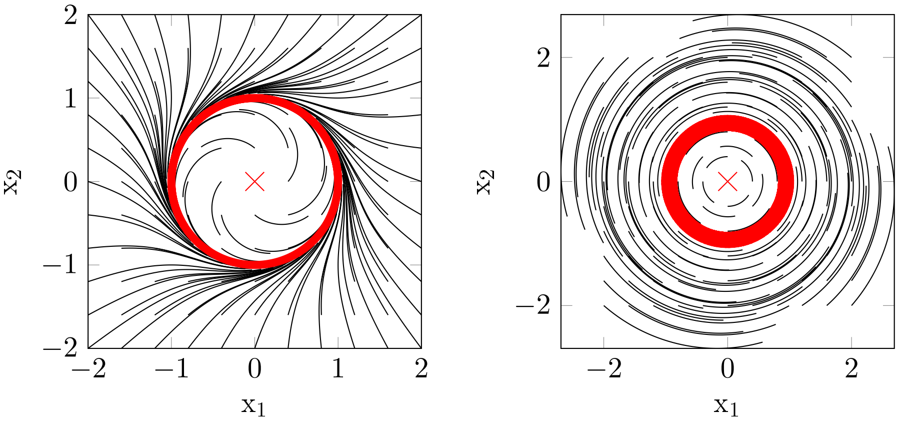

Figure 3.3: Phase diagram of the system in Cartesian coordinates, with a limit cycle and an unstable steady state (red cross). The plots were generated with 121 equally spaced initial conditions, and integrated until $t = 1$ (left plot, $\epsilon = 1$) and $t = 1 \times 10^{-2}$ (right plot, $\epsilon = 1 \times 10^{-2}$).

In this chapter, we construct a data-driven surrogate model for the dynamic of the radius in this system, and include detailed descriptions of the construction for each step. We choose this system because all information is completely known beforehand, and can be checked against the results from the surrogate construction process. The scale parameter ϵ is chosen such that the angle is the fast variable, the radius the slow variable. Apart from that, it does not have an effect on the construction of the surrogate, as the variables are decoupled. This removes the difficulty of finding a suitable coarse variable completely, even though the construction process of the surrogate model would perform this automatically. As the one-dimensional radius is the coarsest description possible, the surrogate model will also have one variable with a bijection to the radius.

Bottleneck scenario

We choose a bottleneck scenario with moving pedestrians as a second motivational example for the surrogate model. In the scenario, passengers of a train leave a wagon through its door, the bottleneck, and move past passengers waiting on the platform. Different to the limit cycle system, it is no longer possible to easily find a macroscopic model that exactly reproduces the observations from the microscopic model. The scenario was published in the context of surrogate models by the author (Dietrich, Köster, and Bungartz, 2016). A train enters a train station where passengers are already waiting on the platform. In this scenario, it is important to know the number of passengers over time, both on and off the train, because very dense situations or a sudden change of density can be dangerous. We use the Gradient Navigation Model (Dietrich and Köster, 2014) to describe and simulate the interactions between passengers and the geometry on the fine scale. The state x of the microscopic model contains all positions, velocities and parameters of the passengers. The function f is the simulator openVADERE (Vadere Crowd Simulation, 2016), with an implementation of the Gradient Navigation Model. The observables in the bottleneck scenario are the expected number of passengers on the train N_T and on the platform N_P, over time. The expectation is computed over several runs of the scenario with the same initial parameters, but different initial positions of the pedestrians on the train and the platform. The parameters p are the initial numbers of passengers on the train and the platform, $N_T(0)$ and $N_P(0)$. Given these values, we distribute passengers uniformly on the two starting areas.

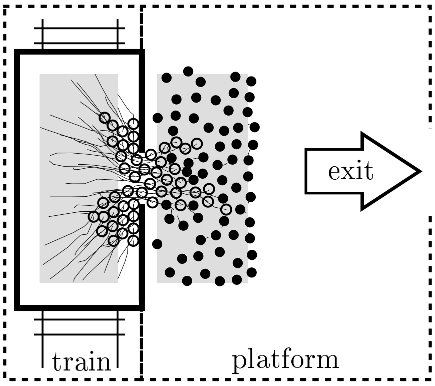

Figure 3.4: Scenario setup for the evacuation of a train. Published by Dietrich, Köster, and Bungartz (2016).

In this example, the macroscopic system cannot be constructed only by knowing the current state of the macroscopic variables: the change in number of passengers on and off the train depends on the initial number of passengers. Hence, more information is needed in order to advance in time. We construct this information through time-delay embedding theory (section 2.3). There are several possibilities to simulate the bottleneck system at the macroscopic scale. One could use equation-driven techniques such as homogenization and averaging to scale up the differential equations describing individual behavior. This would yield a macroscopic model, possibly also a differential equation, for the exit process out of the train. Another approach is taken in the natural sciences. The microscopic simulator or a live experiment generates output data, namely the number of persons on and off the train over time, starting with several different initial settings. A macroscopic model is then proposed and validated using the output. Classical surrogate models would try to approximate the output either for each time step individually, or as a response surface. For example, such a system could yield the mean evacuation time given the initial number of pedestrians inside and out of the train. For dynamic properties of the system, such as the outflow over time, an interpolation of all output would be necessary. Both equation-driven and data-driven approaches yield a macroscopic model, in most cases a system of equations. A data-driven surrogate model as constructed here yields a dynamical system that can be used in simulations. However, similar to classical surrogate models it is not present as equations, but numerical data and interpolating or approximating functions. The data for the internal variables is generated automatically through the construction process outlined in the next section. The result is either more storage efficient than classical surrogate models (see theorem 4), or more accurate while using the same amount of storage. The following assumptions are used for the train setting:

Assumption 1 Passengers follow the rules of the Gradient Navigation Model (Dietrich and Köster, 2014).

Assumption 2 Desired speeds approximately obey a normal distribution with mean $1.34m/s$ and standard deviation $0.26m/s$ (Weidmann, 1992).

Assumption 3 Initially, the positions of passengers in the train and on the platform are uniformly distributed over the starting areas. We use a five second starting phase for the distribution to settle in a state where all passengers on the platform assume their desired distances to others, and passengers in the train queue in front of the door.

Assumption 4 Waiting passengers do not strongly react to the leaving passengers, and only move away if inter-personal distances are too small. By this assumption, we exclude cooperation effects, such as the formation of a passage way in front of the door.

Assumption 5 Small differences in the initial positions of the passengers do not cause large differences in behavior on the system level. This allows us to start several simulation runs with the same initial numbers of passengers but small changes in positions on train and platform, and then average over the results.

We are interested in the number of passengers over time, both on the train and on the platform. A simulation with 40 passengers on the train and 50 passengers waiting on the platform results in the change of passengers depicted in Figure 3.5. After 16 seconds, all 90 passengers are on the platform.

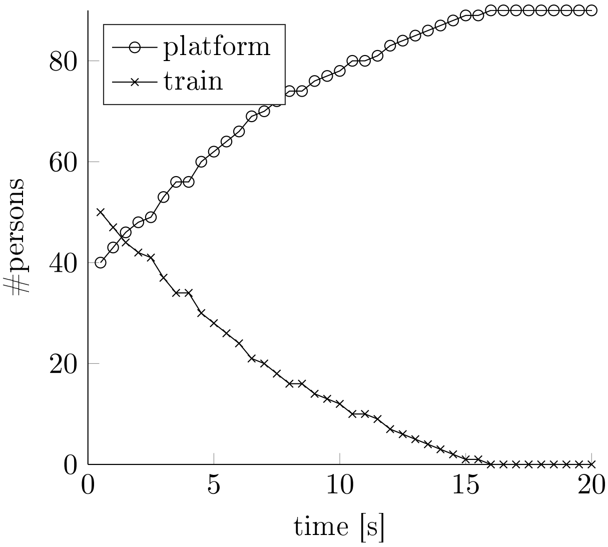

Figure 3.5: Evolution of the passenger number on the train and on the platform. 50 persons start on the train, 40 on the platform. After 16 seconds, all passengers are on the platform.

3.1.2 Construction of the surrogate model

In this section, we describe the algorithmic construction of the data-driven surrogate model. The construction process is the same for all applications shown in chapter 4. The general process is described in combination with the specifics necessary for the spiral system (equation 3.1) and the bottleneck example (figure 3.4).

1a) Choose relevant parameters The choice of parameters is one of the fundamental problems in applications. For example, in crowd dynamics, the number of people entering through specific entrances can decide whether a simulation of the crowd accurately predicts the crowd flow or is completely worthless—because in reality, people will come through other entrances and thus the crowd flow occurs in a different area. Here, we do not discuss how to choose correct parameters for what-if scenarios. For the construction of the data-driven surrogate model, we need systems which react continuously to the parameters chosen. Discrete or finite parameter spaces are possible, but not discussed here.

In the spiral example, we choose the initial positions in Cartesian coordinates as parameters p. We sample them over a grid with $11 \times 11 = 121$ equally distributed points shown in figure 3.6.

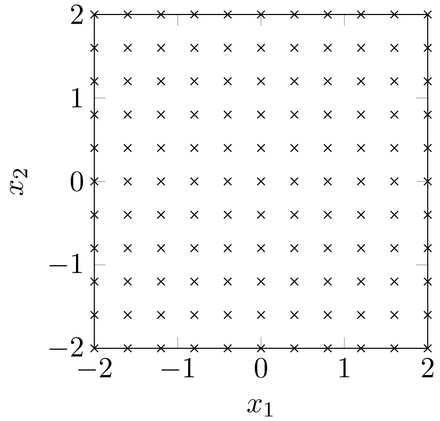

Figure 3.6: Parameter sampling in the spiral example. Each cross is a parameter sample in Cartesian coordinates (x_1, x_2), which is converted to angular coordinates (ϕ, r) and simulated through equation 3.1.

In the bottleneck example, we sample the parameters $p = (N_T, N_P)$ on a grid with $5 \times 11 = 55$ equally distributed points. To reduce the stochastic effects, we run ten simulations with randomly distributed initial positions for each value p, and then compute the average values for $N_T(t)$ and $N_P(t)$ for each individual time t over the ten simulation runs. See section 3.2.7 for a classification of stochastic effects.

1b) Choose relevant observables Choosing the right observables for a given system is the fundamental problem for a successful transitioning of scales. However, for the data-driven surrogate model discussed here, this step concerns the choice of observables *relevant to the application.* A safety officer might be interested in crowd density and flow, but these variables might not hold enough information to advance in time. This is the closure problem discussed in section 2.3. Ultimately, the surrogate model needs intrinsic variables which hold enough information to *advance in time*, which is guaranteed through time delay embedding. We call these intrinsic variables *closed observables.* The observation function then reconstructs the *relevant observables* from the closed observables.

The relevant observable we choose for the spiral example is the radius of the spiral. We compute it by first observing the system in Cartesian coordinates (x_1, x_2) over time, and then computing $r(t) = \sqrt{x_1(t)^2 + x_2(t)^2}$. The data we build the surrogate model from is computed through the whole system (3.1), but only r is stored and processed further. In this particular system, the angle and the radius are independent systems. In general, the variables are not separated, and the observation function is a combination of them, such as the total number of people in a certain area, or the current mean speed. In fact, by observing the system in Cartesian coordinates, we already have such a combination of variables, which is then disentangled again through the computation of the radius. If we observed the angle instead of the radius, the surrogate model would capture the circular behavior with constant speed, on a circle with constant radius. The steady state at $r = 0$ would enter as a bifurcation, as there is no circular behavior when measured in Cartesian coordinates.

In the bottleneck example, we observe two numbers N_T and N_P counting passengers on the train and passengers on the platform. The observer function y on the current state x yields $y(x) = (N_T, N_P) \in \mathbb{N}_0^2$. In case passengers on the train can all leave onto the platform, N_P will equal the sum of the initial values $N_T(0)$ and $N_P(0)$ at the end of the scenario. If there are too many passengers waiting on the platform at the beginning, not all passengers on the train will be able to leave.

1c) Generate observation data After the parameters and observation function are chosen, the microscopic model is used to generate data. Alternatively, the data can come from physical experiments—where all challenges with experiments apply, such as control of variables and environmental conditions.

In the spiral example, we sample the radius for all 121 initial conditions and a time span of $T = 1 \times 10^{-3}$ seconds with $N = 10$ points in time. We do not need long trajectories to capture the dynamic in this example, because no stochastic effects play a role, the system is effectively one-dimensional, and smooth. In total, we generate $121 \times 10 = 1210$ data points (see figure 3.7).

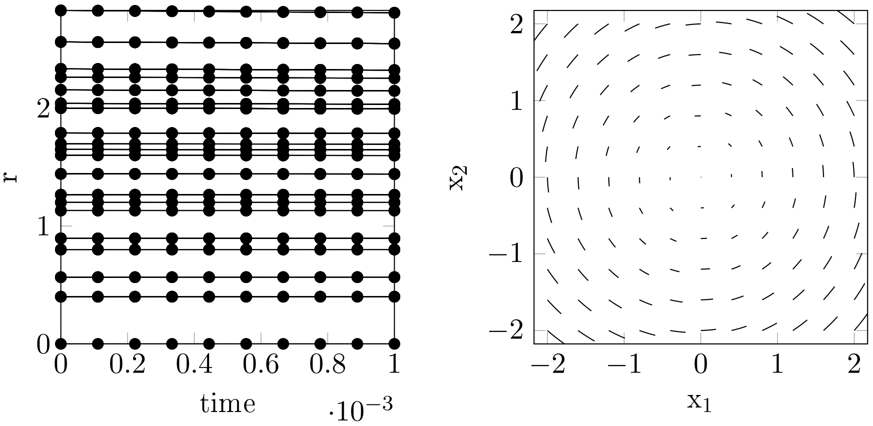

Figure 3.7: Left: the observed data in the spiral example is the radius of the spiral over time. Right: the trajectories including the angle until $t = 1 \times 10^{-3}$, starting at the initial conditions, with scale parameter $\epsilon = 1 \times 10^{-2}$.

In the bottleneck example, we run the simulation over 50 seconds, for the parameter values in table 3.1. If there are more than 100 number of passengers on the platform at the beginning, not all passengers can leave the train.

Parameter	**Values**
$N_T(0)$: Initial numbers of passengers on the train	10,20,30,40,50
$N_P(0)$: Initial numbers of passengers on the platform	0,20,40,60,80,100,120,140,160,180,200
Runs per initial value pair (N_P, N_T)	10

Table 3.1: Parameters for the simulations of the bottleneck example. In total, $5 \cdot 11 \cdot 10 = 550$ simulations were started. In the example, observations were one second apart.

1d) Time-lagged embedding of the data Time delay embedding is needed to ensure the closure of the internal variables of the surrogate model. For a reconstruction of a state space of dimension n, at least $2n+1$ time lags are necessary (also see section 2.3). If more than $2n+1$ delays are used, the ambient dimension of the reconstructed space is higher than necessary, but the intrinsic dimension is still the same. Delay embedding also has a smoothing effect on the new observables. Also, adding more delays does not harm the parametrization with diffusion maps, as they are not affected by the ambient dimension of the points they act on. The fact that the intrinsic dimension is stable, and that diffusion maps are not affected by the ambient dimension, permits to choose more than the $2n+1$ delays. This is crucial to applications where the dimension of the state space is unknown, and has to be guessed beforehand to compute the number of necessary delays. The available data is usually fixed, and choosing more delays cuts off more data at the end of the trajectories. This can be circumvented by using the last data point in every trajectory and concatenating it $2n+1$ times to the end of the trajectory. From a dynamical systems perspective, this introduces a steady state at the end of each trajectory, which is not always favorable. It is not a problem in scenarios that end in a steady state anyway, such as the bottleneck scenario, where nothing changes after all train passengers have left the station. In the spiral example, there are even two steady states for the radius, $r = 1$ and $r = 0$, and all trajectories tend to one of them depending on the initial state ($r_0 = 0$ or $r_0 \neq 0$).

The radius of the spiral already contains enough information to advance in time, which can be seen from equation 3.1: only the radius itself is used to compute its time derivative. We still use five delays in time here, to demonstrate the delay embedding and the different projections between observed space, delay space, and diffusion map space. The delays are separated through $\Delta t = \frac{T}{N-1} = \frac{1 \times 10^{-3}}{9}$. Each five-dimensional delay vector in the spiral example has the form

$$\mathbf{H}(r(t_k)) = \left[e^{-4\kappa} r(t_k), e^{-3\kappa} r(t_k + \Delta t), \ldots, r(t_k + 4\Delta t)\right], \tag{3.2}$$

with $t_k = k\Delta t$, $k = 1, \ldots, 5$, and different initial conditions $r_0 = r(0)$. We choose $\kappa = 1$ to project onto the most stable Oseledets subspace (see section 2.3). This subspace contains all information about the system, since only the one-dimensional radius is necessary to compute future values. The parameter κ has to be chosen in the interval $[0, -\sigma_1]$, where $\sigma_1 < 0$ is the Lyapunov exponent associated with the most stable Oseledets component of the system (Berry et al., 2013). In practice, the Lypapunov exponents are unknown and normally hard to estimate. In general, large values (> 1) of κ are associated with strongly contracting systems, and small numbers ($\ll 1$) with weakly contracting systems. Also, choosing κ closer to zero projects strongly onto the most stable Oseledets subspace, which regularizes the dynamic but also smoothes effects from more unstable subspaces which might be needed to reconstruct the observation data later. Table 4.1 lists all values of κ used in the examples throughout the thesis, and serves as a reference for similar systems. We generated ten points in time for each trajectory, therefore five new delay vectors are generated for each initial condition. In total, we use $121 \times 5 = 605$ delay vectors to form the delay space.

The bottleneck scenario has two parameters, and we observe two variables N_T and N_P over time. If we assumed that the two variables already contained enough information to compute future values, and the parameters both independently influenced the dynamic, we needed a four-dimensional state space for the surrogate model. However, the values of N_T and N_P might not be the optimal choice as state variables, or there are even more than two variables needed to find future values. To make sure that enough information is present in the state space of the surrogate, we assume that the hidden, macroscopic model operates on five state variables. The number five is an informed, conservative guess: it is “more than double the number of observations”, that is, $2m + 1 = 5 = n$. The dimension of the parameter space P is two, as we vary the numbers $N_T(0)$ and $N_P(0)$ to generate trajectories. In total, we thus need $2(\dim P + n) + 1 = 2(2 + 5) + 1 = 15$ time lags in the construction. This is a very conservative assumption, and in fact, the number of state variables needed for the surrogate is only two, a result computed in the next step.

1e) Manifold learning with diffusion maps The delay vectors usually are elements of a high-dimensional space, because the dimension of the delay vectors is the product of the dimension of the individual observations and the number of delays. The high dimension of the delay vector space is only an ambient dimension of the actual manifold the points lie on. We find the intrinsic dimension, as well as a parametrization of the manifold using diffusion maps (see section 2.2.2). To construct the diffusion map, we compute the distance matrix

$$D_{ij} = \|\mathbf{H}^i - \mathbf{H}^j\| \tag{3.3}$$

between all delay vectors $\mathbf{H}^i$, $\mathbf{H}^j$, where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^n$. Then, we construct the diffusion map with the following algorithm (Berry et al., 2013), where the parameter ϵ is chosen larger than the median of all $D_{i,j}$ (see parameter table 4.1).

Diffusion map algorithm for non-uniformly sampled data:

1. Form the transition matrix $W_{i,j} = \exp(-D_{i,j}^2/\epsilon^2)$.
2. Normalize density with parameter $\alpha = 1$:
    (a) Form the symmetric matrix $J = (W + W^T)/2$.
    (b) Form the diagonal normalization matrix $P_{ii} = \sum_j J_{ij}$.
    (c) Use P to form the kernel matrix $K = P^{-\alpha} J P^{-\alpha}$.
    (d) Form the diagonal normalization matrix $Q_{ii} = \sum_j K_{ij}$ and its inverse Cholesky factors $Q^{-1/2}$.
3. Solve the eigensystems $Q^{-1/2} K Q^{-1/2} \mathbf{v}_k = \lambda_k \mathbf{v}_k$ for all k, then sort λ_k by absolute value.
4. The diffusion map coordinates depend on $\mathbf{H}^i$ and can be computed through $\psi_k(\mathbf{H}_i) = \lambda_k Q^{-1/2} \mathbf{v}_k(\mathbf{H}^i)$, such that the diffusion map is

$$\Psi : \mathbf{H}^i \mapsto (\lambda_2 \psi_2(\mathbf{H}^i), \lambda_3 \psi_3(\mathbf{H}^i), \dots).$$

The first eigenvector ψ_1 only contains ones and is omitted. We also truncate the map for small λ_k and remove all ψ_k that are harmonics of the previous eigenvectors. In general, removing harmonics of previous eigenvectors is no trivial task, especially in higher dimensions. With the idea of a surrogate model constructed on the eigenvectors, there is a simple algorithm to remove harmonics of eigenvectors.

Note that the number of combinations in the selection step grows rapidly. But as the simulation with the constructed model is very fast (in the order of milliseconds), we can choose n to be between 10 and 20. An even easier—but less automated—method to select the eigenvectors is to look at plots relating u_2 and u_i (see figure 3.8, where u_3 and u_4 depend on u_2). If they are independent of each other, many values of u_i are stacked on individual values of u_2. This procedure can of course only be applied in cases where there are only two independent eigenvectors.

In all examples and applications discussed here, we can remove harmonics through visual inspection (see figure 3.8). The resulting variables $z^i = \Psi(\mathbf{H}^i)$ are the closed observables. In the spiral system, there is only one relevant eigenvector of the diffusion map, ψ_2. The eigenvector ψ_2 parametrizes the manifold composed of the delay vectors in the five-dimensional delay vector space from the previous construction step. The manifold must be a curve, because only one vector is necessary to parametrize it.

Algorithm to remove dependent eigenfunctions:

1. With the diffusion map algorithm described in section 3.1.2 compute the first n components of the diffusion map, where n is so large that all important, independent eigenvectors are included. As the eigenvectors are sorted by the absolute value of the associated eigenvalues, we can use $u_2, u_3, \ldots, u_n$. The eigenvector u_1 is trivial. The eigenvector u_2 is always included in the set of important eigenvectors.
2. Select k out of $n - 1$ remaining eigenvectors and construct a numerical model, using a linear interpolation method in $k + 1$ dimensions.
3. Simulate a suitable number of time steps with an input parameter set not used in the construction. Compute the error of the output to data from the original model computed with the same input parameter set.
4. Select all sets of eigenvectors that produce a low error (the actual value depends on the original model and scenario).
5. Sort the sets by the number of eigenvectors, and choose the set that has the smallest number of eigenvectors.

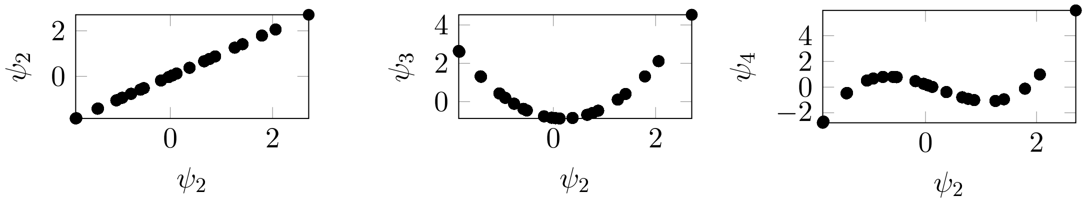

Figure 3.8: Eigenvectors ψ_2, ψ_3, and ψ_4 of the diffusion map plotted against the first nontrivial eigenvector ψ_2. Since only ψ_2 is relevant in the parametrization of the manifold, ψ_3 and ψ_4 are harmonics of ψ_2.

We now interpret the values of the eigenfunction ψ_2 as the values of the intrinsic variable z_1 of the surrogate model. Generally, we call the variables z_i *closed observables*. For the spiral example, we need only one closed observable, which we call $z := \psi_2$. Figure 3.9 shows the trajectories of the new system in diffusion map space. In the spiral example, the radius is one-to-one with z, and thus the trajectories in the new space show the same behavior as the radius (compare figure 3.7). If we plot z against its derivative in time (right plot in figure 3.9), we see that the system is still closed—the time derivative of z can still be computed given only z.

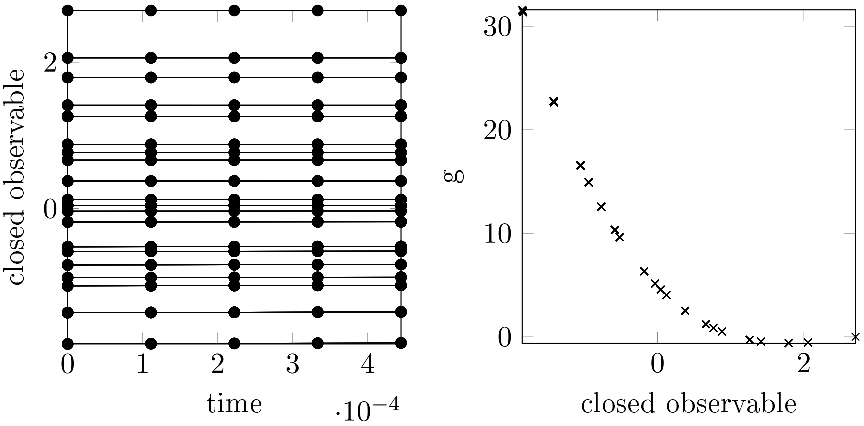

Figure 3.9: Left: the closed observable z plotted for each of the 121 initial conditions over time. There are only eleven distinct trajectories, and the decay over time is not visible because the time interval is too small. Right: the function $g(z) = (z(t-\Delta t) + z(t+\Delta t))/(2\Delta t)$, evaluated at all computed points $z(t)$ for all parameters.

In the bottleneck example, two eigenvectors of the diffusion map are needed to reproduce the observed values to an acceptable accuracy. The stochastic nature of the system on the macroscopic scale cannot be fully eliminated in this example, thus the eigenvectors of higher order also capture information mostly resulting from this stochasticity.

1f) Form the surrogate model by interpolation of the new variables After the delay manifold is parametrized with the diffusion map, we can construct the initial value map z_0, the dynamic g, and the observation function $\tilde{y}$ for the surrogate model. The construction problem is similar to the construction of a model from experimental data. However, because the generation was performed using time-delays and diffusion maps, all data points are guaranteed to contain enough information to advance in time, and have a very low dimension. Closure and low dimension enable us to use simple interpolation and approximation procedures for surrogate models of many different data sources, as will be shown through applications in chapter 4. We use $I[\cdot]$ as a generic interpolation procedure for the given arguments.

1. The coordinate transform z_0 between the parameters p and the initial values $z(0)$ is of the form
$$z_0(p) = I[\Psi(\mathbf{H}(x_0(p))],$$
where $\Psi(\mathbf{H}(x_0(p))) = \Psi(y(x_0(p)), \dots, y(f^{T-1}(p, x_0(p))))$.
2. The dynamic g is an interpolation of the derivative of $z(t)$:
$$g(z(t)) = I\left[\frac{z(t+\Delta t) - z(t)}{\Delta t}\right],$$

where the mapping $z \mapsto g(z)$ is well-defined through the closure guarantee of Takens' theorem (see §2.3). Note that the interpolation of the dynamic can also take more values over time into account to be more accurate.

3. The observer function $\tilde{y}$ of the closed observables is an interpolant of the observations in the original system:

$$\tilde{y}(z(t)) = \tilde{y}(\psi(\mathbf{H}(t))) = \tilde{y}(\psi(y(x(t)), y(x(t+1)), \ldots, y(x(t+T-1))) = I[y_n]$$

For the closed observable z of the spiral, we choose a polynomial of degree three to approximate the dynamic g and the observer function $\tilde{y}$. The initial value map z_0 maps from (x_1, x_2) to the closed observable $z(0)$ and is approximated through a linear, two-dimensional interpolant (MATLABs `griddedInterpolant` function). Figure 3.10 shows all three interpolants together with the data.

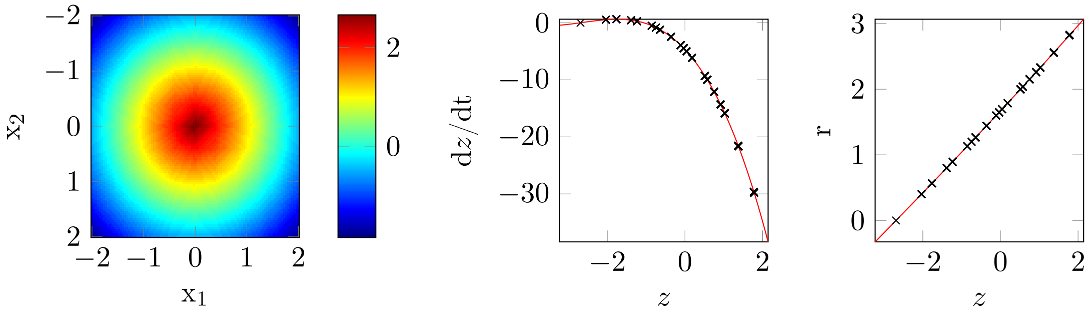

Figure 3.10: Data and associated interpolants for the surrogate model of the radius in the spiral example. The left plot shows the initial map from (x_1, x_2) to the intrinsic variable z. The center plot shows the dynamic g of the surrogate model, mapping z to $\frac{dz}{dt}$. The right plot shows the observer function of the surrogate model, from z back to the radius r, visualizing the bijection of z and r present in this example.

The bottleneck example needs at least two state variables z_1, z_2. This can be seen in figure 3.11, where a two-dimensional polynomial is fitted to the data points (z_1, z_2).

