Learning Reduced Models for Large-Scale Agent-Based Systems

Dissertation

zur Erlangung des Grades eines

Doktors der Naturwissenschaften
– Dr. rer. nat. –

am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

vorgelegt von

Jan-Hendrik Niemann

Berlin, 2022

Betreuer: Prof. Dr. Christof Schütte

Erstgutachter: Prof. Dr. Christof Schütte
Zweitgutachterin: Prof. Dr. Susanna Röblitz

Tag der Disputation: 11. Juli, 2022

Declaration of authorship

I, Jan-Hendrik Niemann, hereby declare to the Freie Universität Berlin that I have completed the submitted dissertation independently and without the use of sources and aids other than those indicated. The present thesis is free of plagiarism. I have marked as such all statements that are taken literally or in content from other writings. This dissertation has not been submitted in the same or similar form in any previous doctoral procedure.

I agree to have my thesis examined by a plagiarism examination software.

Berlin, February 15, 2022 _____

Acknowledgments

It is my pleasure to thank all those who have supported me during the last three years of my research activity. First and foremost, I would like to thank my supervisor Christof Schütte for giving me the opportunity to write this thesis at the Zuse Institute Berlin. I am thankful for his guidance and encouragement.

I am very grateful to Stefan Klus for introducing me to Koopman operator theory and the field of data-driven dynamical systems, as well as his steady guidance through each stage of my research. We had many fruitful discussions and I am thankful for the consistent support and collaboration.

Many thanks to all my colleagues at the Zuse Institute Berlin and Global Climate Forum Berlin for the interesting conversations, professional and private discussions and valuable advice. Especially, I thank Sarah Wolf and Stefanie Winkelmann for giving me new perspectives on agents, our collaboration and for always having an open ear for me. I wish to extend my special thanks to all who helped me to improve this work.

I acknowledge the support provided by Germany's Excellence Strategy (MATH+: The Berlin Mathematics Research Center, EXC-2046/1, project ID: 390685689) and the Berlin Mathematical School.

In particular, I am forever thankful to my parents Christine and Engelbert and my brother Leonhard for their loving emotional and practical support. Thank you for the strength you give me. Last but not least, I am endlessly grateful to my girlfriend Gaby for her unconditional love and support throughout every day. Gracias por estar siempre conmigo.

Abstract

Modeling social systems and studying their dynamical behavior plays an important role in many fields of research. Agent-based modeling provides a high degree of detail into artificial societies by describing the model from the perspective of the agents. The interactions of agents, often characterized by simple rules, lead to complex, time-evolving patterns. Their understanding is of great importance, e.g., for predicting and influencing epidemics. Analysis and simulation, however, often becomes prohibitively time-consuming when the number of agents or the time scale of interest is large. Therefore, this thesis is devoted to learn significantly reduced models of large-scale agent-based systems from simulation data. We show how data-driven methods based on transfer operators can be used to find reduced models represented by ordinary or stochastic differential equations that describe the dynamical behavior of larger groups or entire populations and thus enable the analysis and prediction of agent-based systems. To this end, we first present an extension of EDMD (*extended dynamic mode decomposition*) called gEDMD to approximate the Koopman generator from data. This method can be used to compute eigenfunctions, eigenvalues, and modes of the generator, as well as for system identification and model reduction of both deterministic and non-deterministic dynamical systems. Secondly, we analyze the long-term behavior of certain agent-based models and their pathwise approximations by stochastic differential equations for large numbers of agents using transfer operators. We show that, under certain conditions, the transfer operator approach connects the pathwise approximations on finite time scales with methods for describing the behavior on possibly exponentially long time scales. As a consequence, we can use the finite-time, pathwise approximations to characterize metastable behavior on long time scales using transfer operators. This can significantly reduce the computational cost. The third part addresses the data-driven model reduction since in many cases no analytical limit models are known or existent. We show how the Koopman operator theory can be used to infer the governing equations of agent-based systems directly from simulation data. Using benchmark problems, we demonstrate that for sufficiently large population sizes the data-driven models agree well with analytical limit equations and, moreover, that the reduced models allow predictions even in cases far from the limit or when no limit equations are known. Lastly, we demonstrate the potential of the presented approach. We present an ansatz for the multi-objective optimization of agent-based systems with the help of data-driven surrogate models based on the Koopman generator. In particular, when limit models are unknown or non-existent, this approach makes multi-objective optimization problems solvable that would otherwise be computationally infeasible due to very expensive objective functions.

Contents

Veniet tempus quo ista quae nunc latent in lucem dies extrahat et longioris aeui diligentia. Ad inquisitionem tantorum aetas una non sufficit, ut tota caelo uacet; quid quod tam paucos annos inter studia ac uitia non aequa portione diuidimus? Itaque per successiones ista longas explicabuntur. Veniet tempus quo posteri nostri tam aperta nos nescisse mirentur.

The time will come when diligent research over very long periods will bring to light things which now lie hidden. A single lifetime, even though entirely devoted to the sky, would not be enough for the investigation of so vast a subject. What about the fact that we do not divide our few years in an equal portion at least between study and vice? And so this knowledge will be unfolded only through long successive ages. There will come a time when our descendants will be amazed that we did not knew things that are so plain to them.

Seneca, *Naturales Quaestiones*, Book VII, Chapter XXV

1 Introduction

Modeling social dynamics and studying the collective phenomena that emerge from them is an important task in research. Agent-based modeling offers an intuitive and powerful framework to model interaction and collaboration of single agents, smaller groups or entire populations. A high degree of modeling flexibility allows major insights into complex dynamic patterns emerging from the interactions of discrete entities, so-called *agents*, that are equipped with often simple rules describing their behavior. This makes them equally suitable for experts and layperson without extensive mathematical knowledge. In many cases, a large number of agents is not only desired but also required, e.g., to represent initially low numbers of agents and thus increase the resolution. With up to 1.5 million agents the *Mobility Transition Model* [117] is one such case. Built to support political and social decision-making processes, the model simulates the evolution of private mobility demand allowing to find and test different strategies to reduce the carbon dioxide emissions. To better understand the various processes involved and gain deeper insights into agent-based models, data-driven methods help to analyze the vast amounts of data generated.

Agent-Based Modeling. Agent-based modeling is a microscopic modeling technique that describes a model from the perspective of each single agent. The model is defined by a set of autonomous agents acting and interacting in a shared environment. Its counterpart is macroscopic modeling, which describes the system globally at a high level, e.g., in terms of differential equations or mean-field approximations. An agent can represent any kind of discrete entity, such as individuals, groups or organizations, but also atoms, molecules, or technical devices. Each agent is described by a set of instructions that govern its behavior, i.e., how it interacts with other agents and the environment. Interactions reach from simple extraction of information or goods, to exchange thereof up to complex processes like organization, coordination or cooperation. The environment can be very versatile and include, e.g., geographical conditions, (social) networks and infrastructures. Complex combinations of the aforementioned but also abstract constructs such as rules, laws and norms can be set up as a common environment or integrated into it. Note that not all environmental information must be available to each agent at all times. It can be locally restricted or only temporarily available.

Agent-based modeling has been an ongoing process since 1957, when microsimulation, a technique developed by Orcutt [129], first focused on individuals and their interactions in economic and social systems. However, its roots go back to the 1940s, when von Neumann and Ulam introduced the concept of cellular automata, see [164, 167]. Motivated by the increasing computational power of computer systems in the 1990s,

simulations of agent-based models to analyze their dynamics became the most common tool. Nowadays, many models are implemented either using software frameworks such as NetLogo [170] or directly via programming languages, possibly using toolboxes such as Mesa [77], see, e.g., [1] for a recent review on software for agent-based modeling. A milestone of agent-based modeling was set in the 1970s by Clifford and Sudbury [29] with the introduction of the so-called *voter model*, which marks one of the most prominent applications of agent-based modeling: opinion dynamics. The term was later on coined by Holley and Liggett [71]. The basic idea of the voter model is that agents imitate the opinion of their neighbors. Over the years, many modifications and extensions concerning, for instance, the representation and number of distinct opinions, imitation and interaction, or environments have been developed, see, e.g., [25, 158, 74, 140]. Further milestones in the development of agent-based modeling are, for example, Schelling's *model of segregation* [149] or the *sugarscape model* [46], developed by Epstein and Axtell. Other applications concern, for instance, innovation spreading or infection kinetics [80, 48], emergency response planning [123, 41] or the evolution of languages [99, 159], but also biological systems on various scales [139], see also [25, 5] for overviews of social dynamics and agent-based systems in biology.

The last decades have also produced a wide range of mathematical applications, from (highly detailed) microscopic stochastic descriptions following spatial movement and neighbor interactions [39], individual-based stochastic descriptions in networks without movement [15], Markov chain approaches for collective population dynamics [9] to descriptions in terms of ordinary or stochastic (partial) differential equations for very large numbers of agents [111, 165, 64, 120, 68]. Like the applications, the methods and techniques for modeling agent-based systems are diverse, reaching from data-based micro-simulations of synthetic populations [166] up to abstract individual- or agent-based and multi-agent models. In addition, different communities have developed their own terms and language. The term *individual-based* is commonly used in ecology, while the term *multi-agent* is usually used in control engineering and computer science. In social sciences, the term *agent-based* is most commonly used. While the terms have many things in common, they also differ from one another. For example, while a real multi-agent system, e.g., consisting of unmanned autonomous vehicles, can be simulated by an agent-based model, a virtual multi-agent system denotes the same as an agent-based model.

Agent-based modeling is advantageous whenever complex interactions between discrete entities are to be modeled, the interaction with the environment or its influence or both cannot be neglected, heterogeneity among the agents is desired or required, or agents learn autonomously and adapt to new situations, see also [16, 25, 80]. However, echoing a quote attributed to Einstein and paraphrased in the New York Times in the 1950s that "everything should be made as simple as possible, but no simpler", agent-based models should be made as complex as necessary, but no more complex. To ensure this, documentation is an important challenge that every modeler must face. Guidelines for documentation and characterization of agent-based models have been developed. The ODD (overview, design concepts, details) protocol [65] standardizes the

documentation based on seven aspects designed to make any agent-based model more understandable and reproducible. These concern the purpose of the model, its entities, state variables and scales, an overview of process and scheduling, the description of design concepts (i.e. basic principles, emergence, adaptation, objectives, learning, prediction, sensing, interaction, stochasticity, collectives, observation), the initialization, its input data, and submodels. Of particular importance is *scheduling*, occasionally referred to as *activation*, since the vast majority of agent-based models follows a discrete-time regime. The question behind scheduling is in which order which agent(s) influence(s) which other agent(s) at each time step [169]. To simplify comprehensibility and standardize the discussion of time schedules, the SAS (synchrony, actor type, scale) classification [169] has been proposed. Most commonly used are randomized interaction schedules, which means that at each time step the order of interacting agents changes. A related issue is whether agents can "see" the changes made by previously acting agents in the same time step. This does not only affect the dynamical collective behavior of the system but also plays a role when the model is to be parallelized. See also [31] for a study on the impact of different scheduling approaches. The scheduling problem can naturally be circumvented by using a continuous-time regime since here it is assumed that no two actions occur at exactly the same time. Additionally, a continuous-time schedule can be interpreted as less artificial when modeling systems of social, living individuals. A variety of continuous-time approaches, for instance, based on Markov processes [3, 39] have been developed. A corresponding software framework is proposed in [168]. In this thesis, we consider agent-based models of both time regimes.

Data-Driven Science. With the abundance of data, whether from archaeological or present day records, experiments, or numerical simulations, data science has emerged to encompass almost all areas of research today. Dynamical systems are the key to success to understand the complex processes which created the data. Their analysis is of great importance for various applications in physical, biological, engineering, but also in social sciences. Many of them are characterized as nonlinear and high-dimensional systems, most of which may also be affected by inaccuracies or uncertainties. Data-driven methods are able to extract information about the behavior of dynamical systems from data without requiring a priori knowledge about the system itself. This makes these methods suitable for many problems where a closed system description is not possible or available. In recent years, methods have been developed, e.g., to approximate the associated transfer operators, to gain insight into coherent and metastable sets, or to perform stability analyses, but also to infer the governing equations of the underlying dynamics, for model reduction, or for optimization and control of dynamical systems, e.g., [109, 82, 95, 88, 7, 134, 135, 136, 89].

Neural network-based approaches represent one class of data-driven methods. Inspired by nature, neural networks consist of many hierarchical connected layers of so-called *artificial neurons*, which receive signals from other neurons, process them, and send them to other neurons. Each neuron represents a nonlinear function applied to a

weighted sum of the neuron's inputs. The universal approximation theorem guarantees that any function can be represented by a neural network with sufficiently many layers and linear output layer [107]. Although neural networks are extremely powerful and have high predictive accuracy, they often do not lead to interpretable models of the dynamics.

Another class of data-driven methods, which we will focus on in this thesis, builds on the Koopman and Perron–Frobenius transfer operators associated with the dynamical system. Their eigenvalues and eigenvectors provide important information about the global behavior of the system. In appropriately defined function spaces, both operators are adjoint to each other. However, practical considerations justify the use of one or the other operator. The Perron–Frobenius operator describes the evolution of densities, which can be thought of as the evolution of mass distribution under the action of a flow [23]. The Koopman operator describes the evolution of observables that represent any type of measurement that can be made, e.g., during an experiment or simulation. The benefit is that instead of analyzing nonlinear but finite-dimensional dynamical systems, one can consider the linear but infinite-dimensional operators associated with the dynamical system. Linearity then allows the application of many methods developed for linear systems.

Typically, a projection onto finite-dimensional subspaces spanned by a set of basis functions is done to solve the infinite-dimensional problem numerically. Ideally, the projected representations are sparse and allow for interpretation. Data-driven algorithms such as *dynamic mode decomposition* (DMD) [150, 163], which is shown to be related to the Koopman operator [145, 23], or its generalization *extended dynamic mode decomposition* (EDMD) [171, 82], which provides higher accuracy in approximating the Koopman operator, can be used to analyze high-dimensional nonlinear dynamical systems. Building on EDMD, a method described in [109, 110] uses the matrix logarithm to obtain a finite-dimensional representation of the Koopman generator, which is then used to identify the governing equations of the underlying dynamics, thus showing the connection between the governing equations of a dynamical system, the corresponding transfer operators and generators. Another popular data-driven method is called *sparse identification of nonlinear dynamics* (SINDy) [20], which, however, was developed to learn the governing equations and not the transfer operator. All aforementioned algorithms have constantly been extended and generalized to kernel- [171, 157, 86], tensor- [81, 59, 26], or neural network-based [101, 107, 108] versions.

Aim of This Thesis. Let us return to the Mobility Transition Model. In order to make reasonable statements about carbon dioxide emissions, one must of course be interested in the mobility behavior of the population as a whole and not in the individual decisions of each agent. However, due to the very large number of agents, simulation and analysis of the Mobility Transition Model and agent-based models in general is time-consuming and impractical. To explain the past and, more importantly, to forecast the future in order to develop measures to reduce greenhouse gases in the mobility sector, we are

interested in the evolution of aggregate state variables, such as the number of cars with internal combustion engines. Therefore, the aim of this thesis is to learn reduced models of agent-based systems directly from data that allow for analysis and prediction of the collective behavior of larger groups or entire populations. As agent-based models often lack closed system descriptions, data-driven learning methods are used to find coarse-grained models representing the aggregate dynamics of large-scale agent-based systems.

Outline of This Thesis

This work was preceded by several publications to which the author made significant contributions. They are referred to at the beginning of the respective chapters.

Theoretical Background. In Chapter 2, we introduce all mathematical concepts that are relevant to this thesis. The first part is concerned with the Koopman and Perron–Frobenius transfer operators and their generators associated with deterministic and non-deterministic dynamical systems. We provide an overview of their spectral properties, as well as numerical approximation methods. The second part presents agent-based models, their representation as continuous-time Markov jump processes, and, in this case, known approximation results for aggregate dynamics for large numbers of agents. We also present three models that recur as guiding examples in this thesis.

Data-Driven Approximation of the Koopman Generator. Chapter 3 concentrates on the data-driven approximation of the infinitesimal generator of the Koopman operator, since the generator can also be used to extract important properties of dynamical systems. Due to its close connection to the operator, the Koopman generator recently attracted attention, for instance, to estimate parameters of stochastic differential equations [141], when computing its eigenfunctions from Galerkin-projected eigenvalue problems [60], or when approximating its adjoint, the Perron–Frobenius generator [56]. The novel framework is derived from the classic EDMD algorithm and computes a finite-dimensional matrix representation of the Koopman generator without requiring integration of trajectories. In addition to eigenfunctions, eigenvalues and modes, the framework also identifies the governing equations not only for deterministic but also for non-deterministic dynamical systems from data. We compare its efficacy and show relationships with other methods for system identification.

Population Limits and Large Time Scales. In Chapter 4, we study continuous-time agent-based systems and their pathwise approximation by ordinary and stochastic differential equations for large populations on long time scales. When the number of agents N or the time scale of interest T or both become large, simulation-based analysis of agent-based models becomes impractical as the computational cost virtually explodes due to more operations per time step and the number of time steps in total. The two best-known approaches to addressing these challenges concern

(A) the large population limit $N \to \infty$ for *large but finite*, N-independent time scales, where appropriate limit equations, which are computationally and analytically inexpensive, describe the aggregate dynamics of the agent-based system, or

(B) the behavior on *exponentially long* time scales $T \asymp \exp(\zeta N)$ via the large deviation principle or WKB approximations (after Wentzel, Kramers and Brillouin).

For large numbers of agents N and fixed finite time intervals $[0, T]$, i.e., case (A), the main known result is the approximation of an agent-based model described as Markov jump process by a closed system of (mean-field) ordinary differential equations [91, 47] that can be extended to stochastic differential equations to reproduce also fluctuations [93, 47]. It is important to say that both are pathwise results, i.e., the trajectory of the (rescaled) agent-based model converges for $N \to \infty$ and fixed $T < \infty$ to the trajectories of the limit equations. In the second case (B), when we consider time intervals that scale exponentially with N, i.e., $T \asymp \exp(\zeta N)$, rare events, that occur with very low probability $p \asymp \exp(-\zeta N)$ as large deviations from the expected dynamical behavior of the agent-based model, can accurately be described by asymptotic expansion approaches using the WKB method [8] or the large deviation principle [121]. We consider metastability as one example. A dynamical system is in a metastable state when the process is attracted to one state for a long time before it transitions to another (meta)-stable state. Understanding these rare events, i.e., the transition between metastable states, can be of interest for many applications, e.g., to mitigate or prevent disasters or to initiate changes. The concepts (A) and (B) can be related to each other via the behavior on long (but not necessarily exponentially long) time scales connected to metastability. We demonstrate how transfer operator approaches, e.g., so-called *Markov state models* (MSM) [153, 19], adapted to agent-based systems, can help to close the gap between *finite* and *exponentially long* time scales. Using results from (A), i.e., many *short* trajectories of the corresponding population limit, the transfer operator approach can characterize rare events on (exponentially) long time scales for fixed large N, i.e., (B). This opens up the possibility of computing in parallel and thus considerably reducing the computation time. Analogous issues concerning metastability have been discussed for (discrete-time) Markov chain approaches [66] and the Fokker–Planck equation [58].