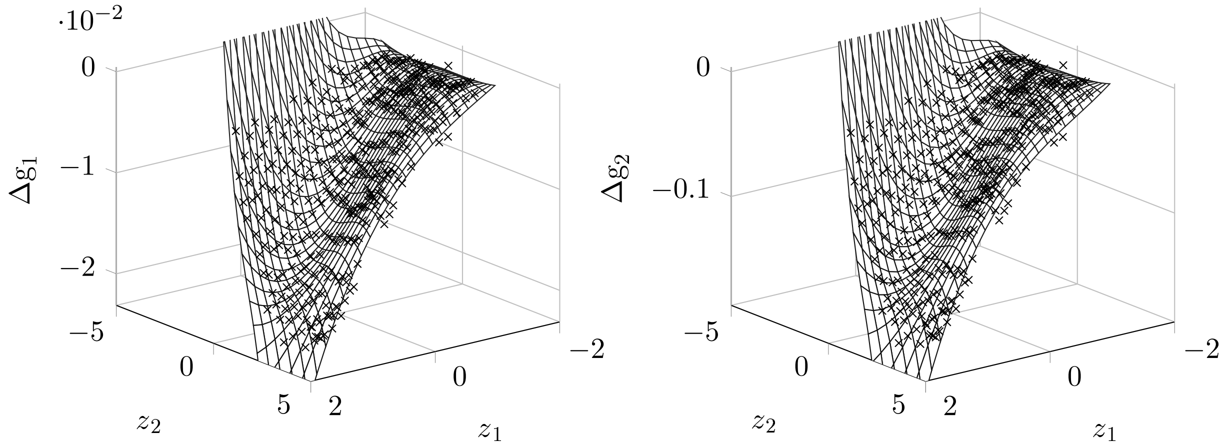

Figure 3.11: Evolution function $\Delta g(z_1, z_2) = (\Delta z_1, \Delta z_2)$ in the form of two functions $\Delta z_1 = g_1 - z_1$, $\Delta z_2 = g_2 - z_2$ on closed observable coordinates z_1 and z_2. The shape of the two functions is quite similar, but their scale (z-axis) is different.

3.1.3 Simulations with the surrogate model

After the construction steps, the numerical model can be used for simulation and analysis, independent of the original system. With the following steps, the surrogate model can be used in a simulation (see figure 3.2):

2a) Choose parameters $p = (p_1, \dots, p_P)$.

2b) Compute initial states $z(0) = z_0(p)$ of the closed observables.

2c) Iterate the given observables with the constructed dynamic, such that $z(t+1) = g(z(t))$, or solve $\frac{d}{dt}z = \Delta g(z)$ numerically, where Δg is a second-order approximation of the gradient. In all applications discussed in this thesis, we solve the latter, continuous system rather than the former, discrete system. Table 4.1 lists all approximation methods used for the examples.

2d) Observe the states $z(t)$ with the constructed observer function $\tilde{y}$ to generate observations of the original system:

$$\tilde{y}(z(t)) \approx y(x(t)). \tag{3.4}$$

The simulation with the numerical model is independent of the original system, because it can generate the observed values over time without queries to the original. The independence of the two systems enables a pre-computation of the numerical model on a high-performance system or with a large array of experiments, which is then condensed into the numerical model that can be used on a much less powerful device, such as a smartphone.

In the spiral system, the dynamic of the radius can be reconstructed from the Cartesian initial coordinates given as parameters. Figure 3.12 shows good agreement to analytically derived trajectories of the radius, here

$$r(t) = \exp(t) / \left(\exp(2t) - 1 + 1/r_0^2\right). \tag{3.5}$$

The data used to construct the surrogate model originates from simulations up to $t = 1 \times 10^{-3}$ (see part 1.c in section 3.1.2). The simulations performed here are over a much larger time span, which demonstrates the power of the concept of an interpolated dynamic. By sampling the parameter space and only performing short simulation runs, we effectively sample the dynamic of the underlying system, which is then interpolated. As soon as the interpolation has converged up to a certain accuracy, we are able to reproduce the time evolution of observations over an arbitrary time horizon. We discuss this convergence further in section 3.2.3. Since only the radius was chosen as a relevant observable, the angle cannot be reconstructed with the surrogate model.

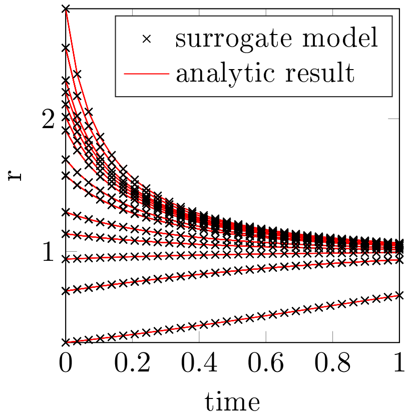

Figure 3.12: Comparison of the radius generated through simulation with the surrogate model, and analytically derived trajectories. The initial conditions are spread over a 10×10 grid of $[-2, 2]$, different from the grid used during the construction of the surrogate model.

In the bottleneck example, the trajectories can only be reproduced to a certain accuracy level. This is because we only use two dimensions for the closed observables z, and use low-order polynomials as approximation functions. Figure 3.13 shows the number of passengers on the platform as predicted by the surrogate model, compared to three simulation runs with the same parameters.

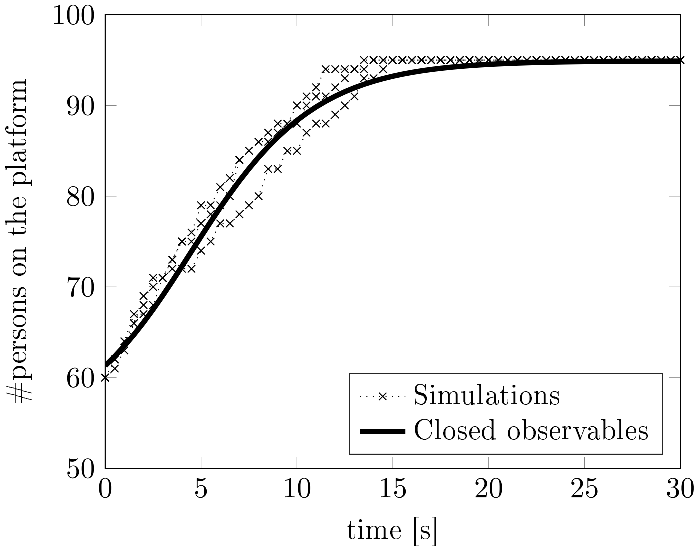

Figure 3.13: A prediction (bold line) of the average number of persons on the platform by the constructed model, compared to three simulation runs (dotted). The incorrect starting values for the prediction results from interpolation errors: the given initial values $(N_P(0), N_T(0)) = (60, 35)$ are transformed by the diffusion map Ψ into the reduced delay-space space $\Psi(H(M))$, and after the simulation recreated by observation with $\tilde{y}$.

3.1.4 Summary

In this section, we described a construction process for a data-driven surrogate model able to capture the dynamics of a given process. We defined a *model* generically as input-output system, with input parameters mapping to initial states, a map acting on states to advance in time, and an observation function to create the desired output of the system. The surrogate model is also defined as an input-output system. Its construction involves a numerical estimation of the state space manifold of a dynamical system. This step is crucial for the data-driven surrogate model. We showed how to construct the state space, and how to create the interpolation of the map from parameters to initial states, the map to future values, and the observation function. All of the construction steps were detailed in general, as well as with their application to two example systems.After the construction process, we showed how to simulate the surrogate model. This simulation is now on the scale determined by the system underlying the observation, and not the microscopic system given at the beginning.

It is not clear that the construction process for the data-driven surrogate model actually produces a system able to reconstruct the original observations, and that it is worth constructing the surrogate model instead of storing all output and interpolating it. This will be shown in the next section.

3.2 Analysis of dynamic surrogate models

The surrogate model structure developed in the previous section can be analyzed with respect to numerical errors, stochastic effects, and infinite-dimensional observer and parameter spaces. As the definition of the intrinsic dynamic of the surrogate model is very generic, it is possible to use very elaborate methods to approximate it without a need to adapt the theorem proved on numerical errors. We elaborate on several advanced methods for the dynamic and the impact on their specific errors. The analyses of numerical errors and storage have been published by Dietrich, Köster, and Bungartz (2016). The microscopic system has the form

$$\begin{aligned} x_0 &= x_0(p), && (3.6)\\ x(t+1) &= f(p, x(t)), && (3.7)\\ y(t) &= y(p, x(t)), && (3.8) \end{aligned}$$

with time t, parameters p, initial map x_0, dynamic f, and observer y (also see figure 3.1). The surrogate model has the same form and input parameters p, and we call the intrinsic variables z, the initial map z_0, the dynamic g, and the observer $\tilde{y}$. The functions f and g are sometimes not used as a map, but as the time derivative of x and z, respectively. That change is explicitly stated for all cases.

3.2.1 Motivating example: spiral and bottleneck revisited

The construction process for the surrogate model was stated in the previous section, along with two motivating examples: the limit cycle system and the bottleneck scenario. Here, we use these two examples again, to demonstrate and motivate the theoretical results that follow. Firstly, we generate observations of the limit cycle system with varying sampling density over space and time, where the analytic solution of the system allows to analyze the numerical error. This error depends on all three interpolants, and we will prove the general form of this dependence in section 3.2.3. Secondly, we use the bottleneck scenario as an example for a stochastic term introduced on the macroscopic scale by unresolved microscopic effects. The example also demonstrates the *assumption of a macroscopic model*, a notion which we will make precise in the next section.

The surrogate model for the radius in the limit cycle system is given by the functions $z_0 : \mathbb{R}^2 \to \mathbb{R}$, $g : \mathbb{R} \to \mathbb{R}$, and $\tilde{y} : \mathbb{R} \to \mathbb{R}$, such that

$$\begin{aligned} z(0) &= z_0(x_1, x_2)\\ \tfrac{d}{dt} z(t) &= g(z(t))\\ y(x_1(t), x_2(t)) = r(t) &= \tilde{y}(\phi(t)). \end{aligned} \qquad (3.9)$$

In this example, the initial map z_0 is the only function of the surrogate model that does not map from $\mathbb{R}$ to $\mathbb{R}$. We observe the radius through $y(x_1, x_2) = \sqrt{x_1^2 + x_2^2} = r$. As this makes observations independent of the angle, it is only necessary to sample points for a particular angle. We choose the axis $x_2 = 0$, so that z_0 is a function of one variable. A piecewise cubic interpolation scheme is used for all functions (MATLABs `interp1` with the `pchip` option). We approximate g by two schemes, forward differences (a first order scheme) and centered differences (a second order scheme), so that for $\Delta t := t_1 - t_0$,

$$g_{\text{f.d.}}(z(t_i)) = \frac{z(t_{i+1}) - z(t_i)}{\Delta t} + O(\Delta t), \tag{3.10}$$

$$g_{\text{c.d.}}(z(t_i)) = \frac{z(t_{i+1}) - z(t_{i-1})}{2\Delta t} + O(\Delta t^2). \tag{3.11}$$

The parameter x_1 is sampled by N_x points, and the trajectories are sampled in time by N_t points. The total error $e(N_t, N_x)$ for a surrogate model is computed through the average over all parameters and trajectories, $e(N_t, N_x) = \frac{1}{N_t N_x} \sum_i \sum_k [(\tilde{y} \circ g^k \circ z_0)(x_i)]$. Figure 3.14 shows the total error for both differencing schemes, in case one of the numbers N_x or N_t is kept constant. The order of the two differencing schemes is visible in the left plot. Since we employ the same interpolation scheme for all functions, the right plot does not show a qualitative difference when changing the number of samples in space N_x. The centered differencing scheme still shows better quantitative performance, which will be made precise through theorem 3.30 below. Figure 3.15 shows the two-dimensional plots, where both N_x and N_t are varied. As expected, the total error decreases with increasing number of samples, where a change in N_t affects the error more drastically than a change in N_x.

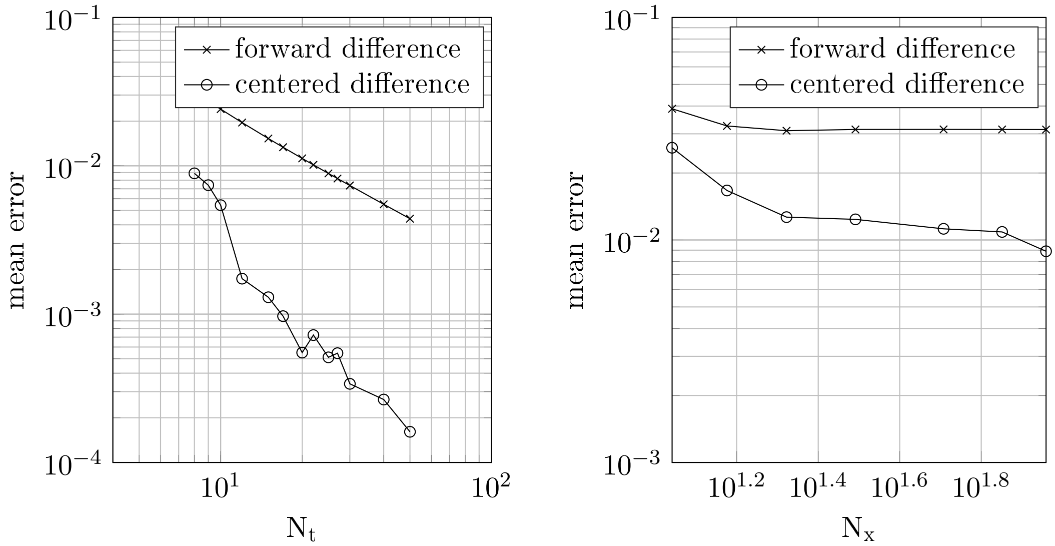

Figure 3.14: Mean error of the limit cycle example for different numbers of samples in time (left plot, at $N_x = 91$) and space (right plot, at $N_t = 50$), for forward and centered difference approximations of the dynamic g.

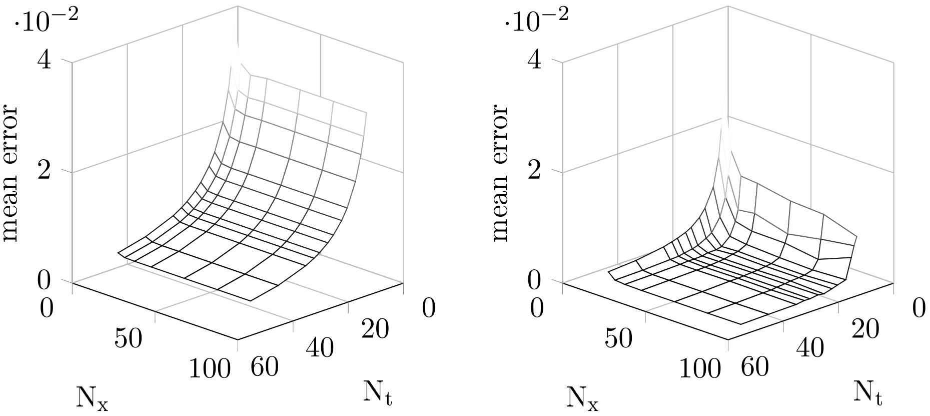

Figure 3.15: Mean error of the limit cycle example for different numbers of samples in time and space, for forward (left plot) and centered difference (right plot) approximations of the dynamic g.

The analysis of the numerical error in the limit cycle system is easily possible, because an analytic solution is known and can be compared to the numerical results. In the bottleneck scenario, where a simulation of 200 pedestrians moves them from one side to the other, no analytic solution is known either on the microscopic or on the macroscopic scale. Still, we have an analytic model on the microscopic scale—in this case, the gradient navigation model. We use it in the simulation to generate the number of pedestrians in front and after the bottleneck, which serve as the two observations on the macroscopic scale. In fact, the macroscopic observations are defined as the *expected* number of pedestrians for each time step. While the positions of the pedestrians in the scenario are uniformly distributed over the starting areas, the exact expectation is no longer be stochastically distributed, and hence the macroscopic model—in theory—is deterministic. However, we can only approximate the expected values numerically by the empirical expectation, which is a random variable. The stochastic component stem from averaging a finite number of simulation runs, and affect the numerical approximation of the surrogate model. In section 3.2.7, we discuss several of these effects.

3.2.2 Equivalence of the surrogate and the hidden macroscopic model

It is not obvious that the construction and simulation processes lead to a surrogate model that is correctly approximating the data observed from the microscopic model. We proof this here, using the theorems and tools discussed in section 2.4. The theorem stated in this section relates the observations of the original system shown by figure 3.1 to the observations by the data-driven surrogate model. In the limit of infinite data, the observations are shown to be the same for arbitrary choice of parameters. For a fixed value of parameters, the theorem can be proofed directly through Takens' theorem. The crucial part of the proof concerns the treatment of the parameters, since the map of the surrogate model does not depend on them, but still generates the same observations.

The proof of the theorem considers three dynamical systems and their relation. The first system is the *original, microscopic system.* We do not assume knowledge of the model equations, but the system must be able to generate deterministic observation data over time after the initial parameters have been set. A relaxation of the assumption of deterministic data is discussed theoretically in section 3.2.7, and demonstrated through applications such as crowd flow through a bottleneck, queuing, and granular flow (sections 4.1.1, 4.1.2, and 4.3). The second system is the *hidden, macroscopic system.* We assume that this system exists, which ensures that an equation-driven homogenization from the original to the macroscopic system would be possible—that is, if the equations of the original system were known. Data-driven methods would usually need the observations to be diffeomorphic to the state space of the hidden system, so that it is possible to define a map on the observations directly. Even if such a map can be defined, approximating it numerically is difficult due to the curse-of-dimension, as the dimension of a single observation can be very high. For example, an observation of density in a grid of 100×100 cells would already have dimension 10000. We also assume that given the same parameters, the hidden system can generate the same observations as the original system. We do not specifically assume that the macroscopic system is different to the original system, but for a numerical approximation, it is necessary that either its state variables have a low dimension, or the dynamic and observation function are both linear. If the original system is a system of many particles or agents, the low dimension of the macroscopic system can result from observing quantities of the original system that change slowly compared to the microscopic time scales. Typical observables to achieve this are time or space averages, applications 4.1.1, 4.2, and 4.1.3 demonstrate this. The third system is the *data-driven surrogate system.* It is constructed through the steps outlined in section 3.1.2. Here, we show that the surrogate system approximates the hidden system so that the systems generate the same observations in the limit of infinite data.

The following definitions are used in the main theorem of this section, and its proof. The dynamical system creating the observations is defined at the beginning of this chapter (equation 3.6–3.8), but we repeat its definition in equations 3.12–3.14 for an easier comparison to the other two systems below. The *original, microscopic system* is a dynamical system with time $T = \mathbb{N}_0$, a compact manifold X, the state space, and a map $f : P \times X \to X$, parametrized through elements of a compact manifold P, the parameter space. The states $x \in X$ can be observed through an observation function $y \in C^2(X, Y)$, where $Y \subset \mathbb{R}^{d_Y}$ is a compact subset of d_Y-dimensional Euclidean space. We require compactness of all sets to ensure a discrete and countable basis of the diffusion operators, which removes the complexity of the extensions to non-compact manifolds (Berry and Harlim, 2016). The observation functions y must fulfill the genericity conditions of Takens theorem (see section 2.3). The relations in the original system are given through

$$\begin{aligned} x_0 &= x_0(p), && (3.12)\\ x(t+1) &= f(p, x(t)), \text{ and} && (3.13)\\ y(t) &= y(p, x(t)). && (3.14) \end{aligned}$$

The map f and the observation function y can both be affected through the parameters p. Given microscopic system (3.12–3.14), we assume existence of a low-dimensional system on the macroscopic scale, which can also generate the observations $y(x(t))$. As we never explicitly construct this system, we refer to it as *hidden system.* We define the hidden system analogous to equations 3.12–3.14, with a compact state space manifold $W \subset \mathbb{R}^{\dim W}$, map $h : P \times W \to W$, and observation function $\hat{y} : W \to Y$. The observation function maps into the space Y of observations from the original system, and generates the same observations as the observation function y. The relations in the hidden system are given through

$$\begin{aligned} w(0) &= w_0(p), && (3.15)\\ w(t+1) &= h(p, w(t)), \text{ and} && (3.16)\\ y(t) &= \hat{y}(p, w(t)). && (3.17) \end{aligned}$$

Finally, the *surrogate system* is defined with a state space Z, map $g : Z \to Z$, and observation function $\tilde{y} : Z \to \tilde{Y}$, where $\tilde{Y} \subset \mathbb{R}^{d_Y}$, $\dim \tilde{Y} = \dim Y$, such that

$$\begin{aligned} z(0) &= z_0(p), && (3.18)\\ z(t+1) &= g(z(t)), && (3.19)\\ \tilde{y}(t) &= \tilde{y}(z(t)). && (3.20) \end{aligned}$$

Unlike the maps f, h and observation functions $y, \hat{y}$ from the original and the hidden system, the map g and observation function $\tilde{y}$ from the surrogate do not depend on the parameters p. This is crucial for the storage efficiency of the numerical approximation, since a direct approximation would need to include the dimensions of the parameter space P. The parameters can affect the values of $y(x(t))$ over time, and if so, the dimension of Z increases. Savings in storage occur when parameters depend on each other, such that the effective dimension of P in relation to the dynamic f could be lower. In this case, we call the parameter space P *degenerate.* The construction process of the surrogate automatically projects the data to a lower dimension. A simple example for a degenerate parameter space is a system with a dynamic $f(p_1, p_2, x) = p_1^2 p_2 x$, for parameters $(p_1, p_2) \in \mathbb{R}^2$, $x \in \mathbb{R}$. The parameter space is two-dimensional, but the dynamic only depends on a one-dimensional image $Q(p_1, p_2) = p_1^2 p_2 \in \mathbb{R}$ of the parameter space. In this case, f could be replaced by $f_Q(q, x) = qx$, $q \in \mathbb{R}$, and the system would still show the same dynamics for the parameter $q = Q(p_1, p_2)$. We call P the *degenerate parameter space*, and the image $Q(P)$ the *effective parameter space.* The map Q from the parameter space of the original model to the effective parameter space is important in the proof of Lemma 1 leading up to Theorem 2, where we show that the surrogate model is an approximation of the hidden macroscopic model. The initial map z_0 of the surrogate model is closely tied to the map Q: z_0 maps from a—possibly degenerate—parameter space into the intrinsic space of the surrogate model. The intrinsic space is not degenerate anymore, as any two trajectories starting from two different parameter values p, p', but the same image $Q(p) = Q(p')$, are mapped onto the same trajectory in the intrinsic space Z. This is why constructing the surrogate model factors out the dynamics generated by the degenerate parameter space, a statement that is made precise in Lemma 1.

Next, we state and proof the main contribution of this section: in the limit of infinite data, given the same value of the initial parameters p and a fixed number of time steps t, the observations of the surrogate system and the original system are the same.

Theorem 2. *For any fixed time $t \in T$, $T \subset \mathbb{N}_0$, and fixed parameter $p \in P$, the observations between the original system and the surrogate are equal, such that*

$$\|(\tilde{y} \circ g^t \circ z_0)(p) - (y \circ f^t \circ x_0)(p)\|_Y = 0. \tag{3.21}$$

The superscript t denotes t applications of the given function, and $\|\cdot\|_Y$ denotes the norm on Y.

To proof this theorem, we first ensure that a diffeomorphism between the delay space and the hidden state space exists. Then, as a corollary, we find that the function g exists and maps variables in a new parametrization of delay space to future values. Finally, we proof that the observation function $\tilde{y}$ exists.

Lemma 1. *There exists a diffeomorphism $d : Q(P) \times W \to Y^{(2(\dim Z+\dim P)+1)}$ from the product space of effective parameters $Q(P)$ with the state space manifold of the hidden system W to the delay vector space Y of the observed values.*

Proof. We form a delay embedding of the observations $y(x(t)) \in Y$ into a space

$$Y^{(2(\dim Z+\dim P)+1)} = \underbrace{Y \times Y \times \cdots \times Y}_{2(\dim Z+\dim P)+1 \text{ times}}, \tag{3.22}$$

such that we have vectors

$$[y(x(t)), y(x(t-1)), \ldots, y(x(t-2(\dim W + \dim P))] \in Y^{(2(\dim W+\dim P)+1)}. \tag{3.23}$$

First, we consider the case $Q(P) = P$, that is, the hidden model depends on the full parameter space P with no hidden dependencies between the different dimensions of P. We consider the parameters of the map $h : P \times W \to W$ in the hidden system as additional state variables. This way, the observed values $y(t) = \hat{y}(p, z(t))$ of the hidden system, when formed into the delay vectors (3.23), will form a diffeomorphic copy of the original space $P \times W$ through Takens' theorem. This implies existence of a diffeomorphism $d : P \times W \to Y^{(2(\dim W+\dim P)+1)}$.

Second, the functions h and $\hat{y}$ of the hidden system might effectively depend on fewer than $\dim P$ parameters q, with a functional relation to the given parameters $p \in P$, such that $q = Q(p)$, and $\dim Q(P) < \dim P$. In this case, there will be a symmetry in the observations $y(x(t))$, because two different sets p, p' can result in the same value q. When searching for a diffeomorphism between $P \times W$ and Z, the symmetry would invalidate one of the assumptions for Takens' theorem (see figure 2.14 for such an example with the Lorenz attractor). However, in precisely this case, we can replace the dependence of h

and $\hat{y}$ on p with a dependence on the effective parameters $Q(p)$—since both h and $\hat{y}$ are never actually computed, and are just assumed to exist, this does not change any of the algorithms involved. Also, as we are ultimately only interested in the generation of the observations $y(x(t))$, it does not matter if they were created from $P \times W$ or $Q(P) \times W$. If we replace P by $Q(P)$, the symmetries vanish, and Takens' theorem holds. □

Corollary 1. *Consider the diffeomorphism d from Lemma 1, and the parametrization*

$$\Psi(t) = \Psi(y(x(t)), y(x(t-1)), \ldots, y(x(t - 2(\dim P \times \dim W)))$$

of the observation manifold embedded in the delay space from Lemma 1, by a diffusion map Ψ. Then, there exists a map $g : Z \to Z$, such that

$$z(t+1) = g(z(t)), \tag{3.24}$$

where $z(t) = \Psi(y(x(t))$ consists of (truncated) diffusion coordinates.

Proof. The construction of the space Z involves a nonlinear change of coordinates $\Psi : Y^{(2(\dim W + \dim P)+1)} \to Z$, so that we can identify points in $Y^{(2(\dim W + \dim P)+1)}$ with points in Z. This allows to define the function $g : Z \to Z$ through

$$g = \Psi \circ d \circ h \circ d^{-1} \circ \Psi^{-1}. \tag{3.25}$$

The inverse of Ψ exists on its image in Z, and maps into the subspace of $Y^{(2(\dim W + \dim P)+1)}$ of time-delayed observations. The function $h : Q(P) \times W \to W$, $h(q, w(t)) = w(t+1)$ exists by assumption of the hidden system. □

Lemma 2. *There exists an observation function $\tilde{y} : Z \to Y$, such that for fixed $t \in T$, $p \in P$,*

$$\tilde{y}(g^t \circ z_0(p)) = y(f^t \circ x_0(p)). \tag{3.26}$$

Proof. Identify $z(t) = g^t(z_0(p))$ and $x(t) = f^t(x_0(p))$. Ψ is a parametrization of the delay manifold in the space $Y^{(2(\dim W + \dim P)+1)}$, such that

$$z(t) = \Psi\left(y(x(t)), y(x(t-1)), \ldots, y(x(t - 2(\dim P \times \dim W)))\right).$$

Hence, we can define $\tilde{y}$ through

$$\tilde{y}(z(t)) = [\Psi^{-1}(z(t))]_1 = y(x(t)),$$

where Ψ^{-1} is the inverse of the parametrization, which is defined on its image. □

We can now proof theorem 2 using Lemmas 1 and 2 together with Corollary 1.

Proof. From the original system (equations 3.6–3.8), we have the sets of parameters and observed values over time, $\{(p, y(x(t))) | t \in T, p \in P\}$. These sets can be used to form a manifold of delay vectors embedded in a space $Y^{2(\dim W + \dim P)+1}$, such that

$$[y(x(t)), y(x(t-1)), \ldots, y(x(t - 2(\dim W + \dim P))] \in Y^{(2(\dim W + \dim P)+1)}.$$

Lemma 1 ensures existence of a diffeomorphism between this space and the product space $Q(P) \times W$. Corollary 1 states that the map g exists and correctly predicts future states in the parametrization of the observation manifold in $Y^{(2(\dim W + \dim P)+1)}$, which is given by a nonlinear change of coordinates Ψ. Lemma 2 then ensures existence of an observation function $\tilde{y}$, which can reconstruct the observation values $\hat{y}(w(t))$ of the hidden system given a point $z(t)$. By assumption, the observed values of the hidden system are the same as the observed values of the original system. $\square$

Theorem 2 is the basis for all further arguments regarding the surrogate model, as it assures that given exact versions of all functions involved, the construction and simulation processes result in observations of the original system. The case where the functions $\tilde{y}$, g, and z_0 are only given as numerical approximations is discussed next.

3.2.3 Numerical errors

The surrogate model yields the observations of the original in the limit of an infinite amount of data. In applications, this limit is not reached, and interpolation or approximation methods must be employed. Here, we analyze how the errors of these methods influence the approximation error of the observations. The numerical experiment described in section 3.2.1 at the beginning of this chapter demonstrates how the total approximation error of the observations is affected by the individual errors of the initial map z_0 and the surrogate dynamic g. We now formulate and proof Theorem 3, which exactly states how the different error sources of the initial map z_0, the dynamic g, and the observation function $\tilde{y}$ contribute to the total error. We ignore measurement errors from observations. The problem of validation is also not considered, which would measure the deviation of the model predictions from reality. The matrix and eigensystem computations necessary for the diffusion map can be performed very accurately and thus their errors are neglected, too. Closed observables are based on a truncated diffusion map, where the truncation is such that all information necessary to close the system is kept. Hence, we can also ignore the truncation error in this case. Three main sources of error remain:

1. Mapping from a given input to the initial state of the closed observables,

2. Iterating the dynamic, and

3. Observing the current state.

We denote interpolated versions of the different functions of the surrogate model with a capital I as a subscript, and the error between the interpolant and the actual function by $E_{z_0,g,\tilde{y}}$, respectively, so that

1. the initial map z_0 is interpolated by $z_{0,I}$ with maximum error $E_{z_0} := \max_{p \in P}(z_{0,I}(p) - z_0(p)) \in Z$,

2. the dynamics g is interpolated by g_I with maximum error $E_g := \max_{z \in Z}(g_I(z) - g(z)) \in Z$, and

3. the observer $\tilde{y}$ is interpolated by $\tilde{y}_I$ with maximum error $E_{\tilde{y}} := \max_{z\in Z}(\tilde{y}_I(z) - \tilde{y}(z)) \in Y$.