Data-Driven Model Reduction of Agent-Based Systems. Chapter 5 addresses the data-driven model reduction of agent-based systems and connects the results of the previous two chapters. The drawback of the approach in Chapter 4 and, in general, methods based on analytically derived differential equations or Markov chain approaches is that they require insider knowledge of the model itself, which, however, might not be available. Therefore, data-driven learning of the governing equations of social dynamics is of growing interest. Several approaches have already been discussed in the literature, e.g., the so-called *equation-free* approach introduced by Kevrekidis et al. [79, 78]. Designed to circumvent the derivation of macroscopic equations at system-level when they are believed to exist but cannot be expressed in closed form, the equation-free approach

is used in [177] to obtain a coarse-grained model of a spatio-temporally varying agent-based system. This approach is also used for bifurcation, stability or rare event analysis of agent-based models, see [162, 103]. The *equation-free-variable-free* approach [102] is an extension of the aforementioned, which includes diffusion maps to learn also the essential coarse-grained variables. Two other data-driven approaches are, e.g., one non-parametric approach similar to parameter estimation problems for ordinary differential equations [105], or one based on SINDy to learn systems of ordinary differential equations from stochastic agent-based models [124]. Although terms like *metastability* or *coherence* exist in the context of social systems, Koopman operator-based methods have barely been used so far. Some applications involve Koopman mode analysis to investigate the dynamics of spatial-temporal distributions of different agent types [52], or the extraction of non-obvious information from the system state that indicate changes in the dynamics [70]. In our approach, we use the Koopman generator to learn reduced models for complex stochastic agent-based dynamics directly from (highly noisy) data. The goal is the data-based analysis of the reduced models of complex agent-based systems. We demonstrate that for sufficiently large numbers of agents, the data-driven reduced models are in good agreement with the analytically derived limit models, which also appear in Chapter 4, and, moreover, also allow prediction in cases far from the limit or when a limit equation is unknown.

Multi-Objective Optimization of Agent-Based Systems. Finally, in Chapter 6 we show how the reduced models obtained in Chapter 5 can be used to solve multi-objective optimization problems associated with agent-based systems. Since agent-based models often have many different parameters, each of which affects the dynamics of the system in a particular way, our goal is to find optimal combinations of them to achieve a certain target behavior, which then can be interpreted, for example, as finding a campaign strategy. From the mathematical perspective, we are dealing with multi-objective optimization problems, i.e., the simultaneous optimization of multiple, possibly competing objectives; in our case objective functions defined by the dynamical behavior of the agents. It is expected that evaluating objective functions defined in this way, and more generally objectives that depend on experimental or simulation results, will be costly and time-consuming. One option to circumvent this problem is to replace expensive objectives by appropriate surrogate functions, i.e., sufficiently accurate substitutes which often can be evaluated by several orders of magnitude faster. Such surrogates can quickly be determined via interpolation, regression or machine learning [133, 12]. Instead of replacing single objective functions, another option is to replace the full model itself by a surrogate and then use that to solve the optimization problem. Data-driven methods can help to find surrogates, which is especially advantageous when the optimization problem involves dynamical systems. See also [133] for a recent review on surrogate modeling.

Although agent-based models often exhibit a wide range of parameters that determine the dynamic behavior, the full potential of (multi-objective) optimization is not yet re-

alized. As most parameters act on the level of individual agents, it is often unclear how they affect the emergent dynamical behavior such that simple trial-and-error methods, e.g., for model calibration, can be unsatisfying or ineffective or both. Multi-objective optimization is commonly used for a guided or automatic calibration of agent-based systems with the goal that the agents mimic the behavior of a given reference system. The vast majority uses multi-objective evolutionary algorithms to solve the problem as these are simple to use even for complex tasks, see, e.g., [4] and references therein. Examples are the statistical fitting of an agent-based system that models the financial market to real-world market data [144], or of an ant foraging model to biological observations [24]. In addition to fitting, multi-objective optimization can also provide insights (e.g., parameters or initial conditions) beyond those discovered from numerical simulations, such as in [139], where it is applied to a highly complex immunological agent-based model simulating experimental autoimmune encephalomyelitis, a brain inflammation model of laboratory animals comparable to human multiple sclerosis. Other works concerning multi-objective parameter calibration of agent-based models can be found in [104, 122, 138, 90, 35, 10]. For a recent review of calibration techniques see [137].

A different approach in taken [97], where the authors propose to learn surrogate models via non-parametric machine learning by mapping parameter inputs and outputs of agent-based models to obtain computationally cheap surrogates. These are then used for simulation, calibration and parameter space exploration. Although an explicit application to multi-objective optimization is missing, the authors stress out the importance of parameter space exploration, the existence of equally good parameter settings and a higher influence of certain parameters compared to others. In [96], departing from the previous works, an architecture for tunable agents using multi-objective reinforcement learning is proposed that allows them to approximate the properties of different types of agents that do not follow this new tunable regime.

Another application of multi-objective optimization in agent-based modeling is the active support of decision-making processes. In [125], a multi-objective evolutionary algorithm is used to find optimal strategies for contingency planning, with objectives such as minimizing fatalities, population disease state, or hospital waiting time given by the output of an agent-based system. In all papers, model stochasticity and the computational complexity are identified as main difficulties. Besides surrogates based on machine learning [97], surrogate modeling based on difference equations [130], partial differential equations [28], or ordinary differential equations [174] each fitted to an agent-based system are also used to reduce the computational effort during the optimization process. Our approach relies on the data-driven reduced models obtained with the aid of the Koopman generator. Under certain conditions, linear interpolation between Koopman generators is possible, which then can be used to construct surrogate models with varying parameters for agent-based systems to solve associated multi-objective optimization problems.

2 Theoretical Background

The following chapter introduces the mathematical concepts that are relevant to this thesis starting in Section 2.1 with the Koopman and Perron–Frobenius transfer operators, their infinitesimal generators, and their data-driven approximation, followed by an introduction to agent-based models in Section 2.2.

2.1 Transfer Operators

With its introduction in 1931, the Koopman operator theory offers nowadays a popular framework for the analysis of dynamical systems [87, 98, 23]. Instead of analyzing nonlinear but finite-dimensional dynamical systems, Koopman operator theory considers linear but infinite-dimensional operators associated with the dynamical systems, which allows the application of methods developed for linear systems. In this thesis, we will focus on the Koopman operator and its adjoint, the Perron–Frobenius operator for the analysis and model reduction of large-scale agent-based systems.

In the following, we first give an overview of the Koopman operator and its generator as well as their respective adjoints, the Perron–Frobenius operator and generator. Afterwards, we discuss some spectral properties and summarize some numerical approximation methods. The methodology in Section 2.1 follows [82, 84, 85].

2.1.1 Koopman Operator and its Generator

Let $\mathbb{X} \subset \mathbb{R}^d$ denote a (finite-dimensional) state space and let $(T, \mathbb{X}, \Phi)$ denote a dynamical system for a time set T (e.g., $T = \mathbb{R}_{\geq 0}$) and evolution operator $\Phi\colon \mathbb{X} \to \mathbb{X}$. Let $f \in L^\infty(\mathbb{X})$ be a real-valued observable of the system which can, e.g., represent any kind of measurement or sensor probe typically taken in discrete time steps during an experiment. Then, instead of tracing the trajectory of the system $\{x, \Phi(x), \Phi^2(x), \dots\}$ for $x \in \mathbb{X}$, we analyze the evolution of the measurements $\{f(x), f(\Phi(x)), f(\Phi^2(x)), \dots\}$. The Koopman operator $\mathcal{K}\colon L^\infty(\mathbb{X}) \to L^\infty(\mathbb{X})$ describes the evolution of observables and is given by

$$(\mathcal{K}f)(x) = f(\Phi(x)),$$

see [87, 98]. It can naturally be extended to continuous-time dynamical systems [23], which we concentrate on in this thesis.

Deterministic Dynamical Systems

We consider autonomous ordinary differential equations (ODEs) of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}x = b(x) \tag{2.1}$$

for $x \in \mathbb{X}$ and vector field $b\colon \mathbb{R}^d \to \mathbb{R}^d$. The *Koopman semigroup* $\{\mathcal{K}^t\}_{t\geq 0}$ of operators $\mathcal{K}^t\colon L^\infty(\mathbb{X}) \to L^\infty(\mathbb{X})$ is defined as

$$(\mathcal{K}^t f)(x) = f(\Phi^t(x)),$$

where $\Phi^t(x)$ denotes the flow map that takes the initial value problem with initial state $x(0) = x_0$ to the solution $x(t)$ at time t, i.e., $\Phi^t(x_0) = x(t)$. The semigroup property follows from the semigroup property of the flow map, i.e., for $t, s \geq 0$ it holds that

$$(\mathcal{K}^t\mathcal{K}^s f)(x) = (\mathcal{K}^t f) \circ \Phi^s(x) = f \circ \Phi^t \circ \Phi^s(x) = f \circ \Phi^{t+s}(x) = (\mathcal{K}^{t+s} f)(x),$$

where $\circ$ denotes the composition of two functions f and g, see [98, 23, 82]. From now on, we refer to the Koopman semigroup of operators shortly as *Koopman operator*. If f is a vector-valued function, the Koopman operator acts component-wise on f. The Koopman operator is an infinite-dimensional, linear and non-expansive operator, i.e.,

(i) $\mathcal{K}^t(\alpha_1 f_1 + \alpha_2 f_2) = \alpha_1 \mathcal{K}^t f_1 + \alpha_2 \mathcal{K}^t f_2$,

(ii) $\|\mathcal{K}^t f\|_{L^\infty} \leq \|f\|_{L^\infty}$

for all $\alpha_1, \alpha_2 \in \mathbb{R}$ and $f, f_1, f_2 \in L^\infty(\mathbb{X})$. The first property follows directly from the linearity of the composition, i.e.,

$$\mathcal{K}^t(\alpha_1 f_1 + \alpha_2 f_2) = \alpha_1 f_1 \circ \Phi^t + \alpha_2 f_2 \circ \Phi^t = \alpha_1 \mathcal{K}^t f_1 + \alpha_2 \mathcal{K}^t f_2.$$

The second property results from $|f(x)| \leq \|f\|_{L^\infty}$ a.e. which implies $|f(\Phi^t(x))| \leq \|f\|_{L^\infty}$ [87, 98]. The infinitesimal generator $\mathcal{L}$ of the Koopman generator is defined by

$$\mathcal{L}f \coloneqq \lim_{t\to 0} \frac{(\mathcal{K}^t f - f)}{t},$$

and with respect to (2.1) given by

$$\mathcal{L}f = \frac{\mathrm{d}}{\mathrm{d}t} f = b \cdot \nabla_x f = \sum_{i=1}^{d} b_i \frac{\partial f}{\partial x_i},$$

which results from the chain rule applied to $f(x(t))$. Thus, the generator $\mathcal{L}$ is the Lie derivative of f along b for the dynamical system (2.1) and hence also called *Lie operator* [87, 98, 85, 22]. If f is continuously differentiable, then $u(t, x) = \mathcal{K}^t f(x)$ satisfies the first-order partial differential equation $\frac{\partial u}{\partial t} = \mathcal{L}u$, which is known as *Liouville equation* and consequently $\mathcal{K}^t = \exp(t\mathcal{L})$ [85].

Non-deterministic Dynamical Systems

The Koopman operator can be extended to stochastic differential equations (SDEs) of the form

$$\mathrm{d}X_t = b(X_t)\,\mathrm{d}t + \sigma(X_t)\,\mathrm{d}W_t, \tag{2.2}$$

where $\{X_t\}_{t\geq 0} \in \mathbb{X}$ is a time-homogeneous stochastic process and $b\colon \mathbb{R}^d \to \mathbb{R}^d$ denotes the drift term, $\sigma\colon \mathbb{R}^d \to \mathbb{R}^{d\times s}$ the diffusion term, and W_t an s-dimensional Wiener process. A stochastic process $\{X_t\}_{t\geq 0}$ is said to be *time-homogeneous* if the transition probability between two states X_t and X_s for $t \geq s$ only depends on the difference between the initial and final time $t - s$ [131]. Given an SDE of the form (2.2), the stochastic Koopman operator is defined by

$$(\mathcal{K}^t f)(x) = \mathbb{E}[f(\Phi^t(x)) \mid X_t = x], \tag{2.3}$$

where $\mathbb{E}[\,\cdot\,]$ denotes the expectation value. Again, it is an infinite-dimensional, linear and non-expansive operator. Using Itô's lemma and given a twice continuously differentiable function f, the infinitesimal generator of the stochastic Koopman operator with respect to (2.2) can be characterized by

$$\mathcal{L}f = b \cdot \nabla_x f + \frac{1}{2} a : \nabla_x^2 f = \sum_{i=1}^{d} b_i \frac{\partial f}{\partial x_i} + \frac{1}{2} \sum_{i=1}^{d} \sum_{j=1}^{d} a_{ij} \frac{\partial^2 f}{\partial x_i \partial x_j}, \tag{2.4}$$

where $a = \sigma\sigma^\top$ and ∇_x^2 denotes the Hessian, see [72] for details. If f is twice continuously differentiable, then $u(t,x) = \mathcal{K}^t f(x)$ satisfies the second-order partial differential equation $\frac{\partial u}{\partial t} = \mathcal{L}u$, which is called *Kolmogorov backward equation* [114]. For this reason the Koopman operator is also referred to as *backward operator*.

2.1.2 Perron–Frobenius Operator and its Generator

Let us now consider the dual perspective. Given a measure space $(\mathbb{X}, \mathfrak{B}, \mu)$ with σ-algebra $\mathfrak{B}$ and probability measure μ, a map $\Phi^t\colon \mathbb{X} \to \mathbb{X}$ is called *measurable* if $\Phi^{-t}(\mathbb{A}) \in \mathfrak{B}$ for all $\mathbb{A} \in \mathfrak{B}$ and $\Phi^{-t}(\mathbb{A}) \coloneqq \{x \in \mathbb{X} \mid \Phi^t(x) \in \mathbb{A}\}$. It is called *non-singular* if $\mu(\mathbb{A}) \neq 0$ implies $\mu(\Phi^{-t}(\mathbb{A})) \neq 0$ for all $\mathbb{A} \in \mathfrak{B}$. Assume that $\Phi^t\colon \mathbb{X} \to \mathbb{X}$ is a measurable, non-singular map and let $f \in L^1(\mathbb{X})$ with $f \geq 0$ a.e. and $\|f\|_{L^1} = 1$ be the probability density of a random variable x in $\mathbb{X}$, i.e., $x \sim f$ [98, 56, 82]. Non-singularity of Φ^t guarantees that sets with positive measure are not mapped to sets with zero measure [82]. While the Koopman operator describes the evolution of observables, the Perron–Frobenius operator describes evolution of densities.

Deterministic Dynamical Systems

The *Perron–Frobenius semigroup* $\{\mathcal{P}^t\}_{t\geq 0}$ of operators $\mathcal{P}^t\colon L^1(\mathbb{X}) \to L^1(\mathbb{X})$ is defined by

$$\int_{\mathbb{A}} \mathcal{P}^t f(x)\mathrm{d}\mu(dx) = \int_{\Phi^{-t}(\mathbb{A})} f(x)\mathrm{d}\mu(dx)$$

for any measurable set $\mathbb{A} \in \mathfrak{B}$ [98, 56]. As for the Koopman operator, we refer to the Perron–Frobenius semigroup of operators shortly as *Perron–Frobenius operator*. The Perron–Frobenius operator is a Markov operator, i.e., a linear, positive and non-expansive operator with

(i) $\mathcal{P}^t(\alpha_1 f_1 + \alpha_2 f_2) = \alpha_1 \mathcal{P}^t f_1 + \alpha_2 \mathcal{P}^t f_2$,

(ii) $\mathcal{P}^t f \geq 0$ for $f \geq 0$,

(iii) $\|\mathcal{P}^t f\|_{L^1} \leq \|f\|_{L^1}$

for all $\alpha_1, \alpha_2 \in \mathbb{R}$ and $f, f_1, f_2 \in L^1(\mathbb{X})$ [98]. Further, it is the adjoint of the Koopman operator, that is

$$\langle \mathcal{P}^t f, g \rangle_\mu = \langle f, \mathcal{K}^t g \rangle_\mu ,$$

where $\langle f, g \rangle_\mu := \int_{\mathbb{X}} f(x) g(x) \mathrm{d}\mu(x)$ denotes the duality paring between L^1 and L^∞ functions. For invariant measure μ, the operator $\mathcal{P}^t \colon L^p(\mathbb{X}) \to L^p(\mathbb{X})$ is well-defined for every $p \in [1, \infty]$ [11]. Its infinitesimal generator $\mathcal{L}^*$, the adjoint of the Koopman generator $\mathcal{L}$, is given by

$$\mathcal{L}^* f = -\sum_{i=1}^{d} \frac{\partial (b_i f)}{\partial x_i},$$

where the b_i correspond to the right-hand side of (2.1). The generator $\mathcal{L}^*$ is also referred to as *Liouville operator*. The function $u(t, x) = \mathcal{P}^t f(x)$ satisfies the partial differential equation $\frac{\partial u}{\partial t} = \mathcal{L}^* u$, which is known as *continuity equation*, and thus $\mathcal{P}^t = \exp(t\mathcal{L}^*)$ [98, 85, 22].

Non-deterministic Dynamical Systems

In order to define the stochastic Perron–Frobenius operator, we first need the definition of the transition density function. Let $\mathbb{P}[X_t \in \mathbb{A} \mid X_{t_0} = x]$ denote the conditional probability for a stochastic process $\{X_t\}_{t \geq 0}$ to be in $\mathbb{A}$ at time $t \geq t_0$ after starting in $X_{t_0} = x$. For a stochastic process $\{X_t\}_{t \geq 0} \in \mathbb{X}$ the *transition density function* or *transition kernel* $p_t \colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}_{\geq 0}$ is defined by

$$\mathbb{P}[X_t \in \mathbb{A} \mid X_{t_0} = x] := \int_{\mathbb{A}} p_t(x, y) \mathrm{d}y$$

for all $\mathbb{A} \in \mathfrak{B}$ [98, 84]. It can be interpreted as the infinite-dimensional counterpart of the transition matrix of a Markov chain. Then, the stochastic Perron–Frobenius operator is defined by

$$(\mathcal{P}^t f)(x) = \int_{\mathbb{X}} p_t(y, x) f(y) \mathrm{d}\mu(y),$$

see [98, 84]. As before in the deterministic case, it is a Markov operator and for invariant measure μ, the operator $\mathcal{P}^t \colon L^p(\mathbb{X}) \to L^p(\mathbb{X})$ is well-defined for every $p \in [1, \infty]$. The

infinitesimal generator with respect to SDE (2.2), which is the adjoint operator of $\mathcal{L}$, is given by

$$\mathcal{L}^* f = -\sum_{i=1}^{d} \frac{\partial(b_i f)}{\partial x_i} + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial^2(a_{ij} f)}{\partial x_i \partial x_j}.$$

The partial differential equation $\frac{\partial u}{\partial t} = \mathcal{L}^* u$ is called *Fokker–Planck equation* or *Kolmogorov forward equation* and hence the Perron–Frobenius operator is also referred to as *forward* or *evolution operator* [98].