All interpolants are constructed from discrete observations and diffusion map coordinates. The numerical model with these interpolants approximates the observed values of the original model. The approximation error only depends on the interpolation errors E_{z_0}, E_g and $E_{\tilde{y}}$, as stated by Theorem 3 and proven below. We proof a generic distance between two nested functions in Lemma 3 using a Taylor series expansion. This is the basis for the approximation theorem (theorem 3) that relates the total error to the individual errors.

Lemma 3. *Consider two vector spaces P and Z. Let $g \in C^2(Z, Z)$ and $z_0 \in C^2(P, Z)$. Consider the interpolants of these functions, $g_I \in C^2(Z, Z)$ and $z_{0,I} \in C^2(P, Z)$, with interpolation errors $E_g(z) = g(z)-g_I(z)$ and $E_{z_0}(p) = z_0(p)-z_{0,I}(p)$, where the maximum interpolation errors E_g and E_{z_0} are defined above. Also, let g be Lipschitz-continuous such that for $M \in (0, 1)$, $\sup_z \|Dg(z)\| < M$. Then, for fixed $p \in P$, $t \in \mathbb{N}$,*

$$\|g^t(z_0(p)) - g_I^t(z_{0,I}(p))\| \leq C_1 \left(\frac{1 - M^{t+1}}{1 - M}\right) \|E_g\| + C_2(M^t\|E_{z_0}\|) \tag{3.27}$$

where the constants C_1, C_2 are positive and do not depend on p and t.

Proof. The Taylor-decomposition of a vector-valued C^2 function f at a point $x + \epsilon$ with small $\|\epsilon\|$ is given by

$$f(x + \epsilon) = f(x) + Df(x) \cdot \epsilon + \mathcal{O}(\|\epsilon\|^2). \tag{3.28}$$

This approximation is the basis for the following arguments. For any $z \in Z$ and $t \geq 1$,

$$\begin{aligned}
\|g_I^t(z)\| &= \|g_I^{t-1} \circ [g(z) + E_g(z)]\| \\
&= \|g_I^{t-2} \circ [g^2(z) + Dg(g(z)) \cdot E_g(z) + \mathcal{O}(\|E_g(z)\|^2) + E_g(z)]\| \\
&= \|g_I^{t-2} \circ [g^2(z) + \mathcal{O}((M + 1)\|E_g(z)\|) + \mathcal{O}(\|E_g(z)\|^2)]\| \\
&= \|g_I^{t-3} \circ [g^3(z) + \mathcal{O}((M(M + 1) + 1)\|E_g(z)\|) + \mathcal{O}(\|E_g(z)\|^2)]\| \\
&= \dots \\
&= \|g^t(z) + \mathcal{O}(\sum_{i=0}^{t} M^i\|E_g(z)\|) + \mathcal{O}(\|E_g(z)\|^2)\|.
\end{aligned}$$

With $M \in (0, 1)$, we thus have

$$\|g_I^t(z)\| \leq \|g^t(z) + \mathcal{O}\left(\frac{1 - M^{t+1}}{1 - M}\|E_g(z)\|\right)\| \tag{3.29}$$

Similarly, for any $p \in P$,

$$\begin{aligned}
\|g^t(z_{0,I}(p))\| &= \|[g^{t-1} \circ g](z_0(p) + E_{z_0}(p))\| \\
&= \|g^{t-1} \circ [g(z_0(p)) + Dg(z_0(p)) \cdot E_{z_0}(p) + \mathcal{O}(\|E_{z_0}(p)\|^2)]\| \\
&= \|g^{t-2} \circ [g^2(z_0(p)) + Dg(z_0(p)) \cdot \mathcal{O}(M\|E_{z_0}(p)\|) + \mathcal{O}(\|E_{z_0}(p)\|^2)]\| \\
&= \dots \\
&\leq \|g^t(z_0(p))\| + \mathcal{O}(M^t\|E_{z_0}\|).
\end{aligned}$$

Note that for $M \in (0,1)$, the right side converges to $\|g^t(z_0(p))\|$ if $t \to \infty$. Combining the two approximations, we conclude that for any $p \in P$,

$$\begin{aligned} \|g_I^t(z_{0,I}(p))\| &\leq \|g^t(z_0(p)) + \mathcal{O}\left(\frac{1-M^{t+1}}{1-M}\|E_g\|\right) + \mathcal{O}(M^t\|E_{z_0}\|)\| \\ &= \|g^t(z_0(p))\| + C_1\left(\frac{1-M^{t+1}}{1-M}\right)\|E_g\| + C_2M^t\|E_\phi\| \end{aligned}$$

for constants $C_1, C_2 > 0$ independent of p and t. □

Lemma 3 can be used together with the general approximation theorem (2) proved in the last section to formulate and prove the main theorem of this section.

Theorem 3. ***Approximation error of the surrogate model*** *Consider the state space X of the model in equations 3.6–3.8, the parameter space P, the initial map $x_0 : P \to X$, the dynamic $f : P \times X \to X$, and the observer $y : P \times X \to Y$. Let the objects $z_0 \in C^2(P,Z)$ (initial map into closed observable space Z), $g \in C^2(Z,Z)$ (dynamic) and $\tilde{y} \in C^2(Z,Y)$ (observer) be approximated by the functions $z_{0,I} \in C^2(P,Z)$, $g_I \in C^2(Z,Z)$ and $\tilde{y}_I \in C^2(Z,Y)$. The errors E_{z_0}, E_g and $E_{\tilde{y}}$ are defined above, as points in the images of z_0, g, and $\tilde{y}$. The norms of the errors are assumed to be much smaller than one. The dynamic g must be Lipschitz-continuous with a constant $M \in (0,1)$ such that $\sup_\phi \|Dg(\phi)\| < M$. Then, for any given initial state $p \in P$ and any fixed number of iterations $t \in \mathbb{N}$,*

$$\|y\left(f^t(x_0(p))\right) - \tilde{y}_I\left(g_I^t(z_{0,I}(p))\right)\| \leq C_1\left(\frac{1-M^{t+1}}{1-M}\right)\|E_g\| + C_2M^t\|E_{z_0}\| + C_3\|E_{\tilde{y}}\|. \tag{3.30}$$

The constants C_1, C_2 and C_3 are positive and independent of t and p.

Proof. **Theorem 3**

1. Given the exact versions of the numerical objects $z_{0,I}$, g_I and $\tilde{y}_I$, the output is reproduced exactly. This is proved through theorem 2 in the previous section.

2. The interpolants of the numerical objects have small errors, so that multiplications of the errors vanish.

3. Taylor-decomposition of the interpolants (Lemma 3) concludes the proof.

□

The theorem states how the total error of the surrogate model compared to the original model depends on the individual errors of the interpolants. Reviewing the numerical experiment at the beginning, the qualitatively similar results for a change in the number of samples N_x in x_1 can be now explained: we used the same interpolation method for all three functions, and only changed the approximation of g between forward and centered differences. This affects only E_g, but not E_{z_0}, hence a qualitative change is only visible looking at a changing number of samples in time.

3.2.4 Different interpolation schemes

Theorem 3 stated in the previous section relates the errors of the three interpolation or approximation functions to the total error made by the surrogate. In the statement, we do not assume any particular form of the interpolants, but that their errors are small, so that powers of the errors are negligible. The general form of theorem 3 allows to replace the error norm $\|E_g\|$ by the specific error of the method.

For low dimensional, non-periodic functions, a polynomial basis in the form of a Taylor-decomposition, Hermite-spline interpolation, or Chebyshev polynomials can be a first choice of an interpolation or approximation method. All of them have analytic forms for the interpolation error. For periodic functions, the same holds for a Fourier basis.

More advanced methods include sparse grids (Bungartz and Griebel, 2004), (Extended) Dynamic Mode Decomposition (Schmid, 2010; Williams, Kevrekidis, and Rowley, 2015), the Empirical Interpolation Method (Barrault et al., 2004; Peherstorfer et al., 2014), Discrete Empirical Interpolation (Chaturantabut and Sorensen, 2010) and out-of-sample extension methods such as the Nyström extension (Bengio et al., 2004). A more general approximator is given through neural networks (Lapedes and Farber, 1989). All of these methods have been used for model order reduction, and can directly be applied as an interpolation scheme for the dynamics g of the surrogate model. A precise error analysis for these methods is often difficult, but all of them provide some form of error estimation. In the applications discussed in section 4, we mostly use linear or spline based interpolation and approximation methods, because the state space dimension of the surrogate model never exceeds three. Table 4.1 in the chapter 4 lists all of the interpolation methods used for the applications.

3.2.5 Storage

The interpolants used by the numerical model require storage for data points or parameters. Theorem 4 formulates a condition for the case that the numerical model needs much fewer data points—in fact, a lower dimensional space—for the same accuracy compared to the storage of all output. This is one of the main motivations to construct the new model, instead of simply storing output and looking up the values when needed. To fulfill the condition of Theorem 4, the input parameters of the original system must have at least one degenerate dimension. This means that the dynamic f of the original system depends on a set of parameters with lower dimension than the dimension of the parameter space used. See the previous section, section 3.2.2, for an introduction to degenerate parameter spaces.

If it is not clear whether the construction of the surrogate model will save storage or not, it is still informative to construct the model and check the condition of Theorem 4. If the condition is fulfilled, there is a degeneracy in the parameter space. The degeneracy might not be apparent beforehand, as changing any of the parameters can still result in a different trajectory. The limit cycle system (equation 3.1) is another example where the reduction of needed storage capacity is significant. Naively storing all observations over time for all initial states would take a three-dimensional hyper-surface: two dimensions for the parameters, and one for time. The reduced model with closed observables only needs two 1D-lines for dynamic and observer (figure 3.10, center and right), and one 2D-surface for the translation from initial state to the new coordinate (figure 3.10, left). Even in this low-dimensional case, the naive storage of all observations over time for all initial states would take up all memory of a supercomputer (Tianhe-2, state 2015). If the same naive sampling is applied together with closed observables, the data fits in the memory of a smartphone (state 2015, figure 3.16).

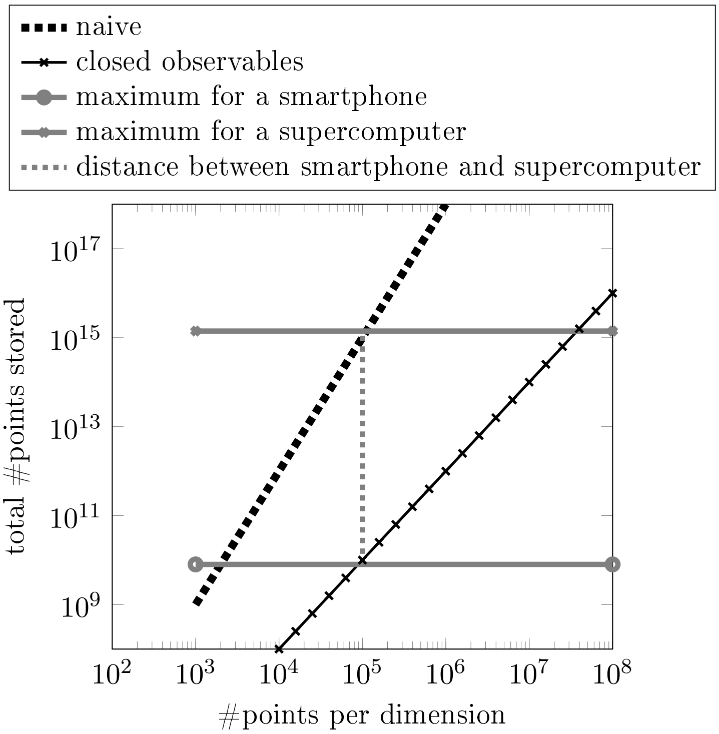

Figure 3.16: Comparison of the capacity needed to store the generated data of the spiral example, compared to the storage of the surrogate model.

In this section, we formalize the storage requirements of an interpolant, and show the reduction of storage through the proof of Theorem 4 below. Consider a function $f \in C^k([0,1]^p, \mathbb{R}^m)$ with $k \in \mathbb{N}_0$, $p, m \in \mathbb{N}$. Let $\epsilon > 0$ and consider an interpolant f_ϵ of f, such that

$$\|f_\epsilon - f\|_\infty < \epsilon. \tag{3.31}$$

If f is a black box (also sometimes called oracle) and can hence only be queried at distinct points $x_i \in [0,1]^p$ to yield the values $f(x_i)$, the interpolant is constructed by sampling the space $[0,1]^p$ with a sampling method S. This method S needs $S(\epsilon, p)$ sampling points in the construction of the interpolant f_ϵ. For the full-grid sampling method, $S(\epsilon, p)$ increases exponentially in p, since if N different samples of one coordinate axis are considered, N^p points must be stored for the p-dimensional space. If the function f has regularity $k > 0$, the error between f_ϵ and f decreases with the order $\mathcal{O}(N^{-k/p})$. More advanced sampling methods such as sparse grids exist (Bungartz and Griebel, 2004), where the number of points for a comparable accuracy increases with $\mathcal{O}(N(\log N)^{p-1})$, mitigating the curse of dimensionality. For all sampling methods, decreasing p without changing the smoothness properties of the function is a good way to reduce the number of points that need to be stored for f_ϵ in order to achieve an accuracy smaller than ϵ.

We approximate the output of a discrete dynamical system (equations 3.6–3.8), to find an interpolant I_ϵ for the function

$$I : \mathbb{N} \times P \to Y,\ I(t, x(0)) = y(f^t(x(0))) \approx I_\epsilon(t, x(0)). \tag{3.32}$$

If P is p-dimensional, the function I has $p+1$ parameters. Hence, an interpolant I_ϵ will need $S(\epsilon, p+1)$ points for an approximation accuracy of at least ϵ. With closed observables, we only need $S(\epsilon, p)$ points for the same accuracy, which will be shown through Theorem 4. The reduction from $p+1$ to p dimension might not seem much, but the dimensionality reduction is for the time variable: as we construct a model, only its intrinsic state space must be sampled and stored. The time variable might need significantly more sampling points than the intrinsic state space, and in that case, the reduction is significantly larger compared to a reduction of one coordinate in an equally-spaced grid.

Theorem 4. ***Storage reduction*** *Let $P \subset \mathbb{R}^{\dim P}$ be the parameter space, and $z_0 : P \to z_0(P)$ the initial map of the surrogate model, mapping P into the intrinsic state space. Define $n_0 := \dim(P)$, and $d := \dim(z_0(P))$. Given a desired interpolation accuracy $\epsilon > 0$ and a sampling method S as defined above, the interpolants ϕ_I, g_I and $\tilde{y}_I$ need at most $\mathcal{O}(S(\epsilon, n_0))$ points if the following condition is true:*

$$d < n_0 + 1. \tag{3.33}$$

In contrast to that, storing the output of the original model needs $\mathcal{O}(S(\epsilon, n_0 + 1))$ points, where the additional dimension is approximating the time variable.

Proof. Define $m := \dim(y(X))$.

1. For the n_0-dimensional input, we need exactly d surfaces with dimension n_0 each to store the mapping from input to the new variables.

2. We also need d surfaces with dimension d each to store the dynamic g on the space $z_0(P)$.

3. Finally, we need m surfaces with dimension d each to store the observer on the new space.

4. Since storing a finite number of surfaces does not add another dimension, the maximum dimensionality we need to store the new model is $\max(d, n_0)$.

5. To store the observed values over time, we need m surfaces (n_0-dimensional) for each iteration step n, so in total we need $n_0 + 1$ dimensions.

6. The exact number of hyper-surfaces including dimension for the new model is given by $d \cdot \underline{d} + m \cdot \underline{d} + d \cdot \underline{n_0}$, where the underlined values stand for*dimensions* and the multiplication between a number and a dimension is noted by $(\cdot)$. Note that the dimension is not distributive, that is $a \cdot (\underline{b} + \underline{c}) \neq a \cdot \underline{b} + a \cdot \underline{c}$. The dimensions are additive, so $\underline{a} + \underline{b} = \underline{a + b}$. This allows for a reformulation into

$$\text{storage(new model)} = (d + m) \cdot \underline{d} + d \cdot \underline{n_0} \tag{3.34}$$

7. The exact number of hyper-surfaces including dimension for the storage of all observations is given by $m \cdot \underline{n_0 + 1}$. Reformulation yields

$$\text{storage(full output)} = m \cdot \underline{n_0 + 1} \tag{3.35}$$

8. The theorem follows from comparing and reducing the storage formulations in items 6 and 7.

□

The storage efficiency (Theorem 4) is relevant since a lot of output needs to be stored for the numerical model to be accurate. The theorem states the conditions when it is highly advantageous to construct closed observables instead of simply storing the output. The difference can be as large as the difference between the memory of a smartphone and a supercomputer (see figure 3.16).

3.2.6 Infinite-dimensional observables and parameter spaces

If the observed values are functions, as is the case for PDEs, the observed space must be decomposed in finitely many basis functions. Then, the constructed observer function must yield the coefficients of the basis functions. See Dietrich, Köster, and Bungartz (2016) for an example with a PDE with two parameters, as well as Dietrich, Albrecht, and Köster (2016) and the application in section 4.1.1 for an example where the observable is the density of particles moving through a bottleneck.

Formally, the observation function y maps from M to Y. In the case treated here, Y is a space of functions. There are numerous methods to treat this case. Here, we restrict Y further, to spaces with a finite-dimensional basis. Note that for a given time t, functions in Y_t always have a finite-dimensional basis if the parameter space P also has a finite-dimensional basis. This is due to the construction of the surrogate: we always start with the same initial condition, and only vary the parameters. Since for a given parameter set, $y(x(t))$ will yield one function for any t, we have $\dim Y_t \leq \dim P$. Depending on the dynamics of the microscopic system, the part of Y observed at the beginning might not be the same after a certain time t. Since the dependence on time is yet another dimension, to fully describe the possible observable states, we have to find a basis for the family Y_t of all observable functions for all times $t \geq 0$.

A common solution for a function of space is to sample the space into grid cells of equal size, and then use indicator functions on each grid cell to form the basis of the observed function space. In most applications, this choice of basis is redundant, and the number of basis functions can be reduced by several orders of magnitude while still providing an accurate representation of the function space Y_t. Numerically, this is possible through the *method of snapshots* described in section 2.4.2. In this case, the coefficients in the finite basis evolve in time, and serve as observations of the system.

Examples include ocean bathymetry in the shallow water equations, structure of the earth in earthquake simulations, floor geometry in porous media simulations, the geometry of buildings and events in crowd dynamic simulations, or the initial distribution of the crowd. In all these examples, the input parameter is a function and hence an element from an infinite-dimensional space. In this case, the function space has to be decomposed into a finite number of basis functions. Their coefficients are the new parameters for the surrogate model.

3.2.7 Stochastic effects

Theorems 2 and 3 on convergence and numerical errors assume that the hidden macroscopic system is deterministic, and hence does not include stochastic effects in the dynamic and the observation function. In many applications and theoretical models, the assumption of fully deterministic behavior is violated. Generally, stochastic effects model behavior on spatial or temporal scales unknown to or unresolved by the observer. It is still possible to construct dynamical surrogate models if stochastic effects are present. If the observed values are distributed stochastically, the surrogate model can include stochastic terms. Alternatively, the surrogate can model modes of the probability distribution, such as expectation and variance, which makes it a deterministic model again. Methods to estimate stochastic models from data are mostly found in statistics and related disciplines. Auto-regression moving average models are a broad class for processes with a linear dependence on a few past values (ARMA, see Brockwell and Davis (1991)). Chaotic systems admit a different type of stochasticity. Even for deterministic, chaotic systems, any error in the initial condition increases exponentially over time, so that solutions differ arbitrarily after a finite time horizon, as if they included a stochastic component. Still, there are methods to find deterministic models for the most important directions of the dynamics, such as Koopman eigenfunctions (Williams, Rowley, and Kevrekidis, 2014; Williams, Kevrekidis, and Rowley, 2015; Williams et al., 2015; Giannakis, Slawinska, and Zhao, 2015), and kernel methods (Giannakis, 2015). Between the microscopic and the macroscopic models, there are four possibilities for stochastic and deterministic behavior, summarized in Table 3.2.

	micro-stochastic	**micro-deterministic**
macro-stochastic	(A) S-S	(B) D-S
macro-deterministic	(C) S-D	(D) D-D

Table 3.2: Four possibilities for stochasticity in upscaling microscopic models to macroscopic models.

(A) Stochastic-Stochastic

There are not many examples where processes on both the microscopic and the macroscopic scale are modeled with stochastic effects. An exception are models on intermediate scales, that try to summarize microscopic motion into larger portions of space. One example is the SPOT model (Bazant, 2006), with stochastic motion of granular particles on the microscopic scale, and also stochastic motion of a larger group of particles, called *spots*, on a larger scale. The spots are advanced in space and time through a Monte-Carlo simulation, whereas a discrete element method (DEM) is used to advance the individual particles. While the particle model and also the related DEM are not intrinsically stochastic, the individual particle properties are highly sensitive to a change in initial conditions, and any observation of particles over a longer time period must include a form of averaging. This also holds for the macroscopic scale, as the spot model simulations are stochastic.

(B) Deterministic-Stochastic

A possibility to scale up a microscopic model is to summarize individual behavior into a stochastic term on the macroscopic scale. A deterministic, microscopic model can thus be modeled macroscopically through a deterministic drift term and a stochastic diffusion term. Examples for this upscaling technique are stochastic cellular automata for the evolution of pedestrians or particles. Even chaotic systems can be predicted on a larger scale, in a statistical sense. From Araujo, Galatolo, and Pacifico (2014, p.2):

> Even if the pointwise description or forecasting of the [chaotic] system is forbidden by the initial condition sensitivity, the statistical behavior is often relatively simple and its properties are often (with a certain effort) predictable.

(C) Stochastic-Deterministic

In many cases, the microscopic process is modeled with a stochastic component, or is highly sensitive to initial conditions and not predictable in a point-wise sense. Then, a deterministic macroscopic model for the same process that is no longer sensitive to initial conditions can help to predict observations for longer time periods, and often captures deterministic behavior of averaged quantities. The classical example are particles driven by Brownian motion, where evolution of the probability distribution is considered on the macroscopic scale. In fact, this example is the principle idea of diffusion maps, where a random walk between the points is modeled by a Markov matrix, and the eigenfunctions of the diffusion map observe this process in the averaged, continuous limit (see section 2.2.2 and (Coifman and Lafon, 2006)).

(D) Deterministic-Deterministic

If both processes on the microscopic and macroscopic scale are deterministic, equation-driven upscaling often assumes a large gap between the scales and effectively ignores behavior on the temporally faster or spatially smaller scales. This approach is different from the introduction of a stochastic term (see part B above), as it cannot capture the dependence of initial conditions on the macroscopic scale. In contrast, the assumption is that the effects of the ignored parts of the microscopic process are unimportant for the evolution on the macroscopic scale.

3.2.8 Summary

In this section, we analyzed the construction and simulation process of the surrogate model, and discussed specific challenges occurring with parameter and observable spaces of high dimension, stochastic effects, and boundary problems.

We proved that the construction process for the data-driven surrogate model produces a system able to reconstruct the original observations in the limit of large data. For limited data, we show how the absolute error between the output of the surrogate and the original model is influenced by the errors of the three interpolants for the initial map, the dynamic map, and the observation function. When a large parameter space is sampled, or a very complex surrogate model is constructed, the interpolants for the surrogate might need a lot of storage capacity. Instead of directly storing the observations of the original model, it can be beneficial to compute the surrogate model and store the interpolants. We proved the process is storage efficient in case the number of intrinsic variables of the surrogate is less than or equal to the number of input parameters varied in the construction.

Following the three proofs, we discussed three specific challenges for the construction of data-driven surrogate models. The first and second challenges we discussed concern the approximation of high-dimensional spaces, separately for parameters and observations. For the surrogate model, high-dimensional parameter spaces pose a greater problem than high-dimensional observation spaces, because the initial map needs as many dimensions as there are dimensions of parameters. Similarly, a high intrinsic dimension of the state space is also challenge. If only the extrinsic dimension of parameter or state space is high, methods for dimension reduction can be used in addition to the usual construction process of the surrogate model. The third challenge concerned stochastic processes and their effect on the surrogate model. Stochasticity can enter the system by definition of the microscopic model, or by a finite number of microscopic simulations averaged to an empirical expectation.

Chapter 4

Applications: from crowd motion to granular flow

The dynamic, data-driven surrogate models introduced in the last chapter has a broad range of applications. In the setting of multiscale dynamics, and specifically scale transitions, the surrogate models are especially well suited to address issues in granular flow media. We discuss the movement of crowds in bottleneck scenarios and queues, the movement of cars on a highway, granular particles in a silo, as well as a combination of surrogates with other methods of system analysis, particularly uncertainty quantification. The subsections are all structured in the same way:

General description of the application We introduce the scenario, the specific problem, parameter space, and dynamic under study.

State-of-the-art approach without dynamic surrogate models We briefly outline how the applications problem could be solved with conventional methods, focusing on strengths and potential research gaps which can be closed using the dynamic surrogate model developed here.

Alternative approach with the dynamic surrogate model Here, we show how to address the research gaps from the state of the art using dynamic surrogate models. This part contains descriptions of the construction process, and visualizations of the surrogate model.

Conclusions We conclude each application with a summary and potential future directions of research.

The complexity of the applications in terms of equation- and data-driven multiscale methods is shown in figure 4.1. We refer to this figure in the description of each application below. It is easy to capture the dynamics of the limit cycle example shown before (section 3.1.1), both in a data-driven and equation-driven way, because the solution is explicitly known. On the top-right of the figure, with highest complexity levels, is uncertainty quantification in a train station (section 4.1.3). A formulaic description of the macroscopic uncertainties based on the microscopic system is challenging, as the agent-based models for passengers are complex and often only given as an implementation in software. Data-driven uncertainty quantification has to deal with the curse of dimensionality, and stochastic effects if the macroscopic quantity is computed as an average over a finite number of different scenario runs. The same holds for applications in granular flow, where we will demonstrate how the surrogate model can be used to extract macroscopic observables automatically. Table 4.1 contains the parameter values used in all applications. It provides a reference for similar applications.

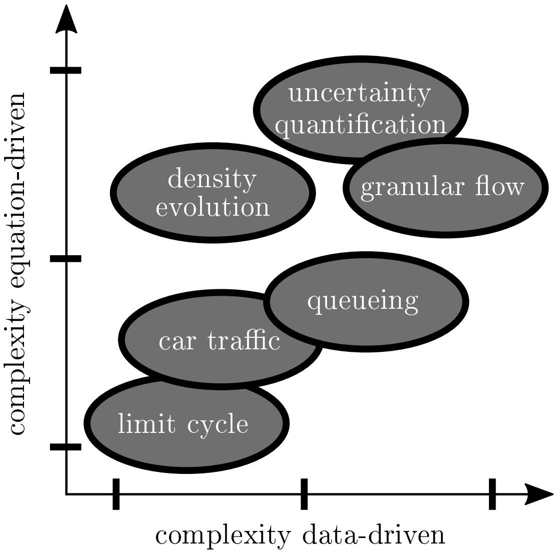

Figure 4.1: Levels of complexity for the applications. The data-driven complexity of an example shows how much it is affected by the curse of dimension, and how complex a numerical simulation of the problem is. The equation-driven complexity signalizes if there is established theory to deal with the problem (low complexity), or not (high complexity). Complexity levels are low if the topic is a well-known result, and are higher if it is still under active research.

Table 4.1: Parameters for all example applications. The method of simulation always is Dormand-Prince 45 (`scipy.integrate.ode`, `dopri5`).

Example	T	ϵ	κ	d	**Interpolation/approximation**
Spiral	60	1×10^3	2×10^{-7}	1	scipy.interpolate.UnivariateSpline
Bottleneck	31	1×10^5	5×10^{-5}	2	scipy.interpolate.-SmoothBivariateSpline
Density evolution	20	1×10^5	1×10^{-3}	3	scipy.interpolate.Rbf
Queueing	10	1×10^9	1×10^{-3}	2	piecewise linear
Uncertainty quantification	80	1×10^7	1×10^{-3}	3	piecewise linear
Car traffic	5	1×10^3	1×10^{-4}	2	piecewise linear
Granular flow	19	1×10^6	1×10^{-3}	1	polynomial

4.1 Crowd dynamics

Two major challenges with macroscopic models for crowd dynamic will be addressed by the two example applications in this section.

The first challenge is the rapid development of microscopic models, which renders any careful analytic homogenization of a given microscopic model invalid after a few months, because the description of the microscopic model has changed too drastically. In this situation, the dynamic surrogate model developed here is a helpful tool. Constructing surrogates for different models, or with experimental data, allows to compare their dynamical properties on a macroscopic scale. We demonstrate this by constructing a macroscopic model for the evolution of crowd density, given only data from a scenario where a crowd evacuates through a bottleneck and is simulated with a microscopic model. As the observed values of density are a function of state, we employ model order reduction for infinite-dimensional observer functions (see section 2.4.2 and 3.2.6).

The second challenge is the analysis of a microscopic model through numerical simulations, given the typically high demand for computational resources. Unlike grains in granular flow, human beings are already complex on the individual level, not only on a macroscopic scale. Microscopic models that incorporate some of that complexity need a lot of computational resources. In physics, a similar challenge exists when trying to model particles on the quantum level. At the same time, the complex microscopic models often have more than two or three parameters, which itself demands a lot of computational power and storage when performing numerical parameter studies. We show how to meet this challenge with an example of queuing pedestrians at the entrance of a music festival. We analyze the scenario for bifurcations by constructing a surrogate model, and show the relations to queuing theory.

4.1.1 Evolution of crowd density - a surrogate for a reduced order model

As described in section 2.4.2, model order reduction reduces the number of degrees of freedom of a system while keeping its accuracy at an acceptable level. In computational fluid dynamics, this technique is widely adopted. Here, we show how surrogate models reduce the degrees of freedom even further, including system parameters, when relevant observables of the system are on a macroscopic scale. We demonstrate the results in a crowd dynamics context, a bottleneck scenario. The equation-driven complexity of this application stems mainly from the formulation of the microscopic model, which is still an unsolved problem. Also, the quantity of interest, the density, is a continuous function of time and space, and must be discretized properly. Here, we use a reduced order model formulation with the method of snapshots. The numerical issues associated with the needed singular value decomposition are already very well resolved in standard software, and hence the data-driven complexity of the example is rated relatively low.