Remark 2.1.1. Reversible stochastic differential equations, precisely reversible with respect to the measure, build an important class of systems. These are systems that satisfy the detailed balance, which means that at the equilibrium state, each elementary process and its reverse are in balance. The Koopman operator extends from $L^\infty_\mu(\mathbb{X})$ to the Hilbert space $L^2_\mu(\mathbb{X})$ with inner product $\langle f, g\rangle_\mu = \int_{\mathbb{X}} f(x)\, g(x)\, \mathrm{d}\mu(x)$ for the process X_t with stationary measure μ [11]. Reversibility is a necessary condition for possessing a stationary measure. These systems can just be described by the diffusion σ and a scalar potential $F\colon \mathbb{R}^d \to \mathbb{R}$. Then, the drift is obtained by

$$b = -\frac{1}{2} a \nabla F + \frac{1}{2} \nabla \cdot a,$$

where the divergence in $\frac{1}{2}\nabla \cdot a$ is applied to each column of a [131]. In this setting both the Koopman operator and the Perron–Frobenius operator are self-adjoint. The Koopman generator becomes self-adjoint and is typically an unbounded operator on a suitable dense subspace of $L^2_\mu(\mathbb{X})$ [84, 85].

2.1.3 Spectral Decomposition and Koopman Modes

The eigenfunctions and eigenvalues of transfer operators contain essential information about the global behavior of dynamical systems. A function φ with corresponding eigenvalue λ is called eigenfunction of the Koopman operator if

$$(\mathcal{K}^t \varphi)(x) = \exp(\lambda t)\varphi(x), \tag{2.5}$$

which is equivalent to

$$(\mathcal{L}\varphi)(x) = \lambda \varphi(x).$$

This means also that eigenfunctions of the Koopman operator are also eigenfunctions of the Koopman generator. In general, eigenvalues and eigenfunctions are complex even if the state space and the dynamics are real-valued. If φ_i and φ_j are eigenfunctions with associated eigenvalues λ_i and λ_j, then also $\varphi_i \varphi_j$ with eigenvalue $\lambda_i + \lambda_j$ as

$$(\mathcal{K}^t \varphi_i \varphi_j)(x) = \exp((\lambda_i + \lambda_j)t)\, \varphi_i \varphi_j(x)$$

holds due to the linearity of $\mathcal{K}^t$. As the Perron–Frobenius operator is the adjoint of the Koopman operator, both share the same spectrum. Their eigenfunctions, however, differ [115, 82, 84].

Remark 2.1.2. In some cases, eigenvalues are defined as follows. We call λ^t an eigenvalue of $\mathcal{K}^t$ with corresponding eigenfunction φ if

$$(\mathcal{K}^t\varphi)(x) = \lambda^t\varphi(x).$$

Obviously, both definitions are related via the exponential function, i.e., $\lambda_\ell^t = \exp(-\lambda_\ell t)$, where λ_ℓ denotes an eigenvalue defined according to (2.5). The definition holds analogously for the Perron–Frobenius operator. Eigenvalues as defined in this remark can be more conveniently in some cases, e.g., in the case of *invariant densities*, which are defined as eigenfunctions satisfying $\mathcal{P}^t\rho = \rho$ for all $t \geq 0$. The corresponding eigenvalue is given by $\lambda^t = 1$ respectively $\lambda = 0$.

Example 2.1.3. Consider the dynamical system

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}x_1\\x_2\end{bmatrix} = \begin{bmatrix}\gamma\, x_1\\ \delta\,(x_2 - x_1^2)\end{bmatrix}. \tag{2.6}$$

Beside the trivial eigenfunction $\varphi_1(x) = 1$ with generator eigenvalue $\lambda_1 = 0$, also $\varphi_2(x) = x_1$ and $\varphi_3(x) = \frac{2\gamma-\delta}{\delta}x_2 + x_1^2$ with corresponding generator eigenvalues $\lambda_2 = \gamma$ and $\lambda_3 = \delta$, respectively, are eigenfunctions of the Koopman generator [21]. Further eigenfunctions with corresponding generator eigenvalues can be obtained via multiplication, e.g.,

$$\begin{aligned}
&\lambda_4 = 2\,\delta, && \varphi_4(x) = \left(\tfrac{2\gamma-\delta}{\delta}\right)^2 x_2^2 + 2\,\tfrac{2\gamma-\delta}{\delta}\,x_1^2 x_2 + x_1^4 = \varphi_3(x)^2,\\
&\lambda_5 = \gamma + \delta, && \varphi_5(x) = \tfrac{2\gamma-\delta}{\delta}x_1x_2 + x_1^3 = \varphi_2(x)\,\varphi_3(x).
\end{aligned}$$

△

Koopman Modes

Let $f\colon \mathbb{X} \to \mathbb{R}$ be an observables of the system and φ_ℓ denote a linearly independent eigenfunction. Then,

$$f(x) = \sum_{\ell=1}^{\infty} v_{j\ell}\,\varphi_\ell(x)$$

which extends for vector-valued $f = [f_1, \ldots, f_n]^\top$ to

$$f(x) = \sum_{\ell=1}^{\infty} v_\ell\,\varphi_\ell(x),$$

where $v_\ell = [v_{1\ell}, \ldots, v_{n\ell}]^\top \in \mathbb{C}^n$. The vector v_ℓ is called *Koopman mode*. An important class of observables are the so-called *full-state observables* $g(x) = x$ for which we have

$$(\mathcal{K}^t g)(x) = (g \circ \Phi^t)(x) = \Phi^t(x).$$

With the help of the Koopman eigenvalues λ_ℓ, eigenfunctions φ_ℓ and modes v_ℓ we can use the Koopman operator to propagate the state of the system [171], i.e.,

$$(\mathcal{K}^t g)(x) = \sum_{\ell=1}^{\infty} (\mathcal{K}^t \varphi_\ell)(x)\, v_\ell = \sum_{\ell=1}^{\infty} \exp(\lambda_\ell t)\, \varphi_\ell(x) v_\ell.$$

Remark 2.1.4. We obtain an analogous expression for the Perron–Frobenius operator. Further, if we assume that the dynamics are reversible and let λ_ℓ denote the eigenvalues of the operator $\mathcal{P}^t$, sorted by decreasing value, it holds that $0 = \lambda_1 > \lambda_2 \geq \dots$ and the eigenvalues satisfy

$$\lim_{\ell \to \infty} |\exp(\lambda_\ell t)| = 0 \quad \text{and} \quad \lim_{t \to \infty} |\exp(\lambda_\ell t)| = 0.$$

Hence, for a given lag time $\tau > 0$ there exists an index $\tilde{d} \in \mathbb{N}$ such that $\exp(\lambda_\ell t) \approx 0$ for all $t \geq \tau$ and all $\ell > \tilde{d}$. Then, for a given function f and eigenfunctions φ_ℓ with $f(x) = \sum_{\ell=0}^{\infty} \varphi_\ell(x)\, \beta_\ell$ for $\beta_\ell \in \mathbb{R}$, we obtain

$$(\mathcal{P}^t f)(x) = \sum_{l=1}^{\infty} \exp(\lambda_\ell t)\, \varphi_\ell(x)\, \beta_\ell \approx \sum_{l=1}^{\tilde{d}} \exp(\lambda_\ell t)\, \varphi_\ell(x)\, \beta_\ell \quad \text{for all} \quad t \geq \tau.$$

This means that a large part of information about the long-term density propagation of the dynamics is contained in $\tilde{d}$ dominant eigenpairs. The associated time scales are called *dominant time scales* and given by $T_\ell = -\lambda_\ell^{-1}$ for $\ell = 2, \dots, \tilde{d}$, see [152, 127] for details.

Example 2.1.5. The one-dimensional Ornstein–Uhlenbeck process is given by the SDE

$$\mathrm{d}X_t = -\alpha X_t \,\mathrm{d}t + \sqrt{2\beta^{-1}}\,\mathrm{d}W_t$$

with inverse temperature β and friction coefficient α. Using recurrence relations for the Hermite polynomials, i.e., $H_{\ell+1}(x) = xH_\ell(x) - H'_\ell(x)$, the eigenfunctions φ_ℓ and eigenvalues λ_ℓ are given by

$$\varphi_\ell(x) = \frac{1}{\sqrt{(\ell-1)!}} H_{\ell-1}\left(\sqrt{\alpha\beta}\, x\right), \quad \lambda_\ell = -\alpha\,(\ell - 1), \quad \ell = 1, 2, \dots$$

where H_ℓ denotes the ℓth probabilists' Hermite polynomial [131]. The Perron–Frobenius eigenfunction can be obtained from the Koopman eigenfunctions by multiplying them with the invariant density ρ given by

$$\rho(x) = \frac{1}{\sqrt{2\pi\alpha^{-1}\beta^{-1}}} \exp\left(-\alpha\,\beta\,\frac{x^2}{2}\right).$$

It is a Gaussian with decreasing variance for increasing friction and decreasing temperature. The Ornstein–Uhlenbeck process is reversible and the generator is self-adjoint in the space $L^2(\rho)$ weighted by the invariant density [131, 85]. △

Remark 2.1.6. In this work we concentrate on discrete point spectra of Koopman operators though systems of high complexity, e.g., chaotic or measure-preserving, (non-dissipative) dynamical systems, give rise to non-compactness or continuous spectra. Their numerical analysis is discussed in detail in [60]. For further details on the Koopman operator spectrum see, e.g., [115, 34, 116].

2.1.4 Numerical Approximation Methods

As the Koopman and Perron–Frobenius operator are infinite-dimensional operators, we cannot handle them numerically in a direct way but often consider projections onto finite-dimensional subspaces. However, there are a few cases where it is possible to construct a linear and finite-dimensional representation of a nonlinear dynamical system as demonstrated in the next example.

Example 2.1.7. Consider again the dynamical system (2.6) and choose $f_1 = x_1$, $f_2 = x_2$ and $f_3 = x_1^2$ as observables. Then, we can write the system as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \underbrace{\begin{bmatrix} \gamma & 0 & 0 \\ 0 & \delta & -\gamma \\ 0 & 0 & 2\gamma \end{bmatrix}}_{L} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

and the dynamics receive a linear representation with L as a finite-dimensional approximation of the Koopman generator. △

In general we cannot find such a finite-dimensional representation. In the following, we consider the projection of transfer operators onto finite-dimensional subspaces.

Galerkin Approximation of the Koopman Generator

Let $\mathbb{H}$ be a Hilbert space and $\mathbb{L} \subset \mathbb{H}$ finite-dimensional subspace with basis $\{\psi_i\}_{i=1}^n$. Then, the unique linear operator $A\colon \mathbb{L} \to \mathbb{L}$ with

$$\langle \psi_j, \mathcal{A}\psi_i \rangle = \langle \psi_j, A\psi_i \rangle$$

for all $i, j = 1, \ldots, n$ is called *Galerkin projection* of operator $\mathcal{A}$ to $\mathbb{L}$ [82]. Assume that $\psi_i\colon \mathbb{R}^d \to \mathbb{R}$ for all basis functions $\{\psi_i\}_{i=1}^n$. Then, we can compute a Galerkin approximation $\mathbf{L}$ of the generator $\mathcal{L}$ with the help of the matrices $A, G \in \mathbb{R}^{n\times n}$ with

$$\begin{aligned} A_{ij} &= \langle \mathcal{L}\psi_i, \psi_j \rangle_\mu, \\ G_{ij} &= \langle \psi_i, \psi_j \rangle_\mu \end{aligned} \tag{2.7}$$

for a given measure μ. We obtain the matrix representation L of the projected operator $\mathbf{L}$ by $L^\top = AG^{-1}$. Given a function

$$f(x) = \sum_{i=1}^{n} c_i \psi_i(x) = c^\top \psi(x)$$

for $\psi(x) = [\psi_1(x), \ldots, \psi_n(x)]^\top$ and $c = [c_1, \ldots, c_n]^\top \in \mathbb{R}^n$, we get $(\mathbf{L}f)(x) = (Lc)^\top \psi(x)$. Defining $\varphi_\ell(x) = \xi_\ell^\top \psi(x)$ we get

$$(\mathbf{L}\varphi_\ell)(x) = (L\xi_\ell)^\top \psi(x) = \lambda_\ell \xi_\ell^\top \psi(x) = \lambda_\ell \varphi_\ell(x),$$

meaning that the coefficients for the eigenfunctions of $\mathbf{L}$ are contained in the eigenvector ξ_ℓ of L with corresponding eigenvalue λ_ℓ. Note that in general the projected generator does not result in a rate matrix [85].

Example 2.1.8. For the Ornstein–Uhlenbeck process it is possible to compute the matrix L analytically. Assume that we have a basis consisting of monomials up to degree $n-1$, i.e., $\psi(x) = [1, x, \ldots, x^{n-1}]^\top$. Note that $\mathcal{L}\psi_k$ is contained in the subspace spanned by $\{\psi_i\}_{i=1}^n$ and, moreover, for $k \geq 3$,

$$(\mathcal{L}\psi_k)(x) = -\alpha(k-1)x^{k-1} + \beta^{-1}(k-1)(k-2)x^{k-3}.$$

The matrix $L \in \mathbb{R}^{n\times n}$ is is then given by

$$\begin{array}{c} \\ 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \\ \vdots \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1 & x & x^2 & x^3 & x^4 & x^5 & x^6 & \ldots \end{array} \\
\left[\begin{array}{ccccccccc}
0 & & 2\beta^{-1} & & & & & & \\
& -\alpha & & 6\beta^{-1} & & & & & \\
& & -2\alpha & & 12\beta^{-1} & & & & \\
& & & -3\alpha & & 20\beta^{-1} & & & \\
& & & & -4\alpha & & 30\beta^{-1} & & \\
& & & & & -5\alpha & & \ddots & \\
& & & & & & -6\alpha & & \\
& & & & & & & \ddots &
\end{array}\right]
\end{array}.$$

For the better understanding the rows and columns are labeled according to their correspondence to the basis functions. As seen in Example 2.1.5, we verify that the eigenvalues are given by $\lambda_\ell = -\alpha(\ell - 1)$, for $\ell = 1, \ldots, n$, and that the coefficients of the eigenfunctions are given by the eigenvectors [85]. △

Further details on Galerkin discretizations of transfer operators and their properties can be found in, e.g., [153, 154, 82, 84]. In general the required integrals cannot be computed analytically, so that instead we estimate them from data via, e.g., Monte Carlo integration. The following algorithms are common ways to compute approximations of projected transfer operators.

Extended Dynamic Mode Decomposition

Extended dynamic mode decomposition (EDMD) [171] is an algorithm that was developed to compute finite-dimensional approximations of the Koopman operator, its eigenfunctions, eigenvalues and modes. Let $x_i, y_i \in \mathbb{R}^d$, $i = 1, \ldots, m$ be pairs of d-dimensional data vectors such that $y_i = \Phi^{t_i}(x_i)$ for $t_i > 0$. The data vectors can be

obtained, e.g., from measurements or black-box simulations. Additionally, EDMD requires a set of uniformly bounded basis functions or observables, also called *dictionary*, $\{\psi_i\}_{i=1}^n \subset L^\infty(\mathbb{X})$. Further, we have that $\{\psi_i\}_{i=1}^n \subset L^r(\mathbb{X})$ for $1 \leq r \leq \infty$ as the state space $\mathbb{X} \subset \mathbb{R}^d$ is assumed to be bounded. Having this we can set up the transformed data matrices $\Psi_X, \Psi_Y \in \mathbb{R}^{n\times m}$ such that

$$\Psi_X = \begin{bmatrix} \psi_1(x_1) & \dots & \psi_1(x_m) \\ \vdots & \ddots & \vdots \\ \psi_n(x_1) & \dots & \psi_n(x_m) \end{bmatrix} \quad \text{and} \quad \Psi_Y = \begin{bmatrix} \psi_1(y_1) & \dots & \psi_1(y_m) \\ \vdots & \ddots & \vdots \\ \psi_n(y_1) & \dots & \psi_n(y_m) \end{bmatrix}$$

and we assume that there exists a matrix M such that $\Psi_Y = M\Psi_X$. In general we cannot solve this equation exactly as it is usually overdetermined. Therefore, we solve the minimization problem

$$\min \|\Psi_Y - M\Psi_X\|_F, \tag{2.8}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The matrix $K = M^\top$ represents a finite-dimensional approximation of the Koopman operator $\mathcal{K}^t$ with respect to the basis spanned by $\{\psi_i\}_{i=1}^n$. The minimization problem (2.8) can be solved analytically and its solution is given by

$$M = \Psi_Y \Psi_X^+ = (\Psi_Y \Psi_X^\top)(\Psi_X \Psi_X^\top)^+ = \widehat{A}\widehat{G}^+$$

for matrices

$$\widehat{A} = \frac{1}{m}\sum_{l=1}^m \psi(y_l)\,\psi(x_l)^\top \quad \text{and} \quad \widehat{G} = \frac{1}{m}\sum_{l=1}^m \psi(x_l)\,\psi(x_l)^\top.$$

Here, A^+ denotes the Moore–Penrose pseudoinverse of a matrix A. The eigenfunctions of the Koopman operator are then given by the left eigenvectors and eigenvalues of $M = \widehat{A}\widehat{G}^+$. For large amounts of data points, that is, $m \to \infty$, it has been shown in [171, 82] that EDMD converges to a Galerkin approximation. For $\widehat{A} = (\widehat{A}_{ij})_{i,j=1,\dots,n}$ and $\widehat{G} = (\widehat{G}_{ij})_{i,j=1,\dots,n}$ it holds

$$\begin{aligned} \widehat{A}_{ij} &= \frac{1}{m}\sum_{l=1}^m \psi_i(y_l)\,\psi_j(x_l) \underset{m\to\infty}{\longrightarrow} \int (\mathcal{K}^t\psi_i)(x)\,\psi_j(x)\,\mathrm{d}\mu(x) = \left\langle \mathcal{K}^t\psi_i, \psi_j \right\rangle_\mu = A_{ij}, \\ \widehat{G}_{ij} &= \frac{1}{m}\sum_{l=1}^m \psi_i(x_l)\,\psi_j(x_l) \underset{m\to\infty}{\longrightarrow} \int \psi_i(x)\,\psi_j(x)\,\mathrm{d}\mu(x) \qquad = \left\langle \psi_i, \psi_j \right\rangle_\mu \qquad = G_{ij}, \end{aligned}$$

where $x_l \sim \mu$ and $\langle f, g\rangle_\mu \coloneqq \int_{\mathbb{X}} f(x)g^*(x)\mathrm{d}\mu(x)$.