General description of the application Bottlenecks are one of the most important scenarios in the analysis of crowd dynamics, because they reduce the crowd flow. Here, we use a prototypical bottleneck, where 180 pedestrians move from their starting position to an exit (see figure 4.2). This scenario was explored experimentally by many different research groups, in this specific setup by Seyfried et al. (2010) and Liddle et al. (2011). The study presented here was published by Dietrich, Albrecht, and Köster (2016), and includes the construction of a surrogate model for two microscopic models for pedestrian dynamics: the gradient navigation model (Dietrich and Köster, 2014) and the optimal steps model (originally from Seitz and Köster (2012), we use the version detailed by Sivers and Köster (2015)). In addition to the two models, data from a controlled experiment was also used (Seyfried et al., 2010; Liddle et al., 2011). Here, we only present the results for data generated with the gradient navigation model. The surrogate models constructed for the optimal steps model has comparable dynamics, and also needs the same number of intrinsic variables (Dietrich, Albrecht, and Köster, 2016). The experimental data contains one run per parameter set instead of the 100 runs we perform with the simulation models. Also, the length parameter is sampled with only three values, at a width of $1.20m$, whereas the width parameter is sampled with nine values, keeping the length at $4m$. This lack of data makes it difficult to reduce noise, and hence we ignore the length dimension and only use the width dimension as a parameter of the surrogate for the experimental data. The reduced parameter space leads to a reduction of the intrinsic dimension of the surrogate model for the experimental data, compared to the simulation models.

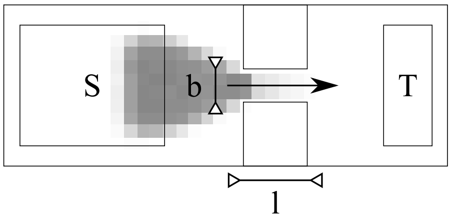

Figure 4.2: Scenario setup for a bottleneck, published by Dietrich, Albrecht, and Köster (2016), and mimicking the experiments by Liddle et al. (2011). 180 pedestrians start from region S at a density of about $2.6P/m^2$, and move through the bottleneck of width b and length l towards the region T. These two parameters are varied to create the surrogate. The crowd density at $t = 10s$ is shown sampled on a grid with cells of $1m \times 1m$. The distance from S to the bottleneck is $5m$, the distance from T to the bottleneck is $7m$.

We demonstrate how to

- Apply model order reduction using the method of snapshots (see section 2.4.2).
- Construct a surrogate for the time evolution of the reduced order model.
- Reconstruct density and velocity profiles of the crowd from the states of the surrogate model.

State-of-the-art approach without dynamic surrogate models A conventional approach in the field of crowd dynamics would be to use a macroscopic model for crowd density and velocity, and fit its parameters such that the evolution over time is close to the output of the microscopic model. As an example, Hughes (2001) derives macroscopic equations from assumptions similar to fluid flow, for highly congested areas such as the Hajj in Mecca. The model parameters are then calibrated to fit the flow around one large pillar in Mecca, and compared to empirical crowd densities. In a similar fashion, Helbing et al. (2006) formulate a macroscopic model for evacuation through a bottleneck. There are also equation-driven approaches available, using generalized microscopic models to form a macroscopic description of the process. Bellomo and Bellouquid (2015) describes such an equation-driven approach, where a mesoscopic, kinetic model describing the probability distribution of pedestrians is derived from a microscopic model, and then the macroscopic model is derived from the kinetic model through an asymptotic limit of the number of pedestrians. Cristiani, Piccoli, and Tosin (2014) combine microscopic and macroscopic models into a measure-theoretic framework with interacting scales. Lachapelle and Wolfram (2011) derive a macroscopic mean field game for two interacting crowds from a stochastic, microscopic model.

In fluid dynamics, there are already established models on the macroscopic scale, mostly in the form of partial differential equations. Computational fluid dynamics then constructs reduced order models for efficient solutions of the equations. The model order reduction approach would usually be followed by an approximation of the non-linear dynamics. This can be achieved through the discrete empirical interpolation method (DEIM, see (Chaturantabut and Sorensen, 2010)).

The macroscopic models, developed through methods from the state of the art, treat parameters and model variables differently. The variable is often a function of space, describing density and velocity of crowds. It is often not immediately clear that this variable holds enough information to advance in time, and closure assumptions are necessary.

Alternative approach with the dynamic surrogate model In case of the dynamic surrogate model developed here, choosing the correct variables for closed dynamics is delegated to the construction algorithm. In the first step, we use a microscopic simulator to generate trajectory data of the scenario (openVADERE, available at `www.vadere.org`). Similar to the experiment from Seyfried et al. (2010), we vary the width and length of the bottleneck. For each fixed set of parameters, we simulate 100 random, uniformly distributed starting position sets for the pedestrians, and then average the number of pedestrians in individual grid cells for each time step, to reduce stochastic effects on the macroscopic scale. The length and width are varied between $0.06m$ and $4.0m$, and $0.9m$ to $2.5m$, respectively (see table 4.2). To test the strength of the surrogate model approach concerning the reduction of storage space, we also introduce the height of the bottleneck as a parameter with three samples for each tuple (w, l). While the height parameter is artificial, it serves to show how the surrogate construction automatically reduces the dimension. The height parameter increases the number of simulations of the gradient navigation model to $3 * 1800 = 5400$, but it does not affect the microscopic simulation results. The constructed surrogate model does not have more than three intrinsic dimensions, two for width and height, and one for time. The lower number of variables reduces the storage needed for all observed data from 5400 down to 1800 trajectories, which is an automatic result of the construction of the surrogate model.

Table 4.2: Parameters width and length of the bottleneck used to generate observation data for the surrogate model. In total, we use $6 \cdot 3 \cdot 3 = 54$ different parameter sets.

width [m]	**length [m]**	**height [m]**
$0.9, 1.0, 1.4, 1.8, 2.0, 2.5$	$0.06, 2.0, 4.0$	$1, 2, 3$

Considering 54 different sets of the parameters width, length, and height, and 100 simulation runs per parameter set, we perform $54 \cdot 100 = 5400$ simulations in total. The desired speeds of individuals in the microscopic gradient navigation model is calibrated to fit the evacuation times found in the experiment, before running the simulations necessary for the surrogate model. This calibration is only important to assure comparable time lengths of trajectories between experimental data and the model, and has no qualitative effect on the surrogate model.

In the second step, the positions and speeds of individual pedestrians generated by the simulations are processed to density and velocity grids covering the scenario. The simulation data on a $40m \times 20m$ grid is averaged over 100 runs of the same scenario with randomly assigned initial positions on the starting region. We remove an initial transient to the jamming phase of $t_0 = 10s$, and sample the data at $\Delta t = 4s$.

Then, the density and velocity grids are arranged in a matrix X, such that the column number enumerates points over both spatial dimensions for $j = 1, 2, \dots, p$, and the row number points in time for $t = 1, 2, \dots, n$. A singular value decomposition of the matrix X is used to extract the principal components, which form the coordinates of a linear subspace of the data space (see section 2.2.1 for a more detailed explanation of principal component analysis). The singular value decomposition of the data matrix X is

$$X = U\Sigma V^T, \tag{4.1}$$

which is split up into score values $S \in \mathbb{R}^{n\times p}$ and principal components $P \in \mathbb{R}^{p\times p}$ through

$$\underbrace{\begin{pmatrix} -x_1- \\ -x_2- \\ \vdots \\ -x_n- \end{pmatrix}}_{X} = \underbrace{\begin{pmatrix} | & | & & | \\ S_1 & S_2 & \cdots & S_p \\ | & | & & | \end{pmatrix} \begin{pmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \ddots & \\ 0 & & & \sigma_p \end{pmatrix}}_{U\Sigma=S} \underbrace{\begin{pmatrix} -P_1- \\ -P_2- \\ \vdots \\ -P_p- \end{pmatrix}}_{V^T=P^T}. \tag{4.2}$$

The matrix Σ contains the singular values on its diagonal. Omitting singular values below a threshold reduces the intrinsic dimension of the problem, with fewer scores and principal components to consider. Here, we set the threshold to 0.5% of the maximum singular value, resulting in 30 remaining values. Figure 4.3 shows the first four principal components $P_1, \dots, P_4$, resulting from the methods of snapshots. To reconstruct the density and velocity grids, the components P_j need to be multiplied with the score values $S_j(t)$ and summed over j. The reduced dimension in this example is still too high to extract numerically the dynamic of the score values $S_j(t) = (U\Sigma)_{jt}$ of the principal components $P_j = V_j$ over time. We would need to fit a surface of dimension 30 for functions $f_j(S(t)) = S_j(t+1)$. Note that this problem can be solved, for example, by estimating the Koopman operator on observables of the system (see section 2.4.2). The dynamic surrogate is a lower-dimensional approach (step three) and also incorporates the parameter dependence directly, reducing the systems dimension even further.

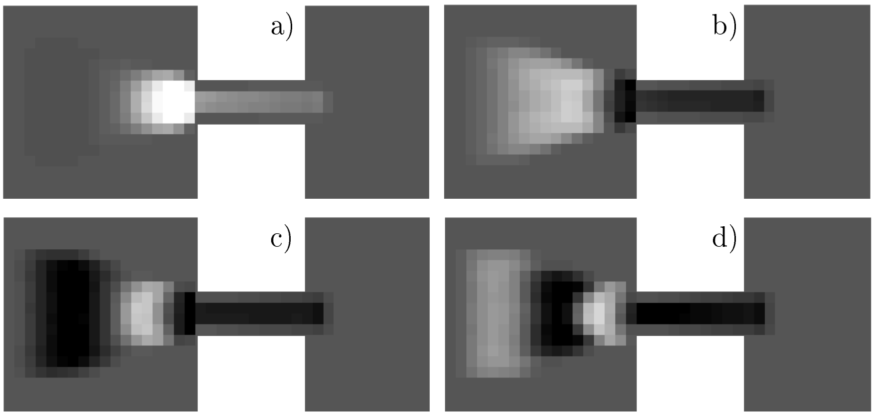

Figure 4.3: The parts of the first four principal components related to the density, resulting from the method of snapshots. The bottleneck is between the small, white boxes. The components are mapped from vector form to the two-dimensional domain of the bottleneck for visualization purposes only. High component values are white, low values are black. A linear combination of the principal components, weighted with the score value $S(t)$, can be used to reconstruct the density at t. (Figure adapted from original publication by Dietrich, Albrecht, and Köster (2016))

The third step uses the time-based score information contained in the matrix $S = U\Sigma$ to construct a surrogate model for the dynamic of the principal components over time, such that the j-th score value $S_j(t)$ can be recovered from the surrogate model by observing the closed observables at time t through the observation function $\tilde{y}$. The construction process is explained in detail in section 3.1.2. For this particular example, the construction uses time-lagged embedding of the values of S, that is, $(S(t), S(t-1), S(t-2), \dots)$, to reconstruct the state space of the underlying, macroscopic dynamical system. The ambient space of the state space can be very high if many time-lags are used, so a (in our case, non-linear) dimension reduction is necessary, where we employ diffusion maps (section 2.2.2). The new parametrization of the state space found through the diffusion map is low-dimensional, hence we can approximate the dynamic on it numerically. As the third step constructs a non-linear model, the number of variables can be much lower than the number of the principal components. The surrogate model constructed for the OSM and the GNM needs three variables, and two for the experimental data. The difference in the number of variables is because we neglected the length parameter in the experimental data—if enough data was available, and the length parameter has an influence on the dynamics, the number of variables would increase to three, too. Figure 4.4 visualizes the trajectories in the three-dimensional diffusion map space, for data generated by the gradient navigation model.

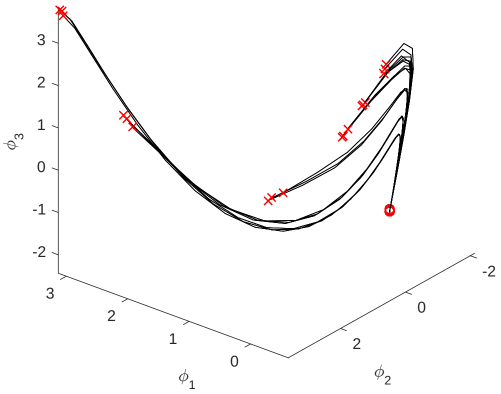

Figure 4.4: Trajectories of the closed observables in diffusion map space. The red crosses show initial values, the red circles the steady state of the system. All trajectories end up in the steady state, which is the point where all pedestrians have left the scenario and the density is zero everywhere.

There are three functions that need to be approximated for the surrogate model. The initial map ϕ_0 maps from given parameters to the initial data of the surrogate, and is visualized in figure 4.5. The dynamic g acts on the state variables $\phi \in R^m$, where $m = 3$ is the intrinsic dimension of the surrogate model. The dynamic is the right hand side of an ordinary differential equation, and is approximated through centered differences, such that $\dot{\phi} = g(\phi) = (\phi(t+\Delta t) - \phi(t-\Delta t))/(2\Delta t)$. It is shown in the plots of figure 4.6, where the color of the plots encodes the numeric value of the derivatives. The third function, called $\tilde{y}$, maps back from the surrogate variables to the score values, which are then used to reconstruct the density and velocity at the given time through multiplication with the principal components. Here, the thirty most important score values are reconstructed through $\tilde{y}$. The Python package `scipy.interpolate.Rbf` is used as interpolant for all three functions, with the smoothing parameter set to 1×10^{-3}, and a `multiquadric` radial basis.

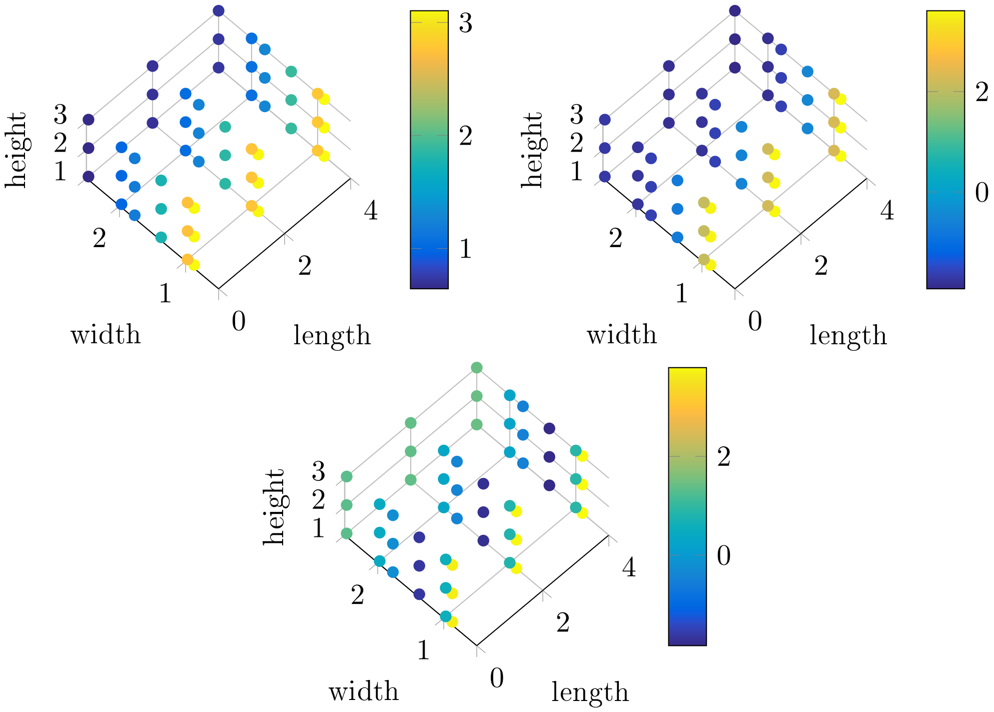

Figure 4.5: Initial value map $\phi_0(w, l, h) = \phi(0) = (\phi_1(0), \phi_2(0), \phi_3(0))$ from the three-dimensional parameter space (width, length, height) to the initial value of the closed observables $\phi(0)$ (color). The three plots show the initial values for the three coordinates of the closed observables.

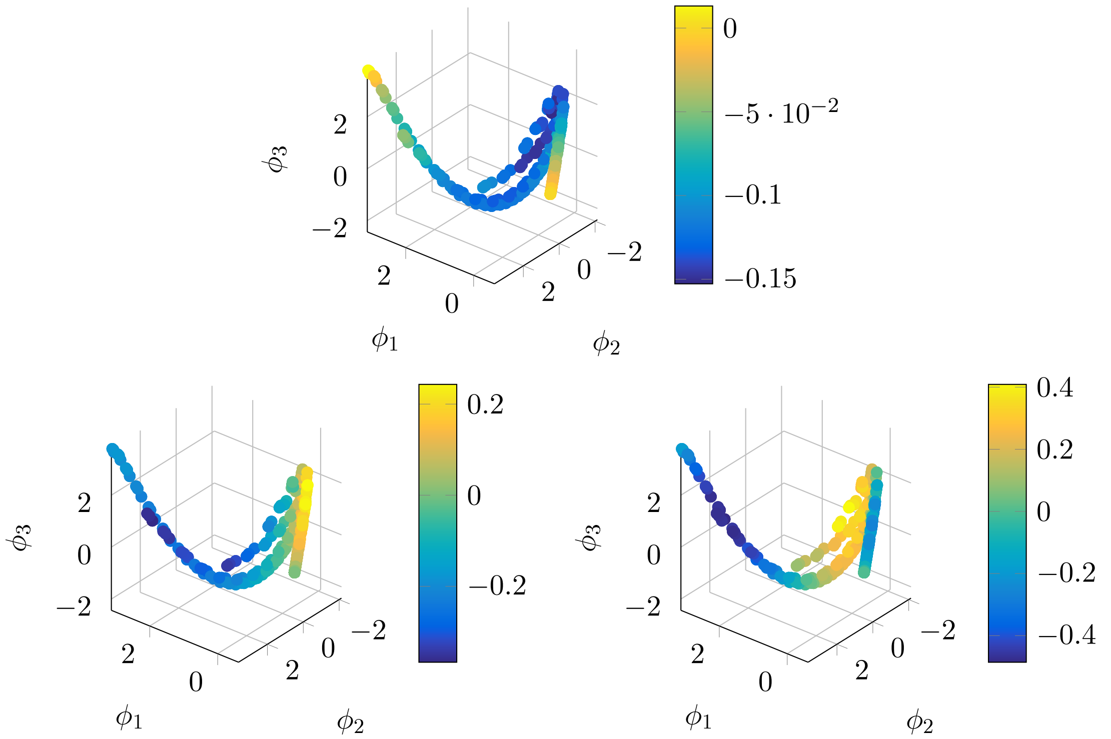

Figure 4.6: The dynamic $g(\phi) = (\frac{d}{dt}\phi_1, \frac{d}{dt}\phi_2, \frac{d}{dt}\phi_3)$, mapping the three closed observables ϕ_1, ϕ_2, ϕ_3 (the coordinates of the plots) onto their time derivative. The value of the derivative $\frac{d}{dt}\phi_j$ is coded in the color of the jth plot.

The score values can be reconstructed over time, figure 4.7 shows results for five different parameter sets and the scores one and five. The reconstruction accuracy is enough to generate density and velocity fields over time, with a relative error of less than ten percent over all time steps (see figure 4.8 and 4.9). The error e is computed relative to the original data, such that for density and velocity (ρ, v) at time t,

$$e = \|(\rho, v)^S(t) - (\rho, v)^O(t)\| / \max_t \|(\rho, v)^O(t)\|, \tag{4.3}$$

where $(\rho, v)^S$ is the data from the surrogate model and $(\rho, v)^O$ is the original, observed data.

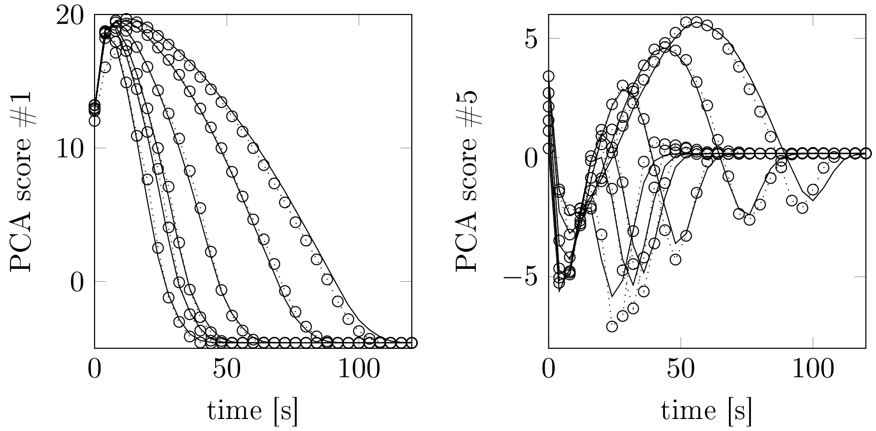

Figure 4.7: Interpolated PCA scores $S_1(t)$ and $S_5(t)$ from the original data (lines) and simulated with the surrogate model (dashed lines with circles), for input parameters $(w, l, h) = ([0.9, 1, 1.4, 1.8, 2.0, 2.5], 4.0, 1)$. Only scores one and five are shown, whereas all of the most important 30 are simulated.

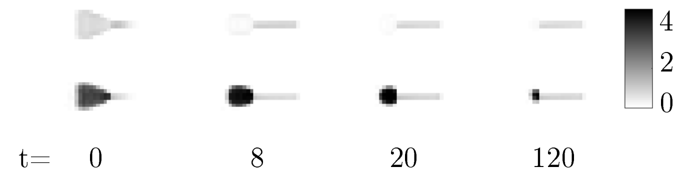

Figure 4.8: Reconstruction of the velocity (first row) and density (second row) over time, shown for the surrogate model at parameter values $(w, l, h) = (0.9, 4.0, 1)$, and $t = 0, 8, 20, 120$ seconds.

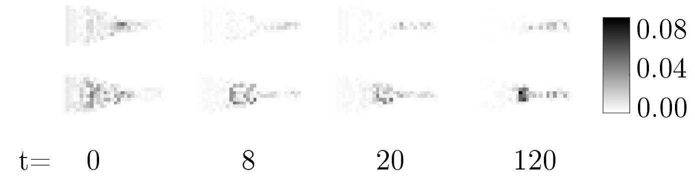

Figure 4.9: Relative reconstruction error over time, $\|surrogate(t) - original(t)\| / \max_t \|original(t)\|$, shown for the same parameters as in Fig. 10.

Conclusions We showed how to apply model order reduction, in particular the method of snapshots, in combination with dynamic surrogate models. Even though the method of snapshots is a commonly used tool in computational fluid dynamics, it is not broadly used in crowd dynamics. The combination allows to construct a three-dimensional surrogate model, which is able to reconstruct the density evolution for the three parameters length, width, and height of the bottleneck. The height parameter was introduced to demonstrate the dimension reduction capabilities of the surrogate model approach, because the parameter does not influence the underlying dynamics, which reduces the dimension of the hidden system. The dimension of the phase space of the surrogate system is reduced accordingly. Even without the surrogate modeling approach, the decomposition of the density and velocity data by the method of snapshots into linear subspaces contains valuable information about the dynamics. In our example, a visualization of the most important principal components helps to understand which parts of the scenario have the greatest impact on the dynamics, in terms of density and velocity. Augmenting model order reduction with the surrogate model then enables to generate trajectories for arbitrary parameter settings inside of the sampled domain. Additionally, irrelevant or dependent parameters are identified through a lower intrinsic dimension of the surrogate model. Here, the height parameter was introduced to demonstrate this.

4.1.2 Queuing in front of an entrance - a bifurcation study

A bifurcation is a qualitative change of system behavior caused by a continuous change of a parameter (Guckenheimer and Holmes, 1986; Kuznetsov, 2004). Bifurcations are present in many systems, and vital to accurate prediction, optimization, and control. Consider a physical system with a bifurcation at a specific value of a parameter, the bifurcation point, and a model for the system that adequately captures this bifurcation. Far away from the bifurcation point, measurements of the parameter might not need to be very accurate for an acceptable result. However, accurate predictions with the model in the vicinity of the bifurcation point are only possible if the parameter is also measured accurately, because small errors can lead to a large, qualitative change of behavior.

General description of the application An example of a system with a bifurcation is the queuing of pedestrians at the entrance of a music festival. Queuing is a behavior ubiquitous in our society. Lately, crowd dynamics research has put a lot of effort into developing models for this process (Kim, Galiza, and Ferreira, 2013; Kneidl, 2015; Zönnchen and Köster, 2016). At the same time, the general concept of queuing theory has been studied in detail for over a century (Erlang, 1909; Kendall, 1953; Kingman, 2009).In this study, we employ data driven surrogate models to study bifurcations in queuing behavior of pedestrians. We analyze the queue at the entrance of a music festival, varying the arrival rate of pedestrians and the service rate at the control point. Figure 4.11 shows the real event, together with a visualization of a simulation with the optimal steps model (originally by Seitz and Köster (2012), we use the version detailed by Sivers and Köster (2015)). The system under study is a corridor of length $90m$ and width $2m$. Pedestrians enter from the left side with an arrival rate of λ pedestrians per second, and walk towards a control point on the right side. Only one pedestrian is controlled at a time, with a service rate of μ pedestrians per second. The mean arrival and service time are the inverse of the rates λ and μ.

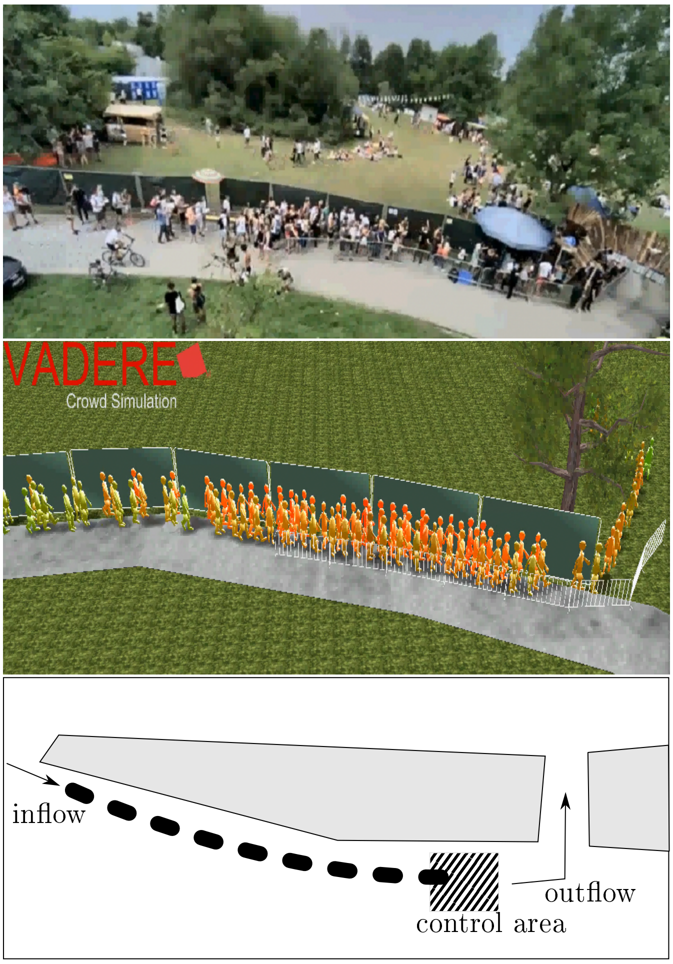

Figure 4.10: Comparison of actual video footage from an entrance to a music festival (top image) with a simulation (center image). The scenario setup of the simulation is shown in the bottom image. (Actual video footage from Back-to-the-Woods, Garching, 2014, filmed in the MultikOSi project))

Pedestrians walk with a free-flow speed of $1.34ms^{-1}$. That means a pedestrian will stay $90m/1.34ms^{-1} \approx 67s$ in the corridor when walking in free flow. If the service rate μ is greater than the arrival rate λ, up to (67λ) pedestrians can be in the corridor without forming a congestion.According to queuing theory, the queuing system only has one effective parameter

$$\rho = \lambda/\mu,$$

a combination of the arrival rate λ and the service rate μ. When increasing observation time of the queue system, the queue length increases to infinity for $\rho \geq 1$, and stays stable for $\rho < 1$. This qualitative change of behavior is visualized in figure 4.11, a bifurcation diagram. We will test whether this bifurcation is present in the given scenario, using the optimal steps model in the simulations. In terms of the complexity graph in figure 4.1, this application is challenging with respect to the correct definition of the bifurcation problem, resulting in a medium equation-driven complexity. Simulating the queuing scenario, and computing the queue length as the number of pedestrians in the scenario, is not particularly challenging. The relation of the simulation results to a real setting is not captured by the comparison of data- and equation-driven complexity, and is a challenge in itself.

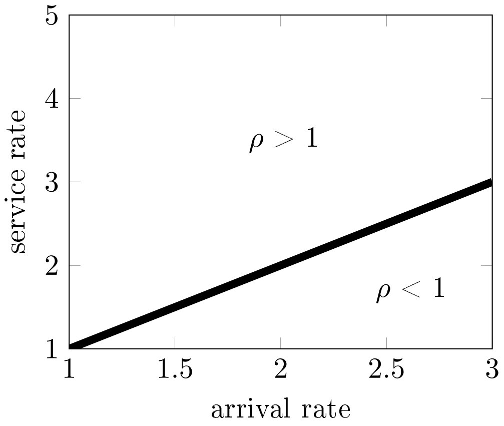

Figure 4.11: Bifurcation diagram for the two parameters of the system, the arrival rate λ of pedestrians and the service rate τ at the control point. For $\rho = \lambda/\tau < 1$, the queue length is stable. For $\rho > 1$, the queue length increases further and further.

After the numerical analysis of the queue, we compare our results with data gathered in a field study of the simulated event. We find that the the simulated queue lengths correspond to the real data. We identify situations where the control points should have been manned more intensively.

The observation data for the calibration of the arrival and service rates was gathered through manual tagging of video data of a music festival in 2016. We find that interarrival and service times are distributed exponentially. Figure 4.12 shows the observed times between two arrivals and controls at the control point. We find the arrival rate λ through the Poisson process

$$P_n(t) = \frac{(\lambda t)^n}{n!} \exp(-\lambda t) \tag{4.4}$$

for $n = 1$, where P_n is the probability for time t between n arrivals. The simulation software VADERE accepts the mean interarrival and service times as an input, which are inverse to λ and μ. The software then uses the negative exponential distribution $P_0(t) = \exp(-\lambda t)$ to find the time between arrivals, and services, analogously.