The Koopman modes are obtained as follows. Let ξ_ℓ be the ℓth eigenvector of M and set $\Xi \coloneqq \begin{bmatrix}\xi_1 & \dots & \xi_n\end{bmatrix}$. Define $\varphi(x) \coloneqq [\varphi_1(x), \dots, \varphi_n(x)]^\top$ and assume $B \in \mathbb{R}^{n\times d}$ such that $x = g(x) = B^\top\psi(x) = \Xi^\top\psi(x)$. Then

$$g(x) = B^\top\psi(x) = B^\top\Xi^{-\top}\varphi(x)$$

and we obtain the Koopman modes v_ℓ from the column vectors $V = B^\top \Xi^{-\top}$ that are required to reconstruct the dynamics.

As shown in [82] EDMD can also be used to approximate the Perron–Frobenius operator with respect to the density of the data points. Exploiting duality of $\mathcal{K}^t$ and $\mathcal{P}^t$ yields

$$M_{\mathrm{PF}} = (\Psi_X \Psi_Y^\top)(\Psi_X \Psi_X^\top)^+ = \widehat{A}^\top \widehat{G}^+.$$

The eigenfunctions of the Perron–Frobenius operator are approximated by the left eigenvectors and eigenvalues of $M_{\mathrm{PF}} = \widehat{A}^\top \widehat{G}^+$.

Remark 2.1.9. *Dynamic mode decomposition* (DMD) [150] was developed in the fluid dynamics community as algorithm to identify coherent structures in fluid flows from time-series data and is nowadays a common algorithm, see [27, 163, 81, 95] and references therein. In this work, we summarize the *exact DMD* [163]. As before for EDMD, for m given data vectors $y_i = \Phi^{t_i}(x_i)$, DMD assumes that there exists a matrix M such that $Y = MX$ for the data matrices

$$X = \begin{bmatrix} x_1 & \dots & x_m \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} y_1 & \dots & y_m \end{bmatrix}.$$

Again, this equation in general cannot be solved exactly. Instead we solve

$$\min \|Y - MX\|_F. \tag{2.9}$$

The DMD eigenvalues and modes are given by the eigenvalues and eigenvectors of M, respectively. As the name suggests, (exact) DMD is a special case of EDMD. Let the set of observables be such that $\psi(x) = x$, i.e., the set of observables consists of linear function only. Let M_{EDMD} and M_{DMD} be the matrices that minimize (2.8) and (2.9), respectively. Then

$$M_{\mathrm{EDMD}} = \Psi_Y \Psi_X^+ = Y X^+ = M_{\mathrm{DMD}},$$

which means that the DMD algorithm approximates the Koopman operator using linear functions only. The Koopman modes correspond with the DMD modes in this case [82].

Remark 2.1.10. The choice of basis function depends on the problem and could, e.g., comprise monomials, radial basis functions, Hermite or Legendre polynomials. The optimal choice of basis functions, however, is still an open question. On the one hand the set of basis functions should be large enough to represent the system's dynamics, running the risk of ill-conditioned matrices and overfitting. On the other hand, if the set of basis functions is not sufficient or inappropriately chosen, the results will be inaccurate. We will discuss what can be done in such a case in Chapter 5. For further discussions on the choice of basis function see, e.g., [171].

Ulam's Method

One of the most common methods to compute finite-dimensional approximations of the Perron–Frobenius operator $\mathcal{P}^t$ is called *Ulam's method* or *box discretization*, see, e.g.,

[164, 151, 147, 55]. Here, we present the box discretization of the stochastic Perron–Frobenius operator. For the deterministic case we refer to [82]. Let $B_i \subset \mathbb{X}$, $i = 1, \dots, n$, denote finitely many disjoint subsets such that they from a complete, non-overlapping partition of the state space $\mathbb{X}$, i.e., $\bigcup_{i=1}^{n} B_i = \mathbb{X}$ with $B_i \cap B_j = \emptyset$ for $i \neq j$. The subsets B_i are called *boxes*. Let $\mathbb{1}_{B_i}$ denote the indicator function of subset B_i, i.e., $\mathbb{1}_{B_i} \colon \mathbb{X} \to \mathbb{R}$,

$$\mathbb{1}_{B_i}(x) = \begin{cases} 1, & \text{if } x \in B_i, \\ 0, & \text{otherwise.} \end{cases}$$

Let $\mathbb{L} \subset L^2(\mathbb{X})$ be a finite dimensional subspace spanned by the indicator functions, that is $\psi_i = \mathbb{1}_{B_i}$, $i = 1, \dots, n$. Then, the projected transfer operator with respect to $\mathbb{L}$ is the operator $Q\mathcal{P}^t Q$, where $Q \colon L^2_\mu(\mathbb{X}) \to \mathbb{L}$ defined by

$$Qu = \sum_{i,j=1}^{m} G^{-1}_{ij} \langle \psi_i, u \rangle_\mu \psi_j \quad \text{with mass matrix} \quad G_{ij} = \langle \psi_i, \psi_j \rangle_\mu$$

denotes the Galerkin projection of the original function space of $\mathcal{P}^t$ to $\mathbb{L}$ with respect to the scalar product $\langle \cdot, \cdot \rangle_\mu$. The matrix representation P^t of the projected Perron–Frobenius operator is given by

$$p^t_{ij} = \mathbb{P}_\mu\Big(\Phi^t(x) \in B_j \mid x \in B_i\Big) = \frac{1}{\mu(B_i)} \mathbb{E}_\mu\Big(\mathbb{1}_j(\Phi^t(x)) \mathbb{1}_i(x\Big)$$

with initial values distributed according to μ and $i, j = 1, \dots, n$. The matrix P^t is row-stochastic. Its entries can be estimated by counting how many times points $\Phi^t(x_i^l)$ are contained in box B_j when starting in B_i, i.e., $x_i^l \in B_i$ for $l = 1, \dots, m_0$. That is

$$p^t_{ij} \approx \frac{1}{m_0} \sum_{k=1}^{m_0} \mathbb{1}_j(\Phi^t(x_i^l)) \tag{2.10}$$

for mutually independent realizations of $\Phi^t(x_i^l)$, see, e.g., [151, 147, 82, 127]. A wide range of works contributes on how well dominant eigenvalues of the original transfer operator $\mathcal{P}^t$ are approximated by the eigenvalues of the projected transfer operator $Q\mathcal{P}^t Q$, such as in [38], where the error is characterized via the projection error induced by Q, or in [153], where upper and lower error bounds for every complete partition of the respective state space into sets are established. In [147] the long-term error $\|\mathcal{P}^{st} - (Q\mathcal{P}^t Q)^s\|$ is studied for $s \gg 1$.

Remark 2.1.11. Choosing indicator functions as observables, i.e., $\{\mathbb{1}_{B_i}\}_{i=1}^n$, and m_0 data points per box, then EDMD computes the same finite-dimensional representation as Ulam's method. In fact, the matrix Ψ_X is given by

$$\Psi_X = \operatorname{diag}\left(\mathbf{1}_{m_0}^\top, \dots, \mathbf{1}_{m_0}^\top\right) \in \mathbb{R}^{n \times n m_0},$$

where $\mathbf{1}_{m_0} = [1, \dots, 1]^\top$ denotes a vector of length m_0 and $\mathrm{diag}(d_1, \dots, d_n)$ a diagonal matrix with entries d_i; here, vectors of length m_0. It holds that matrix $K = M^\top$, which results from the minimization problem (2.8), has the same entries as P^t with entries defined by (2.10), i.e., $K = P^t$. For further details we refer to [82].

2.2 Agent-Based Models

To motivate what follows let us first consider a system of N agents interacting with each other in an *interaction network* and assume that each agent can change between d different *states* according to *transition rules*. The state of an agent can be interpreted, e.g., as health state, opinion or any kind of discrete property. It is also denoted as *type*, *breed*, *choice*, *species* or *strategy* depending on the application and community. Note that in some applications the state may not be changeable for logical reasons, e.g., when modeling different species. The full state of the agent-based model (ABM) is given by $\{1, \dots, d\}^N$ and grows like d^N, which imposes computationally problems when the number of agents N is large. Alternatively, we can describe the model via the population state, which can be favorably when we are not interested in each agent's state but in the collective behavior of larger groups or the entire population. The population state is given by the numbers of agents of each type and grows like N^d in worst case scenario, which makes it computationally more appealing than the full ABM state. For indistinguishable agents and random interactions, e.g., enabled via a complete interaction network, the population state description is exact and transition rules between types imply transition rules between population states. In all other cases the aggregation of the full ABM state space involves an approximation error [127].

In this thesis we consider three different agent-based systems of increasing complexity as guiding examples. The first system is a continuous-time voter model, where agents interact with each other in a network and change their opinion, i.e., their state, based on transition rules. Different interaction networks are considered. The second model is a discrete-time, spatial predator-prey model, where agents move randomly in a given domain. Differently from the voter model, where changes in the population state are due to changing numbers of agents sharing one opinion, for the predator-prey model changes result as a consequence of reproduction and death. That is, the state of an agent is immutable. The third agent-based system models social violence and was originally proposed by Epstein [45]. Depending on a dynamically changing neighborhood and individual parameters, agents switch between peaceful and violent states. A central authority tries to suppress a rebellion among the population by arresting insurgents. This model is the most complex in this work as agents are heterogeneous with respect to their parameters and movement is not necessarily purely randomly.