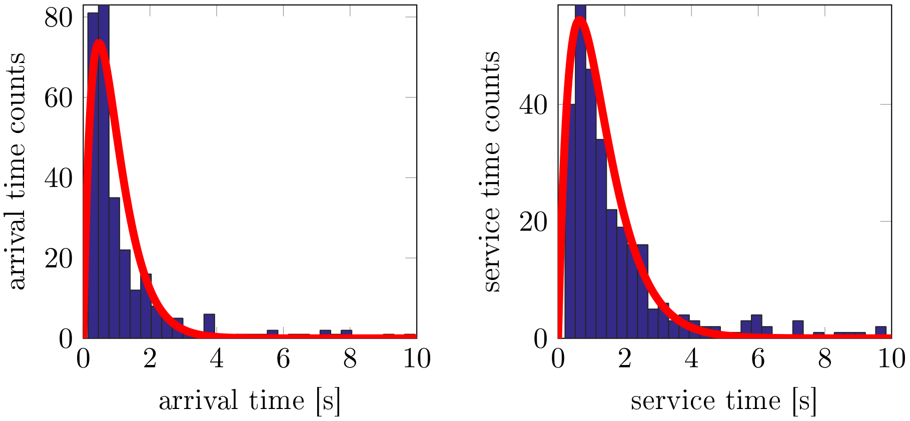

Figure 4.12: Left: distribution of the time between two arrivals. Right: control time for a person after the previous control is complete. We excluded data where the control point was idle. These two distributions allow to compute the arrival and service rates λ and μ.

State-of-the-art approach without dynamic surrogate models If queuing theory could be applied directly, one would only need to measure the arrival and service rates to compute the average queue lengths, the average time a pedestrian spends in the queue, and the average server load, through the effective parameter $\rho = \lambda/\mu$. However, it is not clear yet whether queuing theory yields accurate results in crowd dynamics. Reasons that might hinder accurate predictions through queuing theory might be finite size effects in front of the control point, such as people blocking each other, or social interactions between persons waiting in the queue. These effects have an influence, because persons entering the scenario do not directly enter the control point, and hence cannot immediately exit the scenario again. If these additional effects occur in a systematic way, the distribution of arrival times at the control point might differ significantly from the distribution of arrival times in the scenario (see the bottom of figure 4.10).

The standard approach without utilizing queuing theory would be a sampling of the parameter space, computing the queue lengths for each sample, and constructing the bifurcation diagram. A more sophisticated approach would use equation-free techniques with projective integration, to wrap the microscopic model into an outer code that performs the bifurcation analysis. This way, longer steps in time can be taken due to continuity on the macroscopic scale. Also, efficient bifurcation code allows to sample the parameter space more coarsely, thus avoiding many calls to the microscopic simulator. An example of this method for bifurcation analysis in traffic flow can be found in the paper of Marschler et al. (2014).

With the methods above, all simulation results are lost as soon as the bifurcation diagram is complete. The analysis of the parameters takes no dynamical information into account. That means even if all parameters lead to a similar dynamical progression, the methods in the state of the art do not make use of this information, and are not able to uncover hidden relations between the parameters. In the queuing scenario, only one parameter ρ affects the bifurcation. The constructed surrogate model incorporates this in addition to the dynamical information through its two-dimensional phase space, as shown next.

Alternative approach with the dynamic surrogate model We sample the arrival and service rates given in table 4.3. To reduce noise, we run ten simulations per parameter pair, and hence generate data from $9 \cdot 9 \cdot 10 = 810$ runs in total. The number of pedestrians in the scenario per time step is then averaged, resulting in 81 time series, one per parameter sample. A transient phase of $70s$ at the beginning of the simulations is ignored, because in this phase, pedestrians only move towards the control point, and no queuing behavior is possible.

Table 4.3: Arrival and service rates sampled with the optimal steps model in the queuing scenario.

arrival rate λ	**service rate** μ
$1.0, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5, 2.75, 3.0$	$1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0$

Figure 4.13 shows the time series generated through microscopic simulation, observation of the number of pedestrians present in the scenario, and averaging.

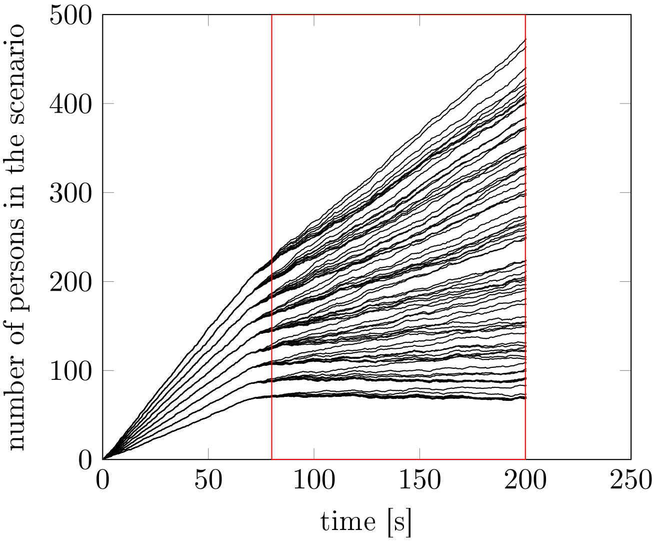

Figure 4.13: Simulated data for the pedestrian count at the entrance of the music festival, over time. The red box surrounds the data we use to generate the surrogate model.

Given the data visualized in figure 4.13, we construct a data-driven surrogate model, which is able to accurately reproduce the queue length in the red box over time, for any given arrival and service rate in the ranges $[1, 3]$ and $[1, 5]$, respectively. We find that the surrogate model needs two-dimensional closed observables to predict the queue length over time. Considering the two-dimensional parameter space, the analysis of storage discussed in section 3.2.5 suggests that the two parameters are not independent, and that only one, effective parameter drives the system dynamics. This is in accordance with queuing theory as discussed above, where the effective parameter can even be stated precisely, as $\rho = \lambda/\tau$. Figure 4.14 shows the map from service and arrival rates (λ, μ) into the state space of the surrogate model (ϕ_1, ϕ_2). This initial map contains crucial information about bifurcations in the system, as any change in the parameters (λ, μ) corresponds to a change in the trajectory in the intrinsic space of the surrogate model, starting at $\phi_0(\lambda, \mu) = \phi(0)$. The simulation results from the surrogate model agree reasonably well with the original data (see figure 4.15). The linear increase of the number of pedestrians present in the scenario is clearly visible, and also reproduced by the surrogate model.

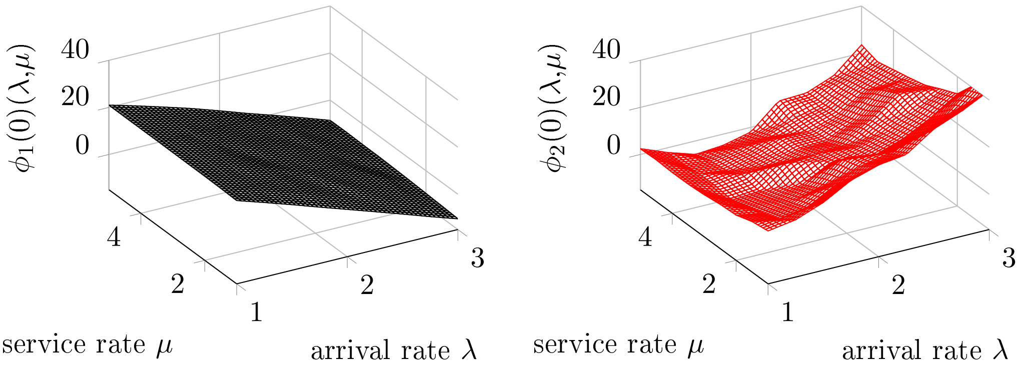

Figure 4.14: Initial value map $\phi_0(\lambda, \mu) = (\phi_1(0), \phi_2(0))$. The black surface shows $\phi_1(0)$, the red surface $\phi_2(0)$.

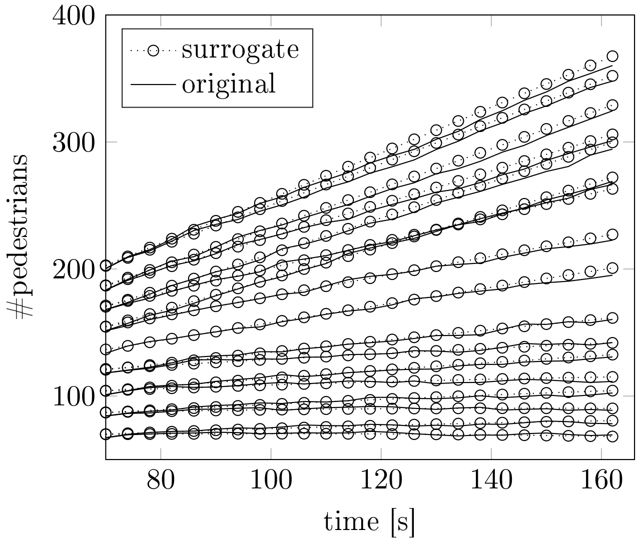

Figure 4.15: Number of pedestrians in the scenario, as a measure of queue length, over time. The simulation results of the surrogate model (circles) are compared to original observations used to create it (line). Only every fifth trajectory is shown.

Regarding bifurcations, we find a similarity to the predictions of queuing theory. Figure 4.16 shows the queue lengths up to $T = 166$ seconds, generated through $9 \times 9 = 81$ simulations with the surrogate model (left plot), as well as the data from the optimal steps model that was used to construct it (right plot). The results between left and right plot agree up to maximum error of 10.7 pedestrians, and a mean error of 3.6 pedestrians, over all initial conditions. This demonstrates the accuracy of the surrogate model compared to the original data. Also, the bifurcation line predicted by queuing theory (black line) is very similar to the points where the queue lengths do not differ more than the error of ± 10 pedestrians (white line).

The prediction by the surrogate model is closer to the original data than the predictions by queuing theory. This suggests the surrogate model should be used to predict queue lengths in this setting, instead of the rather general queuing theory, which does not include possible finite size effects in front of the control point.

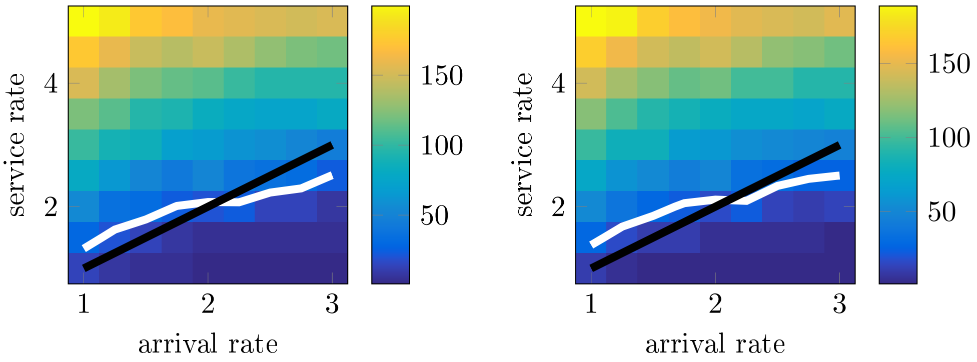

Figure 4.16: The queue lengths (color coded) generated by the surrogate model (left plot), compared to the same analysis performed on the data used to generate the surrogate model (right plot). The black line shows the prediction of the bifurcation by queuing theory, the white contour is the approximation of the bifurcation by the simulated data. The contour is shown where the queue lengths minima and maxima only differ by ±10 over the complete simulation time, that is, where the differences in time is smaller than the error between the surrogate and the original data.

Conclusions We confirmed that queuing theory is a reasonably accurate, first guess on the bifurcation behavior of queuing for the given microscopic model. In situations where high accuracy is not an issue, the theory can therefore be used to make predictions regarding queue lengths. We also confirmed that the data-driven surrogate model could reconstruct and predict queue lengths closer to the original model than queuing theory. This validates the surrogate model, and demonstrates that it can be a valuable tool for bifurcation analysis. The results make it possible to study larger and more complex systems with the powerful tools of numerical analysis. Accurate, data-driven surrogate models can be used to predict behavior of crowd flow with the speed needed for optimization and control, even in on-site situations.

4.1.3 Uncertainty quantification

Uncertainty quantification (UQ) reveals how uncertainties in the parameters of a system are related to uncertainties in the systems behavior and output. We refer to (Iaccarino, 2008) for a presentation of methodology, and to (Smith, 2014; Sullivan, 2015) for an in-depth discussion of the topic. In so-called *forward* uncertainty quantification, parameters are assumed to be distributed stochastically, and the effect of particular distributions on relevant observables of the system, the *quantities of interest*, is analyzed (see figure 4.17). Conversely, *inverse* uncertainty quantification tries to find the distribution of parameters, given the distribution of the quantities of interest. In most applications, only discrete samples or the first few moments of distributions are available. This holds for the quantities of interest in forward UQ and the reconstructed parameter spaces in inverse UQ. The reconstruction of the distribution given only this little information is also a topic of active research, and in most cases, only the first moments can be reconstructed with enough accuracy.

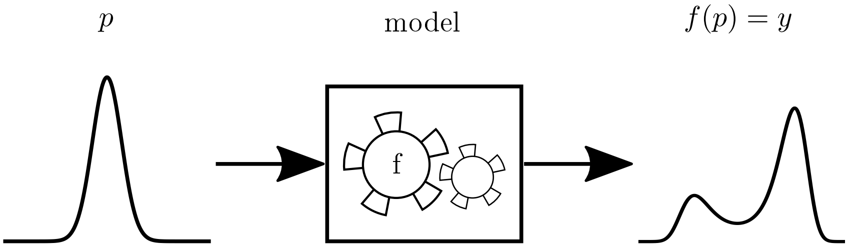

Figure 4.17: Concept of forward uncertainty quantification, searching for the distribution of y. The input parameter p is stochastically distributed, and when processed by the model f, the quantity of interest y will also have a certain distribution $f(p)$, which is result.

Uncertainty quantification operates under assumptions on the stochastic distribution of the parameters (forward UQ) or the quantities of interest (inverse UQ). This is different from sensitivity analysis (Saltelli, Tarantola, and Chan, 1999), which is characterizing the response of a system to a change in parameters without assumptions on uncertainties. A large response to small changes in a parameter might not be relevant for the quantification of uncertainty of the system, if the parameter in question is not uncertain (Iaccarino, 2008). Moderate responses to large changes might still be important, if large uncertainties are present. Here, we demonstrate how to apply uncertainty quantification combined with surrogate models in a concrete example.

General description of the application Public transportation systems are invaluable for modern society. Especially important are train stations, where many passengers move and interact in a large area. Prediction of crowd flows, optimization of facilities, and, ultimately, control of the system are three important goals of crowd dynamics research. Given correct input parameters, modern simulation software for crowd dynamics can accurately simulate the flow of pedestrians in these situations. However, a great challenge are uncertainties of the system that remain until the very moment control should be exercised. These uncertainties include the number of people present in the system, the targets of these people, and which characteristics they possess, such as desired speeds or disabilities.

Here, we propose an integrated approach combining uncertainty quantification and surrogate modeling into a concept we call real-time uncertainty quantification (real-time UQ). Essentially, real-time UQ enables the quantification of uncertainties in real time, even for large systems, such as a train station. Real-time UQ is impossible with current microscopic simulation software, as it requires the evaluation of thousands or even millions of complete scenario simulations in a matter of seconds. We solve this challenge through precomputation of a data-driven surrogate model using the microscopic simulator. The surrogate model can then be queried independently by the software performing

UQ. Figure 4.18 visualizes the train station we use to demonstrate our method. The quantities of interest are the number passengers on a train and on the platform over time. We assume that in the real scenario, the number of passengers initially on and off the train are uncertain, and the safety officer can only guess the number of people to a certain degree. We model this guessing process through the stochastic distributions of the input parameters. The task for the hypothetical safety officer in this case is to estimate if and when the number of people on the platform exceeds 120. This number is chosen because it marks the point where the density on the platform exceeds one person per square meter, which we set as a point where safety guidelines require an evacuation of the platform before more passengers can deboard.

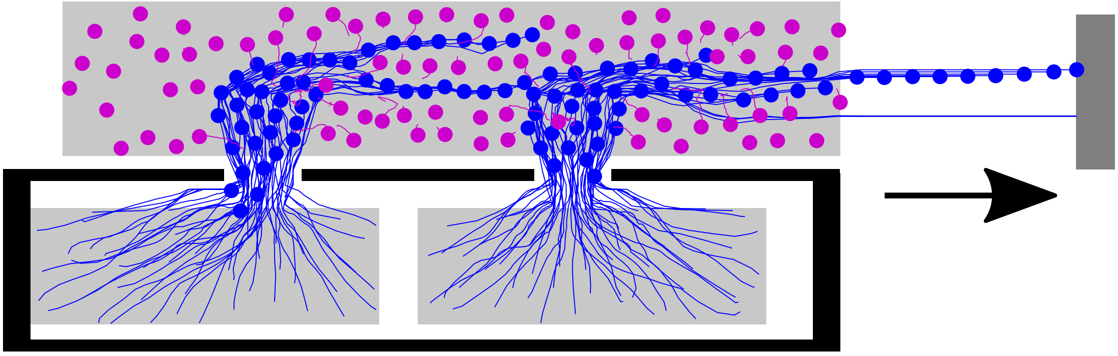

Figure 4.18: Train station scenario used to demonstrate uncertainty quantification with surrogate models in a realistic context. Shown here are 100 passengers, 50 per door, deboarding a train onto a platform where 80 passengers wait. The passengers start at the two gray boxes in the train, move out of the train, and then to the right towards an exit. Waiting passengers move aside when too close to a moving passenger, but remain on the platform and do not board the train.

We discuss and answer two questions for the train station scenario, one about control, the other about methodology:

1. Control: How many people should be allowed to remain on the platform before a train arrives?

2. Methodology: How can uncertainty quantification handle several different input probability distributions?

We discuss how current methods in uncertainty quantification provide an answer to these questions. Then, we show how to combine uncertainty quantification with data-driven surrogate models, including drawbacks and benefits.

State-of-the-art approach without dynamic surrogate models We assume the equations of the microscopic model are not available, and we can only generate the macroscopic data by initializing the simulator with a given parameter set, and measuring its output over time. Data-driven methods in uncertainty quantification are able to handle this setting. Iaccarino (2008) classifies methods for probabilistic uncertainty quantification by sampling techniques and quadrature methods. Spectral methods are mentioned as a class of equation-driven approaches, which is not discussed here.

Sampling techniques include the Monte Carlo method and Latin hypercube sampling, which provide reasonable accuracy under very few assumptions on the data. Essentially, they approximate the distribution of parameters by drawing random samples, and estimate the true probability distribution of the quantity of interest by computing a histogram of the results. The convergence to the true solution does not depend directly on the dimension of the parameter space, which makes sampling methods especially efficient for high-dimensional spaces. Still, convergence might take a prohibitively large number of samples. Schröder et al. (2014) use Latin hypercube sampling to perform a sensitivity analysis of a fire hazard in a train station.

Quadrature methods are more advanced and highly accurate techniques. They can be used for uncertainty quantification, because the modes of the probability distribution are integrals of the quantities of interest over parameter space. Stochastic collocation is a commonly used method for the estimation of integrals of random variables. Assuming one particular distribution for each of the parameters, the usual approach in stochastic collocation computes collocation points in parameter space and defines associated orthogonal polynomials. Then, microscopic simulations generate data with each collocation point as an input. The resulting quantity of interest is assembled so that

$$\sum_{i=0}^{M} f(\zeta_i) w_i(\zeta_i) \approx \int_P f(\zeta)\rho(\zeta) d\zeta, \tag{4.5}$$

where ζ_i are the collocation points in parameter space, $f(\zeta)$ computes the quantity of interest given a parameter ζ, w_i are quadrature weights, and ρ is the true probability density. Extensions to multidimensional parameter spaces are possible through tensor product approaches, or sparse grids.

All methods discussed above assume a fixed probability distribution of the parameters. High-dimensional parameter spaces are difficult to assess, because it is not yet clear how to choose the collocation points and associated polynomials correctly.

It is not possible to change the probability distributions assumed for the parameters after the data was sampled. If a new distribution needs to be chosen, the whole process has to be repeated. The new collocation points might be different from the previous ones, so all microscopic simulations have to be repeated, too. Even though the standard methods characterize the influence of uncertainty in the given parameters on the quantities of interest, they do not uncover possible relations between the parameters. For example, in the queuing scenario discussed in section 4.1.2, the parameters are arrival rate λ and service rate μ, whereas the quantity of interest–the length of the queue–can be described through only one effective parameter $\rho = \lambda/\mu$. An analysis of the influence of uncertainty in λ and μ would not reveal this relation. The data-driven surrogate modeling approach described in the next section enables to overcome these two challenges. After the construction of the surrogate, we will analyze three different probability distributions without querying the microscopic model, and demonstrate the accuracy of the approach.

Alternative approach with the dynamic surrogate model With the surrogate model, we can also adapt to errors intrinsic to the dynamic, not only the input distribution. The surrogate "stores" all information gathered by previous microscopic runs. This means we are able to use several different probability distributions in the quantification of uncertainty, after the model is created. To create the surrogate model, we sample the two input parameters in a grid of 5×5 points as shown in table 4.4, and run the microscopic simulation for 200 seconds at each point.

Table 4.4: Number of passengers on train and platform, used as parameters used to create the surrogate model, which is then analyzed in uncertainty quantification.

#platform	$0, 25, 50, 75, 100$
#train	$10, 25, 50, 75, 100$

To demonstrate that the output of the surrogate model can again be processed into new quantities of interest, we do not directly observe the number of passengers on and off the train, but in four sections of the scenario: two areas in the train, and two adjacent areas on the platform. The four numbers at each point in time can be processed into the coarser information of "number of passengers on the train" and "number of passengers on the platform" by summing the numbers of the two respective areas. By measuring four instead of just two areas, the safety officer in charge of the station could then display a more accurate picture of the deboarding process, after a courser analysis. The choice of these four areas also demonstrates that it is quite arbitrary which observables of the system are chosen, as long as they are not too heavily affected by noise. The exact requirements on the observations are discussed in section 2.3.

We construct the surrogate model with the generated data. The data in its original form is affected quite heavily by microscopic effects, which manifest themselves as noise on the macroscopic scale. To reduce this noise as required by Takens' theorem, we perform the following preprocessing steps, and parameter choices in the construction of the surrogate:

- Only use data points every 2.5 seconds.
- Use the number of passengers that already left the scenario as an additional observable, which can be computed from the given four observables. This fifth observable has a stabilizing effect on the dynamics, as well as the interpolants, as it is a monotonically increasing function of time.
- Choose 80 delays in time to smooth the manifold in delay space. By choosing data points only every 2.5 seconds, this accumulates the complete simulated time of 200 seconds in one data point. We append the last point in time of each trajectory

for another 200 seconds, assuming a steady state has been reached, to be able to reconstruct the whole trajectory over time again.

- Choose $\kappa = 1 \times 10^{-3}$, which is quite large considering the large number of delays.

We compute the diffusion map with scale parameter $\epsilon = 1 \times 10^7$, and use linear interpolation to approximate the input map, the dynamic, and the output map of the surrogate model (Python package `scipy.interpolate.LinearNDInterpolator`, with parameter rescaling, and a fill value of 0). Figure 4.19 shows a comparison of simulation and original data. The results agree reasonably well, and the steady state for each trajectory is also reached, which was introduced by adding constant data after $t = 200$.

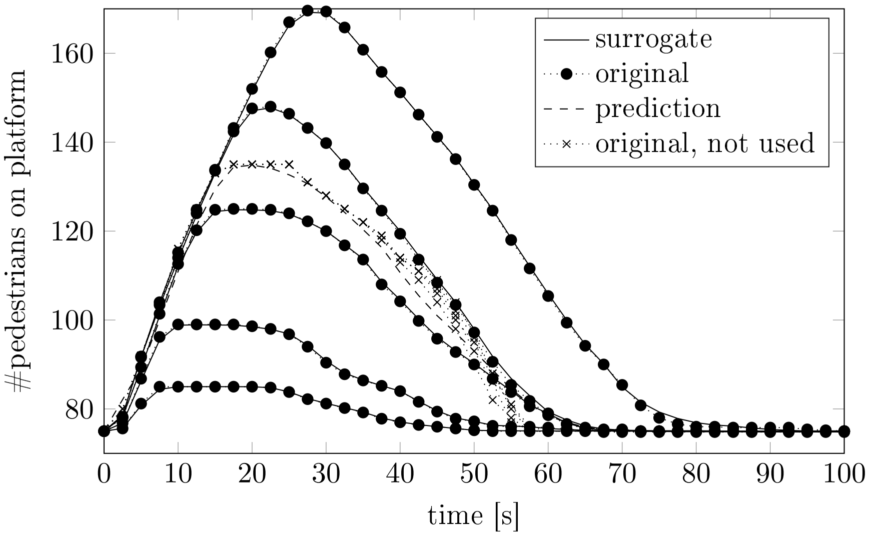

Figure 4.19: Comparison of the simulation results of the surrogate model and the microscopic model, using linear interpolation for the functions comprising the surrogate model. A prediction is shown for $(\#platform, \#train) = (75, 60)$, a parameter set not used in the construction. The prediction agrees well with five trajectories sampled at this parameter value.

Performing data-driven uncertainty quantification with the surrogate model is not different from the microscopic model, but the surrogate model generates output much faster. This computational efficiency allows to use less advanced methods that are usually quicker to implement, such as Monte-Carlo sampling compared to stochastic collocation. Table 4.5 describes the three distributions of the parameters that we use to demonstrate uncertainty quantification with surrogate models. We estimate each distribution with 1000 random samples, and generate the quantities of interest over time with the surrogate model. For one distribution, the whole process takes about 50 seconds, utilizing all cores on a quad core machine (Intel i7 920, 2.67GHz with hyperthreading, 8GB RAM). In comparison, a simulation of 100 pedestrians with the gradient navigation model on the same machine needs approximately 20 seconds for a simulated time of 200 seconds, also utilizing all cores. Thus, the surrogate exceeds the microscopic model by a factor of 400 in terms of computational speed.

Table 4.5: Three distributions used in uncertainty quantification with the surrogate model. We write $U(a, b)$ for a uniform distribution between a and b, and $N(a, b)$ for a normal distribution with mean a and standard deviation b.

	#platform	#train
distribution A	$U(30, 70)$	const. 50
distribution B	const. 50	$U(30, 70)$
distribution C	$N(50, 5)$, truncated to $[5, 95]$	$N(50, 15)$, truncated to $[5, 95]$

Figure 4.20 shows the propagation of the quantities of interest through the system over time. The graphs show the mean, the 90%-quantile and 10%-quantile of the distribution as it is propagated. The upper graph (blue) shows the number of passengers on the platform, the lower graph (gray) shows the number of passengers still on the train. The results are shown for the surrogate model in the left column, and for a separate analysis with stochastic collocation in the right column. The uncertainty quantification with stochastic collocation was performed by Florian Künzner, we thank him for the collaboration and his permission to show his results. Distribution A affects the initial number of passengers on the platform. It does not change shape as passengers deboard the train, only its mean increases, and then decreases as the deboarding passengers leave the scenario. The number of passengers on the train slightly differs over time, due to the different number of passengers in front of the train doors. Distribution B affects the initial number of passengers on the train. This number decreases for all plots, for the second distribution, the mean and the 90%-quantile almost agree at the point where almost no passengers are left on the train. Distribution C affects both the initial number of passengers on and off the train, with a truncated normal distribution for both initial values around a mean of 50. We choose the standard deviation for the guess of the number of people in the train three times higher than for the number of people on the platform, because it is easier for a safety officer at the station to guess the first number. In this case, after about 15 seconds, the number of passengers on the platform is higher than the limit of 120. Figure 4.21 shows the probability density at $t = 20$ seconds. The probability of more than 120 passengers is highlighted, and indicates that an evacuation is necessary. A safety officer facing this situation should stop the train from entering the station, and evacuate the platform.

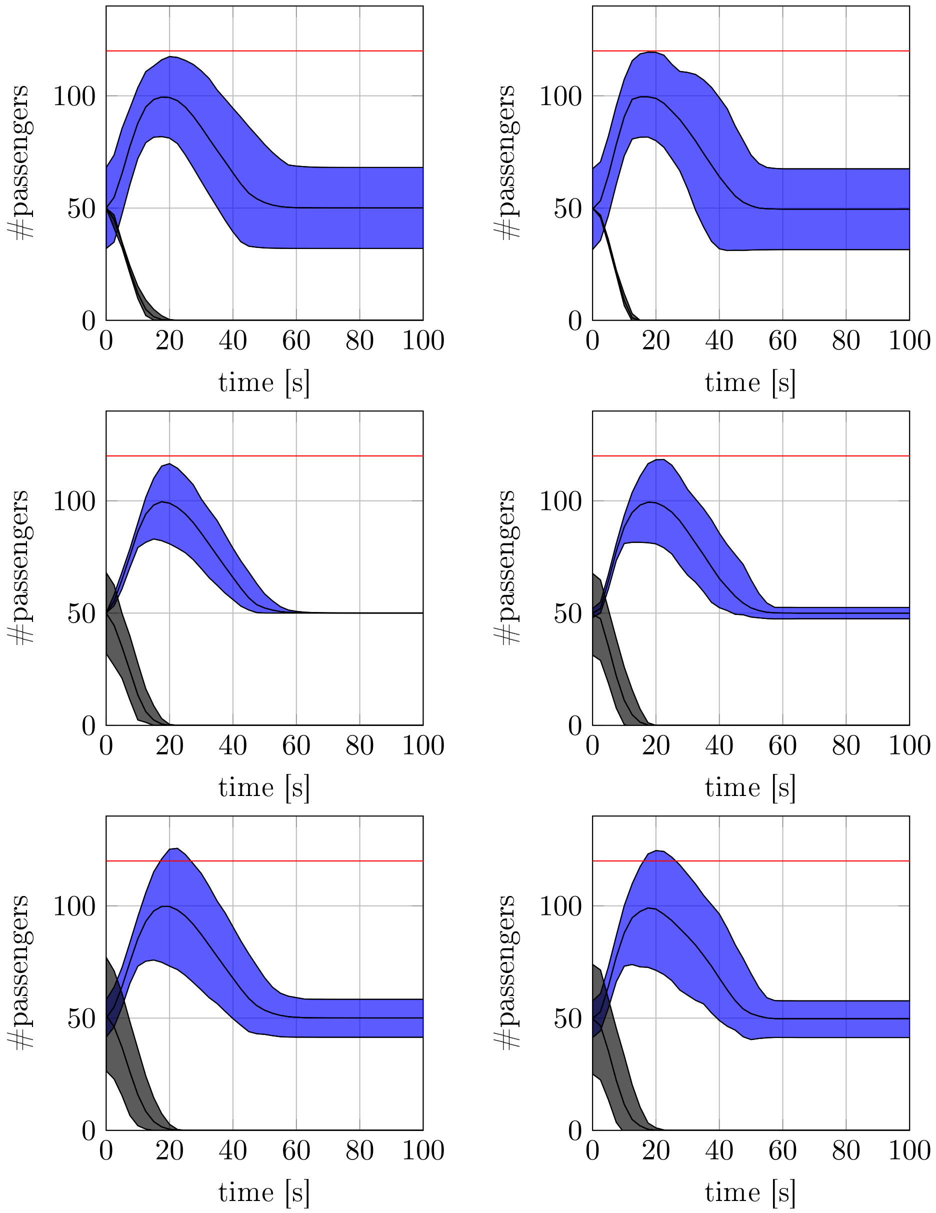

Figure 4.20: Results of uncertainty quantification with the surrogate model (left column) and stochastic collocations (right column), for three probability distributions (rows) over the two parameters $N_P(0)$ and $N_T(0)$. For distribution C, the number of passengers on the platform exceeds the safety limit at $t = 15s$, which would lead to an evacuation of the platform.

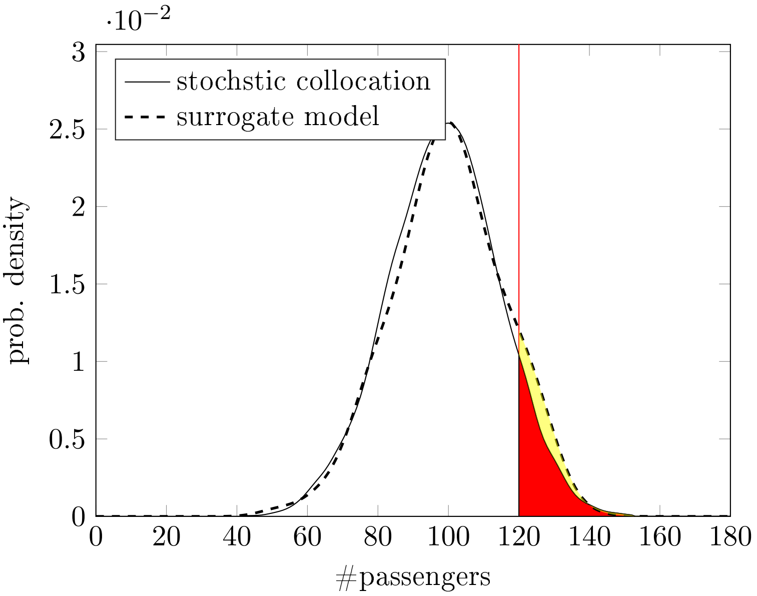

Figure 4.21: Probability density of the number of passengers on the platform at $t = 20s$, given the parameter distribution three. The probability of the number being higher than 120 is highlighted in red for the Monte-Carlo simulation, and overlayed in yellow for the prediction of the surrogate model.

A low number of simulation runs with the microscopic simulator is needed for the surrogate model and stochastic collocation. It is difficult to state in advance the minimum number of simulations necessary to accurately predict results from a brute-force approach. That difficulty is a cause for the results shown in figure 4.22. A Monte-Carlo sampling with 1000 sample points was performed for distributions 1 and 2, and a microscopic simulation run for each sample. If we use the results as a more accurate picture for uncertainty propagation in the system, we can evaluate how well both the surrogate model and stochastic collocation perform in predicting the propagation of the two distributions. The propagation of distribution B (right side of figure 4.22) seems to be predicted quite well through both approaches (see figure 4.20, top row). For distribution A (left side of figure 4.22), the number of passengers on the platform exceeds 120 in the Monte-Carlo simulation, but stays below 120 for predictions of both the surrogate model and stochastic collocation.

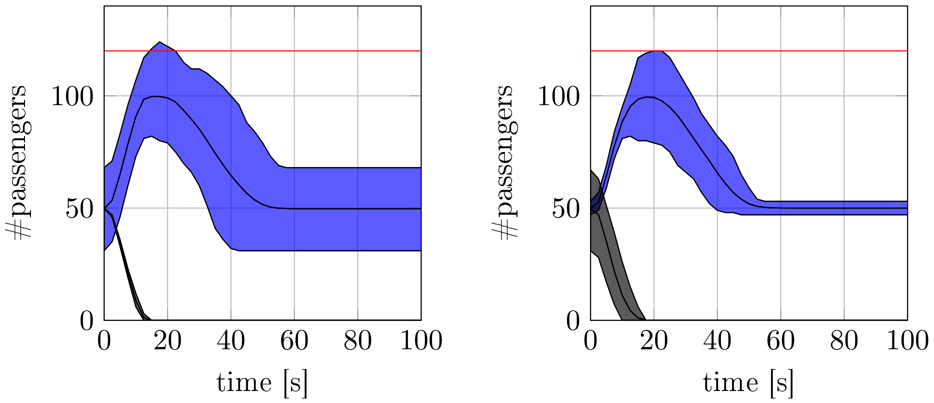

Figure 4.22: Results of uncertainty quantification with a Monte-Carlo approach, for the probability distributions A and B, over the two parameters $N_P(0)$ and $N_T(0)$. Distribution A exceeds the critical number of 120 passengers on the platform at $t = 15$, which is different from the predictions of the surrogate model and stochastic collocation.

Conclusions Uncertainty quantification is a valuable tool to help understand and control the flow of passengers in a train station. We demonstrated how the data-driven surrogate can be used as an intermediate step between the original model and uncertainty quantification. We queried the microscopic simulator $(5 \cdot 5) \cdot 5 = 125$ times to generate the observations for $(5 \cdot 5) = 25$ parameter settings, and 5 random initial conditions per setting. From the generated data, we construct the surrogate model. No additional calls are necessary for the evaluation of arbitrary parameter distributions with support in the range of the parameters used in the construction. This is different for an approach with stochastic collocation, where a change of the probability distribution in parameter space always requires new simulations of the microscopic model. The distributions over time of the quantities of interest could be reproduced accurately by the surrogate model, compared to results from stochastic collocation.

Even though the combination of the surrogate model with uncertainty quantification is a very promising approach, there are still many open questions to answer. Approximations through the construction of the surrogate model introduce errors in the output. It is not clear how exactly these errors influence the propagation of uncertainties through the system. Theorem 3 on the numerical approximation error can only give a first insight into the matter, because it assumes very small magnitudes of errors and also does not specify the magnitude of the constants. There are also numerous improvements possible to the modeling of the deboarding process shown here, which have to be addressed before using the scenario in simulations of a real train station. Passengers on the platform would usually make way for deboarding passengers beforehand, speeding up the process. Groups of passengers deboarding together, such as families or friends, usually also affect the dynamics of the system. Incorporating the mean of the desired speeds of passengers as a parameter of the surrogate model would allow safety officers to also include knowledge about the distribution of passenger types into the real-time UQ process. A train with many tourists and families would be represented by an input distribution of mean speeds around a low value, whereas in a train with commuters the mean speeds would be distributed around a higher value.

As the construction of the surrogate model is a data-driven approach and independent of the actual example, all changes to the underlying, microscopic model can be used directly and without changes to the construction algorithm. There also is a benefit of the surrogate model concerning equation-driven uncertainty quantification. The functions approximating the input map, dynamic, and observer of the surrogate model are often given explicitly, compared to closed-source or legacy simulation software. The explicit formulation allows to use invasive, equation-driven methods on the surrogate, even though the original model is not accessible in such a way. For the train station example discussed here, we use piece-wise linear interpolants for all three functions. As the formulaic description of the interpolants is known, a Galerkin approach could be used. For applications, this equation-driven approach following the construction of the surrogate model could mean an even greater increase in performance.

4.2 Car traffic

General description of the application A large portion of modern society relies heavily on transport by cars. Even though public transport systems are capable of handling a large number of passengers and goods, cars are still prominent on highways and in cities. One of the most studied phenomena in traffic flow on highways is the occurrence of stop-and-go waves. Figure 4.23 illustrates such a stop-and-go wave for three consecutive time steps. The gray cars are initially in a slow "stop" phase, so that cars behind them also have to slow down. The rightmost gray car can start to accelerate, and move into the "go" phase in the third step. The slow part of the wave has moved forward, but slower than the rightmost gray car.

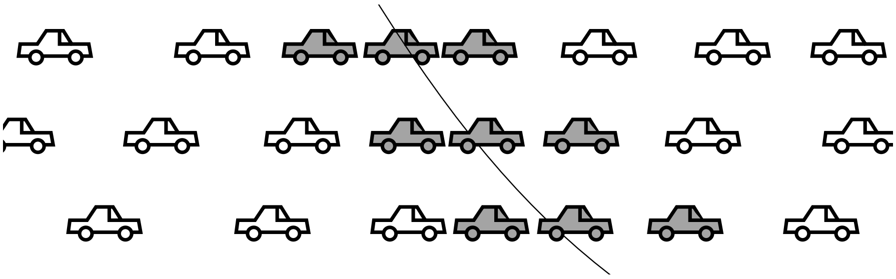

Figure 4.23: Cars on a highway, the three rows show three consecutive time steps. The line indicates the point of highest density, where cars almost stop, and accelerate after reaching the end of the jam. The point of highest density shifts over time, creating the effect of a moving stop-and-go wave.

In the example we study here, N cars drive on a ring of arclength L, where individual driving behavior is governed by the optimal velocity model (Bando et al., 1995), with a numerical model implemented in MATLAB (2015b). If x_n is the position of car n, and τ is its inertia, car n moves according to

$$\tau \ddot{x}_n + \dot{x}_n = V(\Delta x_n), \; n = 1, \dots, N. \tag{4.6}$$

Here, $\Delta x_n = x_{n+1} - x_n$ is the distance to the car in front of car n, and V is the velocity function

$$V(\Delta x_n) = v_0(\tanh(\Delta x_n - h) + \tanh(h)) \tag{4.7}$$

shown in figure 4.24. The maximal velocity reached in the limit of infinite time is $v_0(1 + \tanh(h))$. The parameter h is the inflection point of V, and describes the desired safety distance between two cars. All physical units are removed through nondimensionalization, such that a car with velocity $\dot{x} = 1$ always takes L units of time to move around the circle of length L once.

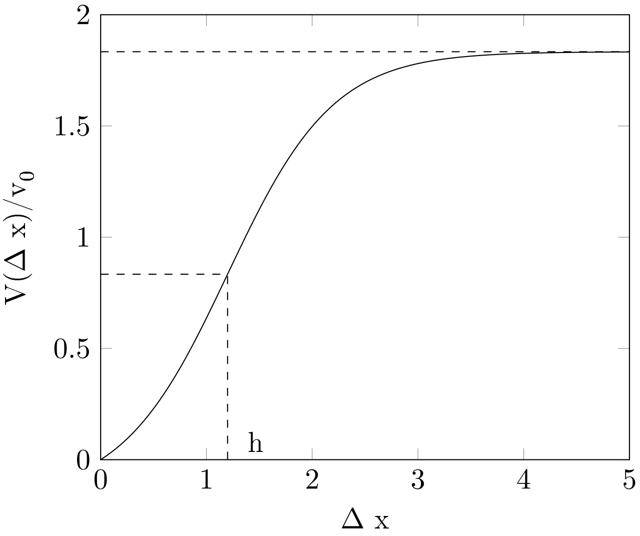

Figure 4.24: Velocity function V of the optimal velocity model used here, including the inflection point h and the maximum $1 + \tanh(h)$.

Initially, the cars are distributed uniformly, with a gap of size $(1 - d)/L$, $d \in [0.8, 1]$, in front of the first car. Over time, the gap closes, and the cars move around the ring road with only small changes of the headway $x_{n+1} - x_n$ between the two cars $n + 1$ and n. Figure 4.25 shows the headway distribution for $N = 60$ cars over the whole ring, for two values of the safety distance h, and for two time steps each.

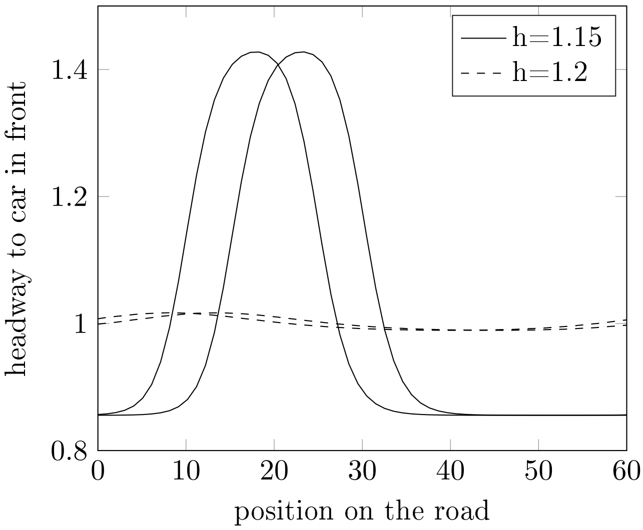

Figure 4.25: Headway distributions for car n at position x_n to the car $n+1$ in front. The time evolution of two distributions are shown, for desired safety distance $h = 1.15$ and $h = 1.2$, and with a time difference of $6s$. Both distributions move to the left over time and keep their shape, but have different maxima for different h.

Here, we show how to construct a surrogate model for stop-and-go behavior of cars on a highway. The system can be reduced to a single point, rotating on a circle with a certain constant speed and radius. This special case allows to find a formulaic description of the state space and the dynamic of the surrogate model. To make the formulaic description clear, we do not introduce specific equation-driven or data-driven challenges here. The data-driven complexity of this application is very low, as we use a simple ODE model to produce the data for the cars, without any additional effects or differences between cars. The goal of this section is an equation-driven method, the formulaic description of the surrogate model, that is inspired by the numerical results.

State-of-the-art approach without dynamic surrogate models Similar to the bifurcation application on queues, stop-and-go behavior depending on the safety distance h and velocity v_0 can be studied through an equation-free bifurcation analysis (Marschler et al., 2014). Again, this would avoid many calls to the microscopic simulator, but all simulation results are lost as soon as the bifurcation diagram is ready. When the bifurcation diagram is constructed with an adaptive method, new parameter sets for a new simulation have to be chosen based on previous results. The choice is usually based only on the available diagram data, and the last solution that was computed. One example is numerical arc-length continuation (Brindley, Kaas-Petersen, and Spence, 1989), which continues the bifurcation path from known points. Only the previous solution and parameter value on the path are taken into account, not the whole dynamical information leading to the diagram. That means even if all parameters lead to a similar dynamical progression, the methods do not make use of this information. We refer to (Orosz, Wilson, and Krauskopf, 2004; Marschler et al., 2014) for an application of a numerical continuation of bifurcation curves in the optimal velocity model.

Another possibility for a bifurcation analysis of the microscopic model is to homogenize it with equation-driven methods. In the optimal velocity model, the equations on the microscopic scale are explicitly available and not very complex, so homogenization is a straight-forward approach (see (Helbing, 2001) for a review of traffic modeling). Equation-driven methods are preferable in case the microscopic model or phenomenon needs to be understood more thoroughly, and we will show how the surrogate model can be used to extract information from data that can then inform a formulaic description on the macroscopic scale.

Alternative approach with the dynamic surrogate model We consider three parameters of the system: the velocity v_0, the desired safety distance h, and the initial distribution parameter d (see table 4.6). The range of the parameters h and v_0 is chosen around the bifurcations found by Marschler et al. (2014).

Table 4.6: Nondimensionalized parameter values, sampled to generate the surrogate model for stop-and-go behavior in car traffic. The values are in the same range as the bifurcation analysis by Marschler et al. (2014).

parameter	values
v_0	$0.88, 0.885, 0.89, 0.895, 0.9$
h	$1.15, 1.175, 1.2, 1.225, 1.25$
d	$0.8, 0.85, 0.9, 0.95, 1.0$

We generate trajectories up to $t = 1 \times 10^5$, and store observations of the car density over the whole ring. We omit all but the last 1×10^3 seconds, to avoid transition effects towards the dynamic steady state. Depending on the parameters, this dynamic steady state is either a homogeneous spatial profile with constant headway, or stop-and-go behavior with varying differences in headways (see Fig.4.25). The dynamic of the resulting surrogate model is the same as a point moving on a circle, with constant speed and radius. Figure 4.26 visualizes the time derivative given the first two diffusion map coordinates, for all data points used to create the surrogate model.

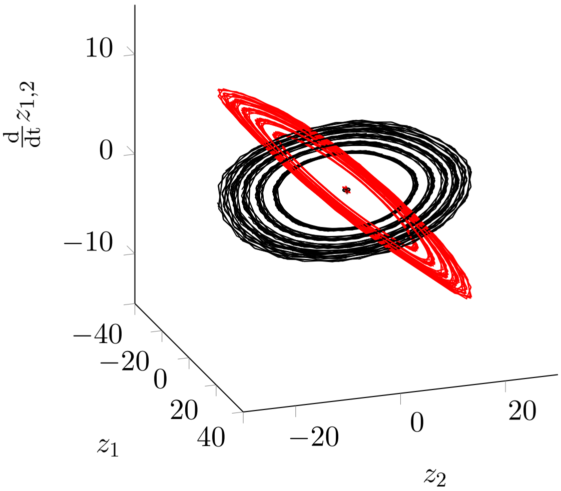

Figure 4.26: Dynamic $g = (g_1, g_2)$ in the intrinsic coordinates (z_1, z_2). The black lines show $\frac{dz_1}{dt} = g_1(\phi_1, \phi_2)$, the red lines $\frac{dz_2}{dt} = g_2(z_1, z_2)$.

The shape of the trajectories in diffusion map space shown in figure 4.26 suggests the hidden model describes rotations on a circle. The state space of the surrogate model is connected to the hidden model by a diffeomorphism (see section 3.2.2), and by transitivity, we can choose any state space diffeomorphic to the state space of the surrogate model, to obtain yet another surrogate model of the hidden one. Here, we choose a state space that allows rotations of angle α on a circle of radius r, where the radius depends on the parameters of the original model. This choice simplifies the surrogate model greatly, as only the initial map and the observer have to be computed numerically. The angle α and radius r are related to the diffusion map coordinates z_1, z_2 by

$$\alpha = \angle (z_1, 0, z_2), \tag{4.8}$$

$$r = \lim_{t \to \infty} \frac{1}{t} \int_0^t \|(\phi_1(t), \phi_2(t))\| dt. \tag{4.9}$$

The norm $\|(z_1(t), z_2(t))\|$ is almost constant over time, so we replace it with the average over the whole trajectory. For different initial conditions $(z_1(0), z_2(0))$, the radii are different, but the angles are the same. The dynamic of the surrogate model on the new state variables α and r can be described by rotations on a circle,

$$\dot{\alpha} = \omega(h, d, v_0),\ \alpha(0) = 0, \tag{4.10}$$

$$\dot{r} = 0,\ r(0) = r_0(h, d, v_0), \tag{4.11}$$

$$\tag{4.12}$$

where ω is the constant angular speed, and r_0 is the constant radius. The functions ω and r_0 map from the system parameters safety distance h, initial distribution d, and speed v_0, to angular speed and radius (see figure 4.27). We compute the angular speed ω as the mean of angular speeds $\frac{d}{dt}\alpha$ on the circle in diffusion map space, by

$$\begin{aligned} \omega &= \lim_{t\to\infty} \tfrac{1}{t} \int_0^t \tfrac{d}{dt}\alpha \\ &\approx \tfrac{1}{T} \sum_{i=1}^{T} \left(\alpha(t_i)) - \alpha(t_{i-1}))\right), \end{aligned} \tag{4.13}$$

where the trajectories $\alpha(t_i)$, $t_i = 0, \dots, t_T = T$ are available through the computation of the closed observables (z_1, z_2) of the surrogate model. The functions $\omega_0(h, d, v_0) = \omega$ and $r_0(h, d, v_0) = r_0$ are shown in figure 4.27. The function values do not change when varying $d > 0$ and keeping h, v_0 constant. After the values of the closed observables (z_1, z_2) are known, they can be transformed back to the density of cars by the observation function shown in figure 4.28.

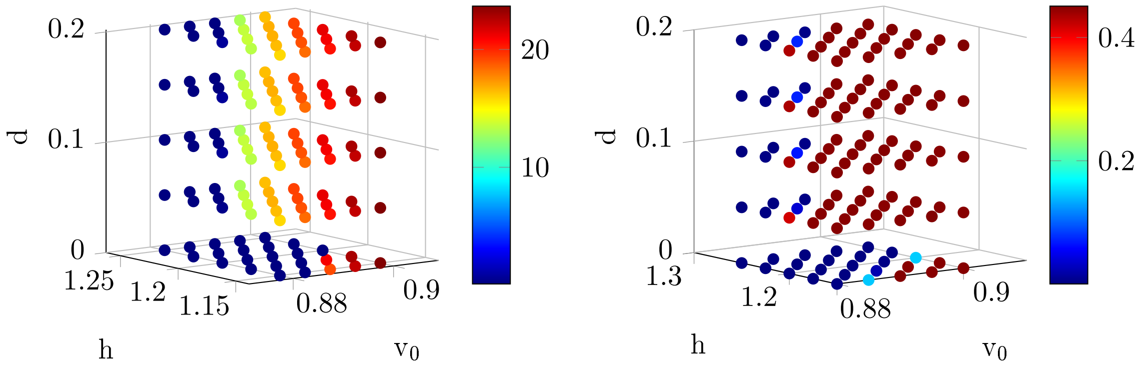

Figure 4.27: The map from the system parameters (h, v_0, d) to the radius r (color, left plot) and angular speed ω (color, right plot). The results ω and r are the parameters for the circular dynamical system forming the surrogate model (equation 4.10 and figure 4.26).

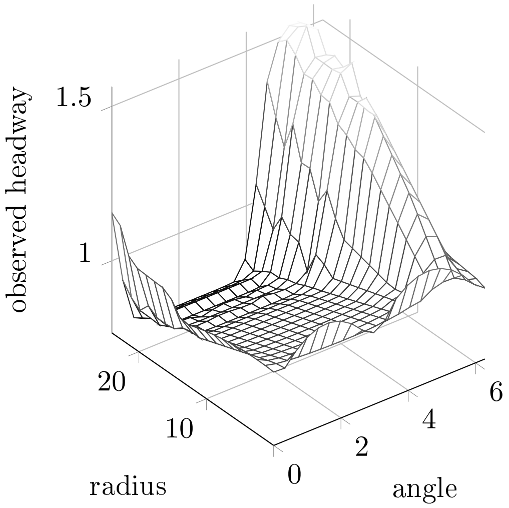

Figure 4.28: Piecewise linear approximation of the observer function $\tilde{y}$, mapping from (α, r) to the headway of cars on the circular road.

The initial map (figure 4.27), the formulaic description of the surrogate model in equation 4.10, and the observer function (figure 4.28), allow to generate the trajectories of the closed observables (z_1, z_2) and then measure the headway of cars on the road over time. Given the angle and the radius, the observer measures the headway as a real number, and not as a distribution function. It is still possible to create the distribution $h(t, x)$ at time t over the entire road of length $[0, L]$ through

$$h(t, x) = \tilde{y}(\alpha(t) + 2\pi x/L, r(t)),\ x \in [0, L],\ t \in \mathbb{R}^+. \tag{4.14}$$

Two headway distributions generated with the surrogate model are compared to the original data in figure 4.29. They agree with a relative error tolerable for many applications in car traffic, where the exact number of cars is often not important. The approximation quality of the surrogate model allows to generate accurate observations very quickly, and also to analyze the model for bifurcations.

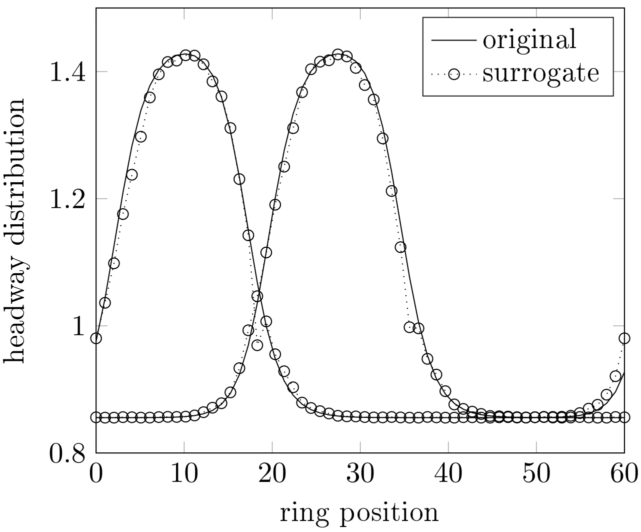

Figure 4.29: Comparison of observations between the original and the surrogate model, for two times t_0 and $t_1 = t_0 + 20$.

Conclusions The stop-and-go behavior of cars on a ring road can be transformed into a much simpler surrogate model with only one point moving around a circle with constant speed and radius. The three system parameters initial density, speed, and safety distance all map to the radius and angular speed, and the rotating point can be mapped back to the headway distributions on the ring road. In this application, we demonstrated how to use a diffeomorphic phase space with a given equation for the dynamic, instead of numerical approximations. The initial map and the observation function are still numerical approximations, and it is not as easy to replace them with a formulaic description, because the exact headway distribution has to be known for all initial parameters. This shows that the intrinsic dynamic of a system can be quite simple, even though the observed values are complex—in the case of road traffic, the observation is a function from space to headways between cars. Compared to state of the art approaches, the surrogate model is able to uncover the topology of the state space on the macroscopic scale, namely, a ring. The bifurcation analysis is now completely decoupled from the dynamics, and can be performed purely on the initial map.

4.3 Granular flow

The term *granular flow* is very generic, and describes the movement of granular particles—which can be, for example, grains of sand, rocks, blood, or molecules. Granular particles are not self-propelled and thus only driven by external forces, which sets them apart from systems with cars, robots, animals, or humans. Differences between individual grains typically make an equation-driven treatment of granular systems challenging. Currently, data-driven and equation-driven methods are successfully applied in combination. One example is running microscopic simulations to compute parameters of equations given on the macroscopic scale. Farmer (2002) reviews these methods in the context of porous media, where both the microscopic and macroscopic system is given as a partial differential equation, and the microscopic system is solved numerically to inform the parameters of the macroscopic PDE. Brandt and Ilyin (2003) discuss the same issue, but with Monte-Carlo simulations on both scales. In general, finding suitable variables for a model on the coarse scale is challenging, because their dynamic has to be independent of the fine scale. In this application, we demonstrate that the construction process of the surrogate model can automatically extract these suitable variables, given only the positions of grains on the fine scale.

General description of the application The particles follow a spring-force model in three dimensions (see (Luding, 2006); the simulation Software was implemented by Christoph Waldleitner (Waldleitner, 2015)). We consider a box-shaped silo with a small opening at the bottom, and dimensions $(\text{length}, \text{width}, \text{height}) = 4 \times 0.7 \times 20$ (see figure 4.30). The small width of the silo forces the grains into an effectively two-dimensional system with a thin third dimension. This facilitates a two-dimensional visualization as in figure 4.31, and still shows nontrivial dynamics.

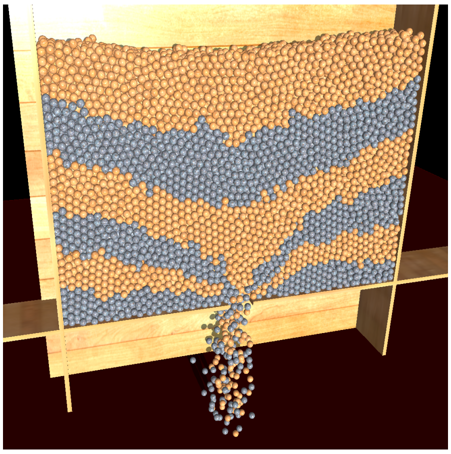

Figure 4.30: Simulation of granular particles in a silo. The colors indicate different initial heights. The simulator and the visualization for this granular flow was implemented by Christoph Waldleitner (Waldleitner, 2015).

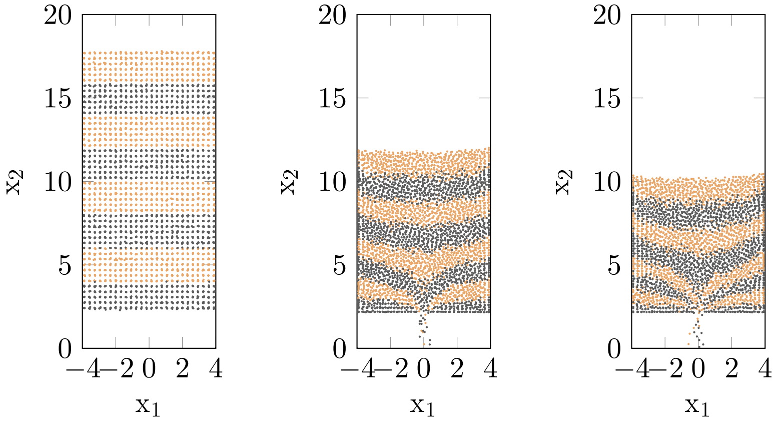

Figure 4.31: Positions of initially 5000 grains in the silo at times $t = 0, 20, 40$. The colors indicate initial positions in the silo for different heights $x_2(0)$.

The dynamics of individual grains is not important in most applications. Primarily, the focus lies on continuous, macroscopic descriptions of the flow and its velocity. Here, we define the macroscopic observable of interest to be the position of large clusters of grains over time. This observable is interesting for a coarser grained view of the system. It can be used to coarse grain a particle system over many spatial and temporal scales, when the coarse graining is applied iteratively on coarser and coarser levels. A cluster is a set of grains with an initial height in a certain range. Figure 4.32 shows an assignment of grains to a cluster at $t = 0$.

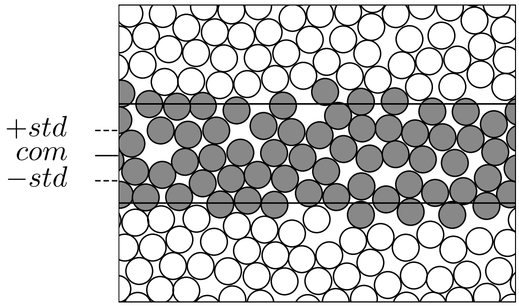

Figure 4.32: Assignment of individual grains to a cluster. All gray grains belong to the cluster in the center, the white grains above and below to two different clusters. The center of mass in the vertical direction is marked *com*, with one standard deviation ($\pm std$) in both directions.

The initial position of the clusters we consider for the construction of the surrogate model are shown as red rectangles in figure 4.33. First, we show how the macroscopic system would be found with conventional methods. Then, we construct a surrogate model and compare the accuracy of its output to a naive interpolation of the complete set of observation data.

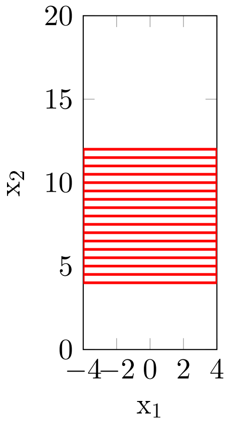

Figure 4.33: Position of the clusters spread over the silo. We construct a surrogate model for the dynamics of grains in each cluster.

State-of-the-art approach without dynamic surrogate models The generality of granular flow makes it an active field of research over many disciplines, both theoretical and applied. We provide references to reviews, and then focus on recent methods that can be used to find the dynamics of each cluster, as outlined in the last section. Similar to crowd flow, there are equation-driven and data-driven methods available. To choose any of the methods, it is important to clarify on which scale the results should be, and on which scale the model for the granular flow is available. Rao and Nott (2008) and Tejchman (2013) introduce the topic of granular flow. The work of Müller (2001) reviews computational approaches, and Bell, Yu, and Mucha (2005) discuss the issue from the perspective of computer graphics. Trujillo and Sigalotti (2014) focus on theoretical approaches, including thermodynamic considerations. Luding (2008) describes the discrete element method, which is widely used to simulate granular matter. Methods for porous media often employ a combination of data-driven methods on the microscopic scale, and equation-driven methods on the macroscopic scale. The spot model (Bazant, 2006) is a computational approach to upscaling in granular flow. Equation-free, data-driven approaches are also available (Moon, Sundaresan, and Kevrekidis, 2007).

In all methods cited above, finding good observables of the macroscopic process is crucial. With the surrogate model, we will not assume any specific macroscopic observable, but use all positions of individual grains as observables to compute closed observables. In this challenging setting, the construction generates slowly changing, intrinsic variables for the surrogate model. The variables allow to observe both the vertical position of the center of mass in a cluster, and the standard deviation of the vertical positions of grains, over time.

Alternative approach with the dynamic surrogate model The data-driven construction of the surrogate model allows to generate models for individual slices of the silo. We assume no knowledge about the correct observables on the macroscopic scale, and observe the vertical position of all grains in a slice at once. This allows the construction process of the surrogate model to automatically determine “optimal” intrinsic variables. With these variables, all functions of the vertical positions of all grains can be measured. We choose the center of mass in a cluster, as well as the standard deviation in the vertical direction over all grains in the cluster, which allows to quantify the cluster shape and position over time. Here, we demonstrate that knowing the center of mass and the standard deviation is not enough to advance these two quantities in time. However, when constructing the surrogate model, time lags of the positions of all grains in a cluster are considered. Together with the diffusion map, this generates intrinsic variables that hold enough information to advance in time, and still allow to observe the center of mass and standard deviation of the vertical positions of grains in a cluster. The initial mean height of the grains in a cluster is the system parameter we vary, so we can generate the evolution of all grains over time when simulating all clusters.

To generate data, we simulate all grains in the silo microscopically. In an initial phase of 15 seconds, the grains settle from their initial positions in a rectangular grid to their positions in the flowing phase. After this initial phase, we capture the positions of 50 randomly chosen grains in each cluster, for 50 consecutive time steps of 2 seconds each. As grains can move with different speeds relative to the center of mass in a cluster, the shape of the cluster changes. We compute a surrogate model for the vertical movement of all grains relative to the starting position of each cluster. The relative movement, in contrast to an absolute movement in the silo, is chosen to demonstrate that the construction of the surrogate can differentiate the clusters based on the dynamics on the grains. If the absolute movement were considered, a differentiation would be possible directly by the vertical position. An observation $y_j(t_k)$ of the vertical position of particle j at time t_k relative to the starting position of the cluster is computed from all vertical positions $v_{1,\ldots,50}(t_k) = (v_1(t_k), v_2(t_k), \ldots, v_{50}(t_k))$ in the silo through

$$y_j(t_k) = y_j(v_{1,\ldots,50}(t_k)) = v_j(t_k) - \frac{1}{50}\sum_{m=1}^{50} v_m(0),\ j \in \{1, \ldots, 50\}. \tag{4.15}$$

Finding a model for the values y_j poses a challenge to methods of the state of the art, because all trajectories start at approximately the same vertical position zero (see figure 4.34). If only the position relative to the cluster is known, and not the cluster number or past positions, it is impossible to compute future positions, as all positions are equal at $t = 0$ and only change afterwards.

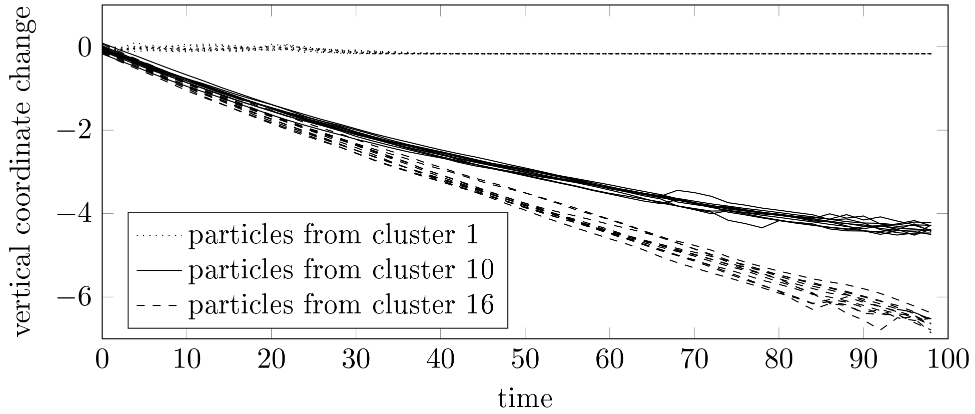

Figure 4.34: Part of the data used to construct the surrogate model. Ten particle trajectories per cluster are shown, whereas 50 particles in each of the 16 clusters were used in the construction of the surrogate model.

After delay embedding and construction of the diffusion map, the first two diffusion eigenfunctions are independent, while the other computed eigenfunctions are harmonics of the first two. The existence of two independent eigenfunctions implies the intrinsic dimension of the surrogate model is two, which indicates the relative particle positions are not enough to predict future values. There are at least two reasons for this. First, all positions are relative to the cluster. If, at the beginning of the simulation, only the relative positions are known and not the cluster position in the silo, it is not possible to predict future values. Second, a more abstract reason is that Hamiltonian systems are defined on the position *and* the velocity of particles. Figures 4.35 and 4.36 show simulation results obtained from the surrogate model, compared to the observations from the original, microscopic model. For the values after 60 seconds, the surrogate model is used to predict future values. The cluster positions are approximated with a low error, whereas the standard deviation has a larger relative error. This is due to the higher order computations necessary to evaluate the standard deviation in comparison to the mean.

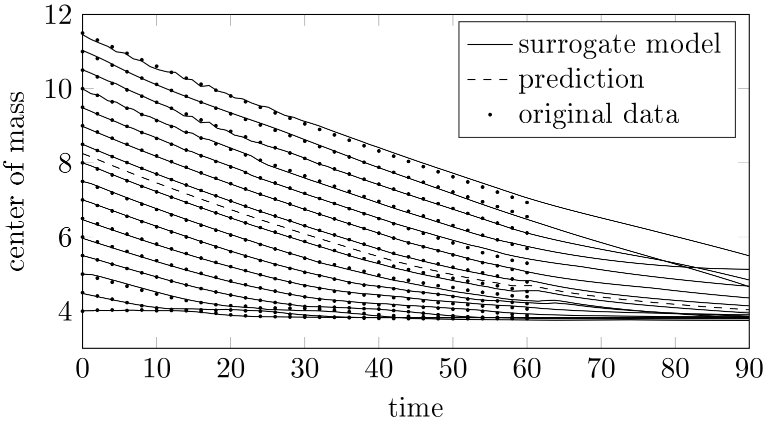

Figure 4.35: Simulation results of the surrogate model, compared to original data. We do not measure all particle positions, but the mean position of the center of mass over all particles in a cluster. A forecast of the mean value is possible for about 25 seconds, before the accuracy decreases at $t > 80$. A prediction of a trajectory for a parameter not in the original data is correctly interpolated through the surrogate (dashed line).

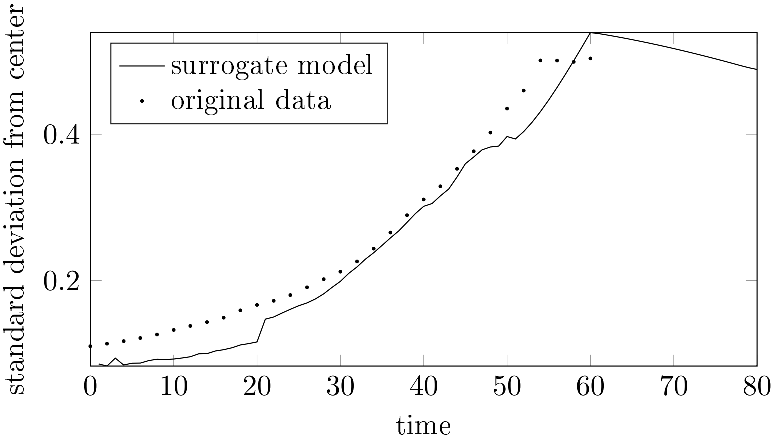

Figure 4.36: Simulation results of the surrogate model, compared to original data, for the standard deviation in the horizontal direction from the center of mass in a cluster. We only show one trajectory out of 16, as the standard deviation varies too much between clusters. The standard deviation is not approximated very accurately, and the forecast is also incorrect, due to the high noise level in the data.

Figure 4.37 shows a standard approach to surrogate modeling, as a comparison to the dynamic surrogate model. A response surface is computed for all data points, with the time treated as an additional parameter. The interpolation method is the same as for the surrogate model functions, a piecewise linear function, and we also extrapolate the data linearly. Different to the dynamic surrogate (figure 4.35), the response surface can reproduce the observation data in the given range of values exactly. This is not surprising for a piecewise linear interpolation method. Comparing prediction quality, the surrogate is able to capture the decrease to the height value of four more accurately than the response surface. This is because the dynamic surrogate is interpolating, and extrapolating, the intrinsic dynamic of the hidden model, while the response surface only considers the dynamic of the last two observations in the linear extrapolation.

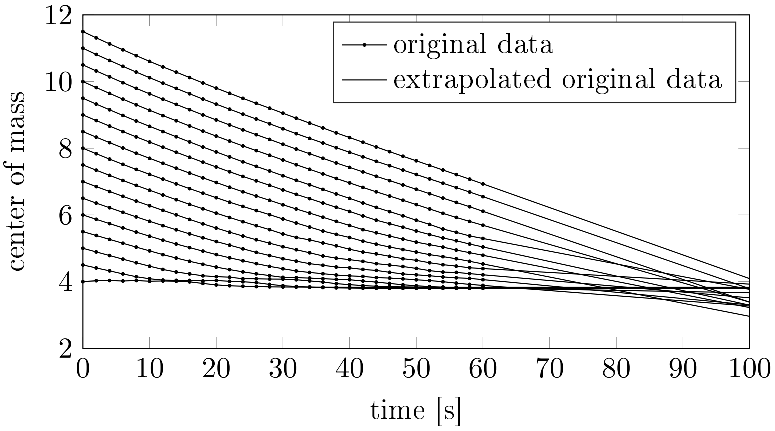

Figure 4.37: Evolution of the center of mass for all clusters, shown as black dots. The system parameter is the initial position of the observed cluster, varied from 4.0 to 11.5. A linear extrapolation from the original data is shown as black lines, by extending the data surface further in time. The prediction results after $60s$ are not as accurate as with the surrogate model, because the intrinsic dynamic of the process is not approximated.

Conclusions Granular flow is a very diverse field with many different goals and applications. We focused on a silo example, and showed how the dynamic surrogate approach can automatically identify suitable observables, given only the vertical positions of all grains in small clusters. The surrogate model can accurately approximate macroscopic observations, such as the average vertical position of grains in a cluster, and is also able to predict future values through the approximation of the dynamic intrinsic to the process. The surrogate performs better than a direct extrapolation from the macroscopic data, which would be used conventionally in equation-free methods. The presence of high-frequency changes in the center of mass in a cluster due to individual grains poses a severe challenge to the current construction process of the surrogate model, as discussed in the section on stochastic effects (section 3.2.7). Here, we used averaging over several simulation runs to smooth out these high frequencies. Another way to resolve the underlying macroscopic dynamic would be to combine more grains into one cluster.

4.4 Summary of the chapter on applications

We demonstrated the construction process, benefits, and possible caveats of the data-driven surrogate model on five example applications: crowd dynamics with queuing at the entrance of a music festival and density evolution through a bottleneck, car traffic with stop-and-go waves on a highway, the flow of granular particles in a silo, and finally, uncertainty quantification in a train station. We briefly outlined how the challenge of each example would be solved with conventional methods, and then detailed how to address potential research gaps using the newly introduced concept of dynamic, data-driven surrogate models. We concluded each of the five applications with a summary and potential future directions of research.

The queuing process at the entrance of the music festival could be reduced to a two-dimensional surrogate model. It was possible to observe the queue length over time, for the inflow and service times at the control point as parameters. We showed how the surrogate model enables to find critical bifurcation points where the queue length exceeds the system capacity, and discussed how the analysis of these critical points can help crowd control. The crowd density in front of the bottleneck was used as an example where the observations are functions. We reduced the dimension of the observation space through an additional linear subspace decomposition with the method of snapshots, and then showed how to construct a non-linear, data-driven surrogate model for the coefficients of the linear decomposition. Considering stop-and-go waves in car traffic, we showed how to automatically reduce a non-linear system of ordinary differential equations to a linear, circular system by constructing a surrogate model. The points on the circle comprised the intrinsic variables of the surrogate model, and could be measured to obtain the car density on the highway, showing stop-and-go behavior. The example of grains in a silo demonstrated the construction of a surrogate model in granular flow, where domain decomposition is commonly used to simulate a large number of particles. We constructed a surrogate model for several parts of the silo, and then linked the parts through the construction of a dynamic map depending on surrounding clusters of the current cluster. As a last application, we employed a data-driven surrogate model for uncertainty quantification of the deboarding process of a train. The number of passengers deboarding the train and the number of waiting passengers on the platform were used as parameters for the system. We showed that by using the surrogate model, many different probability distributions can be tested very quickly. This showed that real-time prediction and control of uncertainty, even for complex systems, is possible with the data-driven surrogate model.

Chapter 5

Summary, future directions, and conclusion

This final chapter is divided into a summary, possible future directions, and a concluding section. Section 5.1 is the summary of the thesis, with a short review of each of the three main chapters. Section 5.2 contains future research directions, expanding on the concept of data-driven surrogate models. The last section concludes the thesis.

5.1 Summary

How can a scale transition from microscopic to macroscopic models be achieved through a data-driven procedure, when the macroscopic equation is not known, but is assumed to exist? To answer this question, we combined results from dynamical system theory and manifold learning with simulations of state-of-the-art microscopic models from crowd dynamics, car traffic, and granular flow. The combination resulted in a surrogate model approach to upscaling, where the microscopic model generates the data needed to learn the macroscopic model, performing a scale transition through focusing on observables that change the slowest. Chapter 2 contains a concise description of the framework of multiscale dynamical systems, integrating crowd dynamics, car traffic, and systems of granular matter. This bridges a gap between problems in safety engineering and the mathematical sciences for dynamical systems and machine learning. In Chapter 3, we defined the concept of a *dynamic, data-driven surrogate model* on a manifold in time-delay embedded observation space. With this surrogate model, it is possible to generate observations of a microscopic system over time on a macroscopic scale. We provided details on the algorithmic construction and simulation process. We also proved three theorems, regarding approximation in the limit of infinite data, approximation quality with finite data, and storage efficiency. Five applications of the surrogate model concept were discussed in Chapter 4. We constructed surrogate models for a bottleneck scenario, queuing, and uncertainty quantification of crowd dynamics, for car traffic on a highway, and for the flow of granular particles in a silo. For all examples, we discussed benefits and caveats of using the data-driven surrogate model compared to state of the art approaches.

5.2 Future directions

The construction of a dynamic and data-driven surrogate model offers many possibilities for future work. The models constructed through time-delay embedding and diffusion maps typically have a low intrinsic dimension, but a nonlinear dynamic. In the Koopman operator framework, this view is reversed, offering linear dynamics in an infinite dimensional system. There is a third option related to these two views of dynamical systems. It is to find low-dimensional systems with a linear dynamic, that can be observed through an observation function independent of time, to yield observations of the original system. The trivial example of a linear system is the linear advance of time itself, $\frac{d}{dt}t = 1$. For any original system, the observation function that transforms time back to the original system is the *solution* of the original system. A more advanced example is the Cole-Hopf transform between the nonlinear, viscous Burgers equation and the linear diffusion equation. In both examples, finding the transformation function between the linear and the nonlinear system is very informative about the dynamical behavior. Finding the transformation with a data-driven procedure would enable to simplify and categorize nonlinear systems without the need for explicit formulaic descriptions.

The surrogate modeling idea can also be used to construct systems with more than one scale. In network optimization, fine scale simulations informing surrogate models on the edges of a network would allow a macroscopic view on the network, because behavior on individual edges can be simulated very efficiently while still capturing microscopic features. A graph combining several surrogate models is closely related to the idea of systematic upscaling, which additionally incorporates models on several scales. Both spatially distributed processes and hierarchical structures could be approximated by the dynamic surrogate models. Adaptive methods can utilize the dynamic information stored in the surrogate to sample more efficiently. Using the surrogate as the macroscopic model in equation-free computations would allow the storage of intermediate results and to rerun the system on the macroscopic scale for offline analysis. The surrogate model could even aid real-time uncertainty quantification, where the output for many different input distributions of multiple parameters can be tested real-time and on-site. This is very challenging with a computationally slow simulation model.

To unlock the full potential of surrogate models with closed observables, the concept must be integrated in simulation frameworks able to run scenarios. This is not only a serious software challenge, but also many problems that the thesis has treated separately may occur simultaneously. When enough experimental data is available, one could also construct numerical models directly from the observations, and in this way extract macroscopic dynamics from experiments.

5.3 Conclusion

Models of dynamical processes are ubiquitous in science and engineering. Many of these processes show different behavior on different temporal and spatial scales, which leads to several models for each scale, and to models incorporating several scales. Given any model of a process, being able to transition between the scales is important to build models on coarser and coarser scales. We introduced a surrogate model, based on time-delay embedding and observation of states on a macroscopic scale. The model variables are closed under the flow; hence the surrogate can be used without the original microscopic model.

The construction and simulation process for the dynamic surrogate model are described in detail. Numerical approximation error, storage capacity improvements, treatment of infinite-dimensional observables and parameter spaces, interpolation schemes, systems with more than two scales, as well as stochastic effects are discussed with regard to the surrogate modeling approach.

Five applications show the diverse use-cases of the dynamic surrogate modeling approach. Uncertainty quantification can now be split into a construction phase, where the numerical model on the scale of interest is constructed with the fine scale simulator, and an analysis phase, where the high performance of the numerical model can be used to perform quantification of uncertainties without running the original simulator.

Generally, the dynamic surrogate modeling approach allows to separate the dynamic process from observations. The separation provides advantages for mathematical modeling and simulation in general, because it untangles the information needed to advance in time from the information needed as an output of the process, the observations. The thought that a given, complex observation might be generated by a simple system is intriguing, and might even hold for chaotic systems. This untangling is possible for a wide range of systems with multiple scales, by the numerical construction of the surrogate model discussed in the thesis.

Bibliography

Ainsworth, Mark and J.Tinsley Oden (1997). "A posteriori error estimation in finite element analysis". In: *Computer Methods in Applied Mechanics and Engineering* 142.1-2, pp. 1–88. ISSN: 0045-7825. DOI: 10.1016/s0045-7825(96)01107-3.

Aoki, Ichiro (1982). "A simulation study on the schooling mechanism in fish". In: *Nippon Suisan Gakkaishi* 48.8, pp. 1081–1088. ISSN: 0021-5392. DOI: 10.2331/suisan.48.1081.

Araujo, Vitor, Stefano Galatolo, and Maria José Pacifico (2014). "Statistical Properties of Lorenz-like Flows, Recent Developments and Perspectives". In: *International Journal of Bifurcation and Chaos* 24.10, p. 1430028. ISSN: 1793-6551. DOI: 10.1142/s0218127414300286.

Bando, M. et al. (1995). "Dynamical model of traffic congestion and numerical simulation". In: *Physical Review E* 51 (2), pp. 1035–1042. DOI: 10.1103/PhysRevE.51.1035.

Barrault, Maxime et al. (2004). "An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations". In: *Comptes Rendus Mathematique* 339.9, pp. 667–672. ISSN: 1631-073X. DOI: 10.1016/j.crma.2004.08.006.

Bazant, Martin Z. (2006). "The Spot Model for random-packing dynamics". In: *Mechanics of Materials* 38.8-10, pp. 717–731. ISSN: 0167-6636. DOI: 10.1016/j.mechmat.2005.06.016.

Bell, Nathan, Yizhou Yu, and Peter J. Mucha (2005). "Particle-based Simulation of Granular Materials". In: *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* SCA '05. New York, NY, USA: ACM, pp. 77–86. DOI: 10.1145/1073368.1073379.

Bellman, Richard (1957). *Dynamic Programming.* Princeton University Press.

Bellman, Richard Ernest (2003). *Dynamic Programming.* illustrated, reprint. Courier Corporation.

Bellomo, Nicola and Abdelghani Bellouquid (2015). "On multiscale models of pedestrian crowds from mesoscopic to macroscopic". In: *Communications in Mathematical Sciences* 13.7, pp. 1649–1664. ISSN: 1945-0796. DOI: 10.4310/cms.2015.v13.n7.a1.

Bellomo, Nicola and Christian Dogbe (2011). "On the Modeling of Traffic and Crowds: A Survey of Models, Speculations, and Perspectives". In: *SIAM Review* 53.3, pp. 409–463. DOI: 10.1137/090746677.

Bellomo, Nicola, Benedetto Piccoli, and Andrea Tosin (2012). "Modeling crowd dynamics from a complex system viewpoint". In: *Mathematical Models and Methods in Applied Sciences* 22.supp02, pp. 1230004–1–1230004–29. DOI: 10.1142/S0218202512300049. eprint: http://www.worldscientific.com/doi/pdf/10.1142/S0218202512300049. URL: http://www.worldscientific.com/doi/abs/10.1142/S0218202512300049.

Bengio, Yoshua et al. (2004). "Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering". In: *Advances in Neural Information Processing Systems 16*. MIT Press. URL: http://research.microsoft.com/apps/pubs/default.aspx?id=64655.

Berry, T. et al. (2013). "Time-Scale Separation from Diffusion-Mapped Delay Coordinates". In: *SIAM Journal on Applied Dynamical Systems* 12.2, pp. 618–649. DOI: 10.1137/12088183x.

Berry, Tyrus and John Harlim (2016). "Variable bandwidth diffusion kernels". In: *Applied and Computational Harmonic Analysis* 40.1, pp. 68–96. ISSN: 1063-5203. DOI: 10.1016/j.acha.2015.01.001.

Berry, Tyrus and Timothy Sauer (2015). "Local kernels and the geometric structure of data". In: *Applied and Computational Harmonic Analysis*. ISSN: 1063-5203. DOI: 10.1016/j.acha.2015.03.002.

Biedermann, Daniel H. et al. (2014). "Towards TransiTUM: A Generic Framework for Multiscale Coupling of Pedestrian Simulation Models based on Transition Zones". In: *Transportation Research Procedia* 2, pp. 495–500. ISSN: 2352-1465. DOI: 10.1016/j.trpro.2014.09.065.

Boltes, Maik and Armin Seyfried (2013). "Collecting pedestrian trajectories". In: *Neurocomputing* 100. Special issue: Behaviours in video, pp. 127–133. DOI: 10.1016/j.neucom.2012.01.036.

Brandt, A. and V. Ilyin (2003). "Multilevel Monte Carlo methods for studying large scale phenomena in fluids". In: *Journal of Molecular Liquids* 105.2-3, pp. 245–248. DOI: 10.1016/S0167-7322(03)00061-8.

Brandt, A. et al. (2011). "Bootstrap AMG". In: *SIAM Journal on Scientific Computing* 33, pp. 612–632. ISSN: 1095-7197. DOI: 10.1137/090752973.

Brandt, Achi (2005). "Multiscale solvers and systematic upscaling in computational physics". In: *Computer Physics Communications* 169.1-3, pp. 438–441. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2005.03.097.

Brindley, J, C Kaas-Petersen, and A Spence (1989). "Path-following methods in bifurcation problems". In: *Physica D: Nonlinear Phenomena* 34.3, pp. 456–461.

Brockwell, Peter J. and Richard A. Davis (1991). *Time Series: Theory and Methods*. 2nd edition. Springer.

Brŝĉić, Draẑen, Francesco Zanlungo, and Takayuki Kanda (2014). "Density and Velocity Patterns during One Year of Pedestrian Tracking". In: *Pedestrian and Evacuation Dynamics 2014*. Ed. by Winnie Daamen, Dorine C. Duives, and Serge P. Hoogendoorn. Vol. 2. Elsevier BV, pp. 77–86. DOI: 10.1016/j.trpro.2014.09.011.

Brunton, Steven L. et al. (2016). "Chaos as an Intermittently Forced Linear System". In: *arXiv*. arXiv: 1608.05306v1 [math.DS].

Budišić, Marko, Ryan Mohr, and Igor Mezić (2012). "Applied Koopmanism". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 22.4, p. 047510. ISSN: 1089-7682. DOI: 10.1063/1.4772195.

Bungartz, Hans-Joachim and Michael Griebel (2004). "Sparse grids". In: *Acta Numerica* 13, pp. 147–269. DOI: 10.1017/s0962492904000182.

Bungartz, Hans-Joachim et al. (2014). *Modeling and Simulation: An Application-Oriented Introduction.* Springer Undergraduate Texts in Mathematics and Technology. Berlin Heidelberg: Springer. DOI: 10.1007/978-3-642-39524-6.

Burstedde, C. et al. (2001). "Simulation of pedestrian dynamics using a two-dimensional cellular automaton". In: *Physica A: Statistical Mechanics and its Applications* 295, pp. 507–525. DOI: 10.1016/S0378-4371(01)00141-8.

Car, R. and M. Parrinello (1985). "Unified Approach for Molecular Dynamics and Density-Functional Theory". In: *Physical Review Letters* 55.22, pp. 2471–2474. ISSN: 0031-9007. DOI: 10.1103/physrevlett.55.2471.

Carlberg, Kevin et al. (2013). "The {GNAT} method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows". In: *Journal of Computational Physics* 242.0, pp. 623–647. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2013.02.028.

Carrillo, J. A., S. Martin, and V. Panferov (2013). "A new interaction potential for swarming models". In: *Physica D: Nonlinear Phenomena* 260.0, pp. 112–126. DOI: 10.1016/j.physd.2013.02.004.

Chaturantabut, Saifon and Danny C. Sorensen (2010). "Nonlinear Model Reduction via Discrete Empirical Interpolation". In: *SIAM Journal on Scientific Computing* 32.5, pp. 2737–2764. ISSN: 1095-7197. DOI: 10.1137/090766498.

Chorin, Alexandre J., Anton P. Kast, and Raz Kupferman (1998). "Optimal prediction of underresolved dynamics". In: *Proceedings of the National Academy of Sciences* 95.8, pp. 4094–4098. DOI: 10.1073/pnas.95.8.4094. eprint: http://www.pnas.org/content/95/8/4094.full.pdf+html.

Chraibi, Mohcine (2012). "Validated force-based modeling of pedestrian dynamics". PhD thesis. Universität zu Köln.

Chraibi, Mohcine et al. (2011). "Force-based models of pedestrian dynamics". In: *Networks and Heterogeneous Media* 6.3, pp. 425–442. DOI: 10.3934/nhm.2011.6.425.

Coifman, R. R. et al. (2008). "Diffusion Maps, Reduction Coordinates, and Low Dimensional Representation of Stochastic Systems". In: *Multiscale Modeling & Simulation* 7.2, pp. 842–864. ISSN: 1540-3467. DOI: 10.1137/070696325.

Coifman, Ronald R. and Stéphane Lafon (2006). "Diffusion maps". In: *Applied and Computational Harmonic Analysis* 21.1, pp. 5–30. ISSN: 1063-5203. DOI: 10.1016/j.acha.2006.04.006.

Coifman, Ronald R. et al. (2005). "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps". In: *Proceedings of the National Academy of Sciences of the United States of America* 102.21, pp. 7426–7431. DOI: 10.1073/pnas.0500334102. eprint: http://www.pnas.org/content/102/21/7426.full.pdf+html.

Cristiani, Emiliano, Benedetto Piccoli, and Andrea Tosin (2011). “Multiscale Modeling of Granular Flows with Application to Crowd Dynamics”. In: *Multiscale Modeling & Simulation* 9.1, pp. 155–182. ISSN: 1540-3467. DOI: 10.1137/100797515.

— (2014). *Multiscale Modeling of Pedestrian Dynamics*. Springer International Publishing. DOI: 10.1007/978-3-319-06620-2.

Czirók, András, Albert-László Barabási, and Tamás Vicsek (1999). “Collective Motion of Self-Propelled Particles: Kinetic Phase Transition in One Dimension”. In: *Physical Review Letters* 82 (1), pp. 209–212. DOI: 10.1103/PhysRevLett.82.209.

Daubechies, Ingrid (1992). “Ten Lectures on Wavelets”. In: *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial & Applied Mathematics (SIAM). ISBN: http://id.crossref.org/isbn/978-1-61197-010-4. DOI: 10.1137/1.9781611970104.

Davidich, Maria and Gerta Köster (2013). “Predicting Pedestrian Flow: A Methodology and a Proof of Concept Based on Real-Life Data”. In: *PLoS ONE* 8.12, pp. 1–11. DOI: 10.1371/journal.pone.0083355.

Degond, P. et al. (2013). “A Hierarchy of Heuristic-Based Models of Crowd Dynamics”. In: *Journal of Statistical Physics* 152.6, pp. 1033–1068. DOI: 10.1007/s10955-013-0805-x.

Deyle, Ethan R. and George Sugihara (2011). “Generalized Theorems for Nonlinear State Space Reconstruction”. In: *PLoS ONE* 6.3. Ed. by Matej Oresic, e18295. DOI: 10.1371/journal.pone.0018295.

Dietrich, Felix, Florian Albrecht, and Gerta Köster (2016). “Surrogate Models for Bottleneck Scenarios”. In: *Proceedings of the 8th International Conference on Pedestrian and Evacuation Dynamics (PED2016)*. Hefei, China.

Dietrich, Felix and Gerta Köster (2014). “Gradient navigation model for pedestrian dynamics”. In: *Physical Review E* 89.6, p. 062801. DOI: 10.1103/PhysRevE.89.062801.

Dietrich, Felix, Gerta Köster, and Hans-Joachim Bungartz (2016). “Numerical Model Construction with Closed Observables”. In: *SIAM Journal on Applied Dynamical Systems* 15.4, pp. 2078–2108. DOI: 10.1137/15M1043613.

Dietrich, Felix et al. (2014). “Bridging the gap: From cellular automata to differential equation models for pedestrian dynamics”. In: *Journal of Computational Science* 5.5, pp. 841–846. DOI: 10.1016/j.jocs.2014.06.005.

Drury, John and Steve Reicher (2010). “Collective action and psychological change: The emergence of new social identities”. In: *British Journal of Social Psychology* 39.4, pp. 579–604. DOI: 10.1348/014466600164642.

Duives, Dorine C., Winnie Daamen, and Serge P. Hoogendoorn (2013). “State-of-the-art crowd motion simulation models”. In: *Transportation Research Part C: Emerging Technologies* 37.0, pp. 193–209. DOI: 10.1016/j.trc.2013.02.005.

E, Weinan (2011). *Principles of Multiscale Modeling*. Cambridge: Cambridge University Press.

E, Weinan and Björn Engquist (2003). “The Heterogeneous Multiscale Methods”. In: *Communications in Mathematical Sciences* 1.1, pp. 87–132.

E, Weinan et al. (2007). “Heterogeneous Multiscale Methods: A Review”. In: *Communications in Computational Physics*.

Eells, James and Domingo Toledo, eds. (1992). *Hassler Whitney Collected Papers*. Contemporary Mathematicians. Birkhäuser Boston. ISBN: http://id.crossref.org/isbn/978-1-4612-2972-8. DOI: 10.1007/978-1-4612-2972-8. URL: http://dx.doi.org/10.1007/978-1-4612-2972-8.

Erlang, Agner Krarup (1909). “The theory of probabilities and telephone conversations”. In: *Nyt Tidsskrift for Matematik B* 20, pp. 33–39.

Ethier, Stewart N. and Thomas G. Kurtz (1986). *Stochastic Processes and Martingales*. Wiley-Blackwell. DOI: 10.1002/9780470316658.ch2.

Farmer, C. L. (2002). “Upscaling: a review”. In: *International Journal for Numerical Methods in Fluids* 40.1-2, pp. 63–78. ISSN: 1097-0363. DOI: 10.1002/fld.267.

Fenichel, Neil (1972). “Persistence and Smoothness of Invariant Manifolds for Flows”. In: *Indiana. Univ. Math. J.* 21.3, pp. 193 –226.

— (1979). “Geometric singular perturbation theory for ordinary differential equations”. In: *Journal of Differential Equations* 31.1, pp. 53–98. ISSN: 0022-0396. DOI: 10.1016/0022-0396(79)90152-9.

Francescoa, Marco Di et al. (2011). “On the Hughes’ model for pedestrian ow: The one-dimensional case”. In: *Journal of Differential Equations* 250.3, pp. 1334–1362. DOI: 10.1016/j.jde.2010.10.015.

Frigg, Roman and Stephan Hartmann (2012). “Models in Science”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Fall 2012. Metaphysics Research Lab, Center for the Study of Language and Information, Stanford University. URL: http://plato.stanford.edu/archives/fall2012/entries/models-science/.

Garcia, Alejandro L et al. (1999). “Adaptive Mesh and Algorithm Refinement Using Direct Simulation Monte Carlo”. In: *Journal of Computational Physics* 154.1, pp. 134–155. ISSN: 0021-9991. DOI: 10.1006/jcph.1999.6305.

Gear, C.W., Ioannis G. Kevrekidis, and Constantinos Theodoropoulos (2002). “’Coarse’ integration/bifurcation analysis via microscopic simulators: micro-Galerkin methods”. In: *Computers & Chemical Engineering* 26.7–8, pp. 941–963. ISSN: 0098–1354. DOI: http://dx.doi.org/10.1016/S0098-1354(02)00020-0.

Giannakis, Dimitrios (2015). “Data-driven spectral decomposition and forecasting of ergodic dynamical systems”. In: *arXiv* 1507.02338.v1. URL: http://arxiv.org/abs/1507.02338.

Giannakis, Dimitrios, J. Slawinska, and Zhizhen Zhao (2015). “Spatiotemporal feature extraction with data-driven Koopman operators”. In: *J. Mach. Learn. Res. Proceedings* 44, pp. 103–115.

Givon, Dror, Raz Kupferman, and Andrew Stuart (2004). “Extracting macroscopic dynamics: model problems and algorithms”. In: *Nonlinearity* 17.6, pp. 55–127. DOI: 10.1088/0951-7715/17/6/R01.

Golub, G. H. and C. Reinsch (1970). “Singular value decomposition and least squares solutions”. In: *Numerische Mathematik* 14.5, pp. 403–420. ISSN: 0945-3245. DOI: 10.1007/bf02163027.

Golub, G.H. and C.F. Van Loan (1996). *Matrix Computations.* Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press. ISBN: 9780801854149.

Greengard, L. and V. Rokhlin (1997). "A Fast Algorithm for Particle Simulations". In: *Journal of Computational Physics* 135.2, pp. 280–292. ISSN: 0021-9991. DOI: 10.1006/jcph.1997.5706.

Guckenheimer, John and Philip Holmes (1986). *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields.* Springer.

Helbing, Dirk (2001). "Traffic and related self-driven many-particle systems". In: *Review of Modern Physics* 73.4, pp. 1067–1141. ISSN: 1539-0756. DOI: 10.1103/revmodphys.73.1067.

Helbing, Dirk, Illés Farkas, and Tamás Vicsek (2000). "Simulating dynamical features of escape panic". In: *Nature* 407, pp. 487–490. DOI: 10.1038/35035023.

Helbing, Dirk, Joachim Keltsch, and Péter Molnár (1997). "Modelling the evolution of human trail systems". In: *Nature* 388, pp. 47–50. DOI: 10.1038/40353.

Helbing, Dirk and Péter Molnár (1995). "Social Force Model for pedestrian dynamics". In: *Physical Review E* 51.5, pp. 4282–4286. DOI: 10.1103/PhysRevE.51.4282.

Helbing, Dirk et al. (2001). "Self-organizing pedestrian movement". In: *Environment and Planning B: Planning and Design* 28, pp. 361–383. DOI: 10.1068/b2697.

Helbing, Dirk et al. (2006). "Analytical Approach to Continuous and Intermittent Bottleneck Flows". In: *Physical Review Letters* 97.16, p. 168001. ISSN: 1079-7114. DOI: 10.1103/physrevlett.97.168001.

Herbert-Read, J. E. (2016). "Understanding how animal groups achieve coordinated movement". In: *Journal of Experimental Biology* 219.19, pp. 2971–2983. DOI: 10.1242/jeb.129411.

Hestenes, M.R. and E. Stiefel (1952). "Methods of conjugate gradients for solving linear systems". In: *Journal of Research of the National Bureau of Standards* 49.6, p. 409. ISSN: 0091-0635. DOI: 10.6028/jres.049.044.

Hofmann, Thomas, Bernhard Schölkopf, and Alexander J. Smola (2008). "Kernel methods in machine learning". In: *The Annals of Statistics* 36.3, pp. 1171–1220. DOI: 10.1214/009053607000000677.

Hotelling, Harold (1936). "Simplified calculation of principal components". In: *Psychometrika* 1.1, pp. 27–35. ISSN: 1860-0980. DOI: 10.1007/bf02287921.

Hughes, Roger L. (2001). "A continuum theory for the flow of pedestrians". In: *Transportation Research Part B: Methodological* 36.6, pp. 507 –535. ISSN: 0191-2615. DOI: 10.1016/S0191-2615(01)00015-7.

Huke, J P and D S Broomhead (2007). "Embedding theorems for non-uniformly sampled dynamical systems". In: *Nonlinearity* 20.9, pp. 2205–2244. ISSN: 1361-6544. DOI: 10.1088/0951-7715/20/9/011.

Iaccarino, Gianluca (2008). "Quantification of Uncertainty in Flow Simulations Using Probabilistic Methods". In: *VKI Lecture Series.* Sept. 8 – 12.

Isham, Chris J. (2002). *Modern Differential Geometry for Physicists.* Allied Publishers Pvt. Ltd. ISBN: 9788177643169.

Karamouzas, Ioannis, Brian Skinner, and Stephen J. Guy (2014). "Universal Power Law Governing Pedestrian Interactions". In: *Physical Review Letters* 113.23, p. 238701. DOI: 10.1103/PhysRevLett.113.238701.

Karhunen, Kari (1946). "Zur Spektraltheorie stochastischer Prozesse". In: *Ann. Acad. Sci. Fennicae. Ser. A. I. Math.-Phys.* 34.7.

Kendall, David G. (1953). "Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain". In: *The Annals of Mathematical Statistics* 24.3, pp. 338–354. ISSN: 0003-4851. DOI: 10.1214/aoms/1177728975.

Kevorkian, J.K. and J.D. Cole (1996). *Multiple Scale and Singular Perturbation Methods.* Vol. 114. Applied Mathematical Sciences. Springer. DOI: 10.1007/978-1-4612-3968-0.

Kevrekidis, I. G. et al. (2003). "Equation-free, coarse-grained multiscale computation: enabling microscopic simulators to perform system-level analysis". In: *Commun. Math. Sci.* 1.4, pp. 715–762.

Kevrekidis, Ioannis G. and Giovanni Samaey (2009). "Equation-Free Multiscale Computation: Algorithms and Applications". In: *Annual Review of Physical Chemistry* 60.1, pp. 321–344. DOI: 10.1146/annurev.physchem.59.032607.093610.

Kim, Inhi, Ronald Galiza, and Luis Ferreira (2013). "Modeling pedestrian queuing using micro-simulation". In: *Transportation Research Part A: Policy and Practice* 49.0, pp. 232–240. DOI: 10.1016/j.tra.2013.01.018.

Kingman, J. F. C. (2009). "The first Erlang century–and the next". In: *Queueing Systems* 63.1-4, pp. 3–12. ISSN: 1572-9443. DOI: 10.1007/s11134-009-9147-4.

Kneidl, Angelika (2015). "How do people queue – a study of different queuing models". In: *Traffic and Granular Flow '15.* Springer.

Kneidl, Angelika, Dirk Hartmann Hartmann, and André Borrmann (2013). "A hybrid multi-scale approach for simulation of pedestrian dynamics". In: *Transportation Research Part C: Emerging Technologies* 37, pp. 223–237. DOI: 10.1016/j.trc.2013.03.005.

Kuehn, Christian (2015). "Multiple Time Scale Dynamics". In: *Applied Mathematical Sciences.* ISSN: 2196-968X. DOI: 10.1007/978-3-319-12316-5.

Kuznetsov, Yuri A. (2004). *Elements of Applied Bifurcation Theory.* Springer New York. DOI: 10.1007/978-1-4757-3978-7.

Lachapelle, Aimé and Marie-Therese Wolfram (2011). "On a mean field game approach modeling congestion and aversion in pedestrian crowds". In: *Transportation Research Part B: Methodological* 45.10, pp. 1572–1589. DOI: doi:10.1016/j.trb.2011.07.011.

Lafon, S. and A.B. Lee (2006). "Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 28.9, pp. 1393–1403. ISSN: 2160-9292. DOI: 10.1109/tpami.2006.184.

Lapedes, Alan and Robert Farber (1989). "How Neural Nets Work". In: *Evolution, Learning and Cognition*, pp. 331–346. DOI: 10.1142/9789814434102_0012.

Lee, John M. (2012). *Introduction to Smooth Manifolds*. Springer New York. DOI: 10.1007/978-1-4419-9982-5.

Legoll, Frédéric and Tony Lelièvre (2010). "Effective dynamics using conditional expectations". In: *Nonlinearity* 23.9, pp. 2131–2163. ISSN: 1361-6544. DOI: 10.1088/0951-7715/23/9/006.

Li, Jinghai et al. (2004). "Multi-scale methodology for complex systems". In: *Chemical Engineering Science* 59.8-9, pp. 1687–1700. ISSN: 0009-2509. DOI: 10.1016/j.ces.2004.01.025.

Liddle, Jack et al. (2011). "Microscopic insights into pedestrian motion through a bottleneck, resolving spatial and temporal variations". In: *arXiv* 1105.1532.v1. URL: http://arxiv.org/abs/1105.1532.

Loève, Michel (1946). "Fonctions alèatoires de second ordre (French)". In: *Revue Sci.* 84, pp. 195–206.

Luding, Stefan (2006). "About contact force-laws for cohesive frictional materials in 2D and 3D". In: URL: http://www2.msm.ctw.utwente.nl/sluding/PAPERS/Luding_ING1.pdf.

— (2008). "Introduction to discrete element methods". In: *European Journal of Environmental and Civil Engineering* 12.7-8, pp. 785–826. DOI: 10.1080/19648189.2008.9693050. eprint: http://dx.doi.org/10.1080/19648189.2008.9693050.

Marschler, Christian et al. (2013). "Implicit Methods for Equation-Free Analysis: Convergence Results and Analysis of Emergent Waves in Microscopic Traffic Models". In: *arXiv* 1301.6044.v1. URL: http://arxiv.org/abs/1301.6044.

— (2014). "Implicit Methods for Equation-Free Analysis: Convergence Results and Analysis of Emergent Waves in Microscopic Traffic Models". In: *SIAM Journal on Applied Dynamical Systems* 13.3, pp. 1202–1238. ISSN: 1536-0040. DOI: 10.1137/130913961.

Moon, Sung Joon, S. Sundaresan, and I. G. Kevrekidis (2007). "Coarse-grained computations of demixing in dense gas-fluidized beds". In: *Physical Review E* 75.5. ISSN: 1550-2376. DOI: 10.1103/physreve.75.051309.

Moore, B. (1981). "Principal component analysis in linear systems: Controllability, observability, and model reduction". In: *IEEE Trans. Automat. Contr.* 26.1, pp. 17–32. ISSN: 0018-9286. DOI: 10.1109/tac.1981.1102568.

Moussaïd, Mehdi et al. (2012). "Traffic Instabilities in Self-Organized Pedestrian Crowds". In: *PLoS Computational Biology* 8.3, e1002442. DOI: 10.1371/journal.pcbi.1002442.

Mroz, Hubert, Jarosław Was, and Pawel Topa (2014). "Parallel Processing and Applied Mathematics: 10th International Conference, PPAM 2013, Warsaw, Poland, September 8-11, 2013, Revised Selected Papers, Part II". In: Springer Berlin Heidelberg. Chap. The Use of GPGPU in Continuous and Discrete Models of Crowd Dynamics, pp. 679 –688. ISBN: 978-3-642-55195-6. DOI: 10.1007/978-3-642-55195-6_64.

Müller, Matthias S. (2001). "FAST ALGORITHMS FOR THE SIMULATION OF GRANULAR PARTICLES". PhD thesis. Universität Stuttgart. URL: http://elib.uni-stuttgart.de/opus/volltexte/2002/988/pdf/diss.pdf.

Nagai, Ken H. et al. (2015). "Collective Motion of Self-Propelled Particles with Memory". In: *Physical Review Letters* 114 (16), p. 168001. DOI: 10.1103/PhysRevLett.114.168001.

Nishinari, Katsuhiro, Minoru Fukui, and Andreas Schadschneider (2004). "A stochastic cellular automaton model for traffic flow with multiple metastable states". In: *Journal of Physics A: Mathematical and General* 37.9, p. 3101. DOI: 10.1088/0305-4470/37/9/003.

Orosz, Gábor, R. Eddie Wilson, and Bernd Krauskopf (2004). "Global bifurcation investigation of an optimal velocity traffic model with driver reaction time". In: *Physical Review E* 70 (2), p. 026207. DOI: 10.1103/PhysRevE.70.026207.

Panton, Ronald L. (2013). *Incompressible Flow*. 4th. Wiley.

Pavliotis, Grigoris and Andrew Stuart (2008). *Multiscale Methods: Averaging and Homogenization (Texts in Applied Mathematics)*. Springer. ISBN: 0387738282.

Pearson, Karl (1901). "LIII. On lines and planes of closest fit to systems of points in space". In: *Philosophical Magazine Series 6* 2.11, pp. 559–572. ISSN: 1941-5990. DOI: 10.1080/14786440109462720.

Peherstorfer, Benjamin et al. (2014). "Localized Discrete Empirical Interpolation Method". In: *SIAM Journal on Scientific Computing* 36.1, A168–A192. ISSN: 1095-7197. DOI: 10.1137/130924408.

Perrault-Joncas, Dominique and Marina Meilă (2013). "Non-linear dimensionality reduction: Riemannian metric estimation and the problem of geometric discovery". In: *arXiv* 1305.7255.v1. eprint: 1305.7255.

Perrault-Joncas, Dominique C. and Marina Meilă (2011). "Directed Graph Embedding: an Algorithm based on Continuous Limits of Laplacian-type Operators". In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor et al. Curran Associates, Inc., pp. 990–998. URL: http://papers.nips.cc/paper/4282-directed-graph-embedding-an-algorithm-based-on-continuous-limits-of-laplacian-type-operators.pdf.

Qian, Zhiguang et al. (2006). "Building Surrogate Models Based on Detailed and Approximate Simulations". In: *Journal of Mechanical Design* 128.4, p. 668. ISSN: 1050-0472. DOI: 10.1115/1.2179459.

Quarteroni, Alfio and Alberto Valli (1999). *Domain Decomposition Methods for Partial Differential Equations*. Numerical Mathematics and Scientific Computation Series. Oxford University Press, USA.

Rao, K. Kesava and Prabhu R. Nott (2008). *An Introduction to Granular Flow*. Cambridge Series in Chemical Engineering. Cambridge: Cambridge University Press.

Reicher, Stephen, Russell Spears, and S. Alexander Haslam (2010). "The Social Identity Approach in Social Psychology". In: *The SAGE Handbook of Identities*. Ed. by Chandra Talpade Mohanty Margaret Wetherell. SAGE Publications Ltd, pp. 45–63. DOI: 10.4135/9781446200889.

Reynolds, Craig W. (1987). "Flocks, herds and schools: A distributed behavioral model". In: *ACM SIGGRAPH Computer Graphics* 21.4, pp. 25–34. DOI: 10.1145/37402.37406.

Richmond, P. and D. Romano (2008). “A High Performance Framework For Agent Based Pedestrian Dynamics on GPU hardware”. In: *European Simulation and Modelling.* URL: http://www.dcs.shef.ac.uk/\~{}daniela/pedestrians.pdf.

Romijn, Reinout et al. (2008). “A grey-box modeling approach for the reduction of nonlinear systems”. In: *Journal of Process Control* 18.9, pp. 906–914. ISSN: 0959-1524. DOI: 10.1016/j.jprocont.2008.06.007.

Rowley, Clarence W. et al. (2009). “Spectral analysis of nonlinear flows”. In: *Journal of Fluid Mechanics* 641, p. 115. ISSN: 1469-7645. DOI: 10.1017/s0022112009992059.

Ruelle, David and Floris Takens (1971). “On the nature of turbulence”. In: *Commun.Math. Phys.* 20.3, pp. 167–192. ISSN: 1432-0916. DOI: 10.1007/bf01646553.

Sacks, Jerome et al. (1989). “Design and Analysis of Computer Experiments”. In: *Statistical Science* 4.4, pp. 409–423. ISSN: 0883-4237. DOI: 10.1214/ss/1177012413.

Saltelli, Andrea, Stefano Tarantola, and Karen Chan (1999). “A Quantitative Model-Independent Method for Global Sensitivity Analysis of Model Output”. In: *Technometrics* 41.1, pp. 39–56. DOI: 10.1080/00401706.1999.10485594.

Schilders, Wilhelmus H.A., Joost Rommes, and Henk A. van der Vorst, eds. (2008). *Model Order Reduction: Theory, Research Aspects and Applications.* Springer Berlin Heidelberg. ISBN: http://id.crossref.org/isbn/978-3-540-78841-6. DOI: 10.1007/978-3-540-78841-6.

Schmid, Peter J. (2010). “Dynamic mode decomposition of numerical and experimental data”. In: *Journal of Fluid Mechanics* 656, pp. 5–28. DOI: 10.1017/s0022112010001217.

Schröder, Benjamin et al. (2014). “High parametric CFD-analysis of fire scenarios in underground train stations using statistical methods and climate modelling”. In: *10th International Conference on Performance Based Codes and Fire Safety Design Methods.*

Seitz, Michael J. and Gerta Köster (2012). “Natural discretization of pedestrian movement in continuous space”. In: *Physical Review E* 86.4, p. 046108. DOI: 10.1103/PhysRevE.86.046108.

Seyfried, Armin et al. (2010). “Enhanced Empirical Data for the Fundamental Diagram and the Flow Through Bottlenecks”. In: *Pedestrian and Evacuation Dynamics 2008.* Ed. by Wolfram W. F. Klingsch et al. Springer Berlin Heidelberg, pp. 145–156. DOI: 10.1007/978-3-642-04504-2_11.

Shalizi, Cosma Rohilla (2006). “Methods and Techniques of Complex Systems Science: An Overview”. In: *Topics in Biomedical Engineering International Book Series*, pp. 33–114. DOI: 10.1007/978-0-387-33532-2_2.

Sime, Jonathan D. (1995). “Crowd psychology and engineering”. In: *Safety Science* 21.1, pp. 1–14. DOI: 10.1016/0925-7535(96)81011-3.

Singer, Amit et al. (2009). “Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps”. In: *Proceedings of the National Academy of Sciences* 106, pp. 16090–16095. DOI: 10.1073/pnas.0905547106.

Sirovich, Lawrence (1987). “Turbulence and the dynamics of coherent structures Part I-III: dynamics and scaling”. In: *Quarterly of Applied Mathematics* 45.3, pp. 583–590.

Sivers, Isabella von and Gerta Köster (2015). "Dynamic Stride Length Adaptation According to Utility And Personal Space". In: *Transportation Research Part B: Methodological* 74, pp. 104 –117. DOI: 10.1016/j.trb.2015.01.009.

Sivers, Isabella von et al. (2016). "Modelling social identification and helping in evacuation simulation". In: *Safety Science* 89, pp. 288–300. ISSN: 0925-7535. DOI: http://dx.doi.org/10.1016/j.ssci.2016.07.001.

Small, Michael and C.K. Tse (2004). "Optimal embedding parameters: a modelling paradigm". In: *Physica D: Nonlinear Phenomena* 194.3-4, pp. 283–296. DOI: 10.1016/j.physd.2004.03.006.

Smith, Alastair et al. (2009). "Modelling contra-flow in crowd dynamics DEM simulation". In: *Safety Science* 47.3, pp. 395–404. DOI: 10.1016/j.ssci.2008.05.006.

Smith, Lincoln et al. (2007). "Linked Local Navigation for Visual Route Guidance". In: *Adaptive Behavior* 15.3, pp. 257–271. DOI: 10.1177/1059712307082091.

Smith, Ralph C. (2014). *Uncertainty Quantification: Theory, Implementation, and Applications.* Computational Science and Engineering. Society for Industrial and Applied Mathematics. ISBN: 978-1-611973-21-1. URL: http://www.cambridge.org/de/academic/subjects/mathematics/mathematical-modelling-and-methods/uncertainty-quantification-theory-implementation-and-applications.

Stark, J. et al. (1997). "Takens embedding theorems for forced and stochastic systems". In: *Nonlinear Analysis: Theory, Methods & Applications* 30.8, pp. 5303–5314. DOI: 10.1016/s0362-546x(96)00149-6. URL: http://dx.doi.org/10.1016/S0362-546X(96)00149-6.

Stewart, G. W. (1993). "On the Early History of the Singular Value Decomposition". In: *SIAM Review* 35.4, pp. 551–566. ISSN: 1095-7200. DOI: 10.1137/1035134.

Stuart, A. M. and A. R. Humphries (1996). *Dynamical Systems and Numerical Analysis.* Ed. by P.G. Ciarlet et al. Cambridge: Cambridge University Press.

Sud, Avneesh et al. (2008). "Real-Time Path Planning in Dynamic Virtual Environments Using Multiagent Navigation Graphs". In: *IEEE Transactions on Visualization and Computer Graphics* 14, pp. 526–538. ISSN: 1077-2626. DOI: 10.1109/TVCG.2008.27.

Sullivan, T.J. (2015). *Introduction to Uncertainty Quantification.* 1st ed. Springer International Publishing. ISBN: http://id.crossref.org/isbn/978-3-319-23395-6. DOI: 10.1007/978-3-319-23395-6.

Sumpter, David J. T., Richard P. Mann, and Andrea Perna (2012). "The modelling cycle for collective animal behaviour". In: *Interface Focus* 2.6, pp. 764–773. DOI: 10.1098/rsfs.2012.0031. eprint: http://rsfs.royalsocietypublishing.org/content/2/6/764.full.pdf.

Tadmor, E. B., M. Ortiz, and R. Phillips (1996). "Quasicontinuum analysis of defects in solids". In: *Philosophical Magazine A* 73.6, pp. 1529–1563. ISSN: 1460-6992. DOI: 10.1080/01418619608243000.

Takens, Floris (1981). "Detecting strange attractors in turbulence". In: *Lecture Notes in Mathematics*, pp. 366–381. DOI: 10.1007/bfb0091924.

Tejchman, Jacek (2013). *Confined Granular Flow in Silos.* Springer Science & Business Media. DOI: 10.1007/978-3-319-00318-4.

Theodoropoulos, Constantinos, Yue-Hong Qian, and Ioannis G. Kevrekidis (2000). “Coarse stability and bifurcation analysis using time-steppers: A reaction-diffusion example”. In: *Proceedings of the National Academy of Sciences* 97.18, pp. 9840–9843. DOI: 10.1073/pnas.97.18.9840. eprint: http://www.pnas.org/content/97/18/9840.full.pdf+html.

Trujillo, Leonardo and Leonardo Di. G. Sigalotti (2014). “Theoretical Physics of Granular Fluids and Solids”. In: *Computational and Experimental Fluid Mechanics with Applications to Physics, Engineering and the Environment.* Ed. by Leonardo Di G. Sigalotti, Jaime Klapp, and Eloy Sira. Environmental Science and Engineering. Springer International Publishing, pp. 165–91. DOI: 10.1007/978-3-319-00191-3_8.

Turner, John C. et al. (1987). *Rediscovering the social group: A self-categorization theory.* Ed. by John C. Turner. Basil Blackwell.

Vadere Crowd Simulation (2016). URL: https://gitlab.lrz.de/vadere/vadere.

Verhulst, Ferdinand and Taoufik Bakri (2007). “The dynamics of slow manifolds”. In: *Journal of the Indonesian Mathematical Society* 13, pp. 1–16.

Vicsek, Tamás et al. (1995). “Novel Type of Phase Transition in a System of Self-Driven Particles”. In: *Physical Review Letters* 75 (6), pp. 1226–1229. DOI: 10.1103/PhysRevLett.75.1226.

Waldleitner, Christoph (2015). “Three-Dimensional Kinetic Simulation of Granular Matter”. Bachelor’s thesis.

Weidmann, Ulrich (1992). *Transporttechnik der Fussgänger.* 2nd. Vol. 90. Schriftenreihe des IVT. Zürich: Institut für Verkehrsplanung, Transporttechnik, Strassen- und Eisenbahnbau (IVT) ETH. DOI: 10.3929/ethz-a-000687810.

Williams, Matthew O., Ioannis G. Kevrekidis, and Clarence W. Rowley (2015). “A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition”. In: *J. Nonlinear Sci.* 25.6, pp. 1307–1346. ISSN: 1432-1467. DOI: 10.1007/s00332-015-9258-5.

Williams, Matthew O., Clarence W. Rowley, and Ioannis G. Kevrekidis (2014). “A Kernel-Based Approach to Data-Driven Koopman Spectral Analysis”. In: *arXiv* v4.1411.2260.

Williams, Matthew O. et al. (2015). “Data fusion via intrinsic dynamic variables: An application of data-driven Koopman spectral analysis”. In: *EPL (Europhysics Letters)* 109.4. ISSN: 1286-4854. DOI: 10.1209/0295-5075/109/40007.

Zinke, Robert, Gesine Hofinger, and Laura Künzer (2013). “Psychological Aspects of Human Dynamics in Underground Evacuation: Field Experiments”. In: *Pedestrian and Evacuation Dynamics 2012.* Springer Science + Business Media, pp. 1149–1162. ISBN: http://id.crossref.org/isbn/978-3-319-02447-9. DOI: 10.1007/978-3-319-02447-9_94.

Zönnchen, Benedikt and Gerta Köster (2016). “Detecting Arbitrarily Shaped Queues Using the Fast Marching Method”. In: *Proceedings of the 8th International Conference on Pedestrian and Evacuation Dynamics (PED2016).* Hefei, China.