We will now introduce the representation of agent-based systems as Markov jump process and their approximation for large numbers of agents in terms of ordinary and stochastic differential equations. Afterwards, we will present the three guiding examples. The methodology in Section 2.2 follows [126, 127].

2.2.1 Agent-Based Models as Markov Jump Processes

Agent-based modeling is a bottom-up technique, meaning that the collective dynamical behavior and global properties result from the definitions at agent-level. We use a formalism commonly used in chemical contexts that focuses on the population state, resulting in a clear and transparent description. Consider an agent-based system consisting of

(i) a fixed number $N \in \mathbb{N}$ of agents,

(ii) a set $\{S_i\}_{i=1}^{d}$ of *types* available to the agents,

(iii) a set $\{R_k\}_{k=1}^{K}$ of *transition rules* defining possible changes between the agents' types,

(iv) a set $\{\alpha_k\}_{k=1}^{K}$ of *propensity functions* specifying the rates at which transitions randomly occur,

and assume that the interaction network is complete such that the population state representation is exact. The latter assumption simplifies the following. Nevertheless, generalizations, e.g., by weighted interaction networks or by weakly connected, complete components or clusters, are possible. These affect the propensity functions and the population state space, which then additionally depend on the neighborhood of each agent.

At any point in time, each agent is assigned a type S_i and the *population state* $x \in \mathbb{X}_N$ of the ABM is completely described by the vector

$$x = [x_1, \dots, x_d]^\top \in \mathbb{N}_0^d,$$

where x_i denotes the number of agents of type S_i. The *population state space* $\mathbb{X}_N$ is given by the $d-1$-dimensional simplex

$$\mathbb{X}_N := \left\{ x \in \mathbb{N}_0^d \mid \textstyle\sum_{i=1}^{d} x_i = N \right\}. \tag{2.11}$$

The index N emphasizes the fact that $\mathbb{X}_N$ is a discrete space. This description is equivalent to the partition of the space of all possible configurations in [9]. Transition rules R_k that are mostly described from the viewpoint of each agent define feasible changes of the type S_i via actions or interactions. They are represented by a reaction equation of the form

$$R_k\colon \quad a_{1k}\,S_1 + \ldots + a_{dk}\,S_d \quad \mapsto \quad b_{1k}\,S_1 + \ldots + b_{dk}\,S_d,$$

where the coefficients a_{lk}, $b_{lk} \in \mathbb{N}_0$ denote the numbers of agents of each type that are involved in the transition R_k. To ensure that the number of agents N remains constant in time, we assume that $\sum_{i=1}^{d} a_{ik} = \sum_{i=1}^{d} b_{ik}$ for each $k = 1, \dots, K$. Each transition rule induces an immediate change in the population state of the form

$$x \mapsto x + \nu_k,$$

where the vector $\nu_k = [\nu_{1k}, \dots, \nu_{dk}]^\top$, defined by $\nu_{ik} := b_{ik} - a_{ik}$, describes the *net change* in the number of agents of type S_i due to transition R_k. Note that distinct

transitions $R_k \neq R_l$ can have identical net change vectors $\nu_k = \nu_l$; more on this in Section 2.2.3. The rates of occurrence for each transition R_k are determined by the *propensity functions* $\alpha_k^N \colon \mathbb{X} \to [0, \infty)$, which are assumed to be proportional to the number of combinations of agents in x, and, moreover, scale with the number of agents N, i.e.,

$$\alpha_k^N(x) = \begin{cases} \gamma_k N \prod_{i=1}^{d} \frac{1}{N^{a_{ik}}} \binom{x_i}{a_{ik}}, & \text{if } x_i \geq a_{ik} \text{ for all } i = 1, \ldots, d, \\ 0, & \text{otherwise.} \end{cases}$$

Here, $\gamma_k > 0$ denotes the *rate constant* for the kth transition R_k. The probability for transition R_k is then given by $\alpha_k^N(x)\,\mathrm{d}t$ for an infinitesimal time step $\mathrm{d}t$ [126, 127].

Temporal Evolution

The temporal evolution of the population state can be described as continuous-time, discrete-space stochastic process

$$\{X_t^N\}_{t \geq 0} \quad \text{with} \quad X_t^N = [x_1(t), \ldots, x_d(t)]^\top \in \mathbb{X}_N \tag{2.12}$$

and randomly occurring jumps of the form $X_t^N \mapsto X_t^N + \nu_k$, where $x_i(t)$ denotes the number of agents of type S_i at time $t \geq 0$. It is a piece-wise constant process, i.e., a *Markov jump process*. We are shortly referring to (2.12) as *ABM process*. Given the current state $X_t = x$ of the process at time t, the waiting time $\tau(x)$ until the next jump occurs is exponentially distributed with mean $\lambda(x)^{-1}$, where $\lambda \colon \mathbb{X}_N \to [0, \infty)$ denotes the *jump rate function* and is defined by

$$\lambda(x) \coloneqq \sum_{k=1}^{K} \alpha_k^N(x).$$

At time $t + \tau(x)$ the process jumps according to one transition R_k to the new state $x + \nu_k$ with probability $\alpha_k^N(x)\,\lambda(x)^{-1}$. Let $P(x, t)$ be the conditional probability of finding the process in state x at time t after starting in x_0, i.e.,

$$P(x, t) \coloneqq \mathbb{P}\left[X_t^N = x \mid X_0^N = x_0\right].$$

Then, the temporal evolution of the ABM can be described in two ways using $P(x, t)$. The first option is to characterize the evolution of the probability using the *Kolmogorov forward equation*

$$\frac{\mathrm{d}}{\mathrm{d}t} P(x, t) = \sum_{k=1}^{K} \left[\alpha_k^N(x - \nu_k) P(x - \nu_k, t) - \alpha_k^N(x) P(x, t)\right], \tag{2.13}$$

where we set $\alpha_k^N(x) \coloneqq 0$ and $P(x, t) \coloneqq 0$ for $x \notin \mathbb{N}_0^d$. The equation (2.13) is also known as *chemical master equation* [62]. Alternatively, we can use the Perron–Frobenius

operator $\mathcal{P}_N^t$, which is given by $\mathcal{P}_N^t = \exp(t\mathcal{G}_N)$, where $\mathcal{G}_N$ denotes the right-hand side of (2.13). To simplify the notation we denote this specific generator by $\mathcal{G}_N$ instead of $\mathcal{L}_N^*$. The index N indicates for which number of agents the dynamics are considered. It acts on functions $f \in L_\pi^2(\mathbb{X})$ of x via

$$(\mathcal{P}_N^t f)(y) = \sum_{x \in \mathbb{X}_N} f(x) \mathbb{P}\left[X_t^N = y \mid X_0^N = x\right].$$

As the master equation (2.13) cannot be solved analytically, the distribution of the ABM process is usually estimated by Monte Carlo simulations generated using Gillespie's stochastic simulation algorithm [61]. It constructs exact realizations of the ABM process in continuous-time and thus naturally circumvents the scheduling problem of ABMs that arises in discrete-time. However, as the number of agents N increases, numerical computation using the Gillespie algorithm becomes inefficient, as the waiting time between each jump decreases and thus jumps occur more frequently, resulting in smaller progress in each iteration. Additionally, the influence of a single agent vanishes since the size of an individual jump with respect to N decreases. Approximations of the population state for large numbers of agents N based on ODEs or SDEs mitigate this problem as their time iterations are in general independent of N.

2.2.2 Population Limits for Finite Time Intervals

Defining the relative frequency $c \coloneqq x/N$ and introducing the so-called *smallness parameter* $\varepsilon \coloneqq 1/N$, we can rewrite the master equation (2.13) of the ABM process in terms of the frequency-based probability distribution

$$\rho^N(c,t) \coloneqq \mathbb{P}\left[X_t^N = Nc\right] = P(Nc,t),$$

with scaled propensities

$$\tilde{\alpha}_k^\varepsilon(c) \coloneqq N^{-1}\alpha_k^N(Nc).$$

The rescaled master equation is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho^\varepsilon(c,t) = \frac{1}{\varepsilon}\sum_{k=1}^K \left[\tilde{\alpha}_k^\varepsilon(c - \varepsilon\boldsymbol{\nu}_k)\,\rho^\varepsilon(c-\varepsilon\boldsymbol{\nu}_k,t) - \tilde{\alpha}_k^\varepsilon(c)\,\rho^\varepsilon(c,t)\right] \tag{2.14}$$

and the corresponding transfer operator is denoted by $\widetilde{\mathcal{P}}_N^t$. It is generated by $\widetilde{\mathcal{P}}_N^t = \exp(t\widetilde{\mathcal{G}}_N)$, where $\widetilde{\mathcal{G}}_N$ denotes the generator of the rescaled master equation (2.14). It depends on N and acts on functions $f \in L_\pi^2(\mathbb{X}/N)$ on the rescaled population space $\mathbb{X}/N$. Assuming convergence of the rescaled propensity functions $\tilde{\alpha}_k^\varepsilon \to \tilde{\alpha}_k$ for $\varepsilon \to 0$, it has been shown in the 1970s by Kurtz [94] that for $N \to \infty$ the rescaled process, whose distribution follows the master equation (2.14), converges to the frequency process $\{C_t\}_{t\geq 0}$ given by the ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}C_t = \sum_{k=1}^K \tilde{\alpha}_k(C_t)\,\nu_k \tag{2.15}$$

with $C_0 = \lim_{N\to\infty} \frac{X_0^N}{N}$. We refer to (2.15) as *ODE limit model*. It is a pathwise approximation

$$\sup_{t\in[0,T]} \|X_t^N/N - C_t\| \leq \zeta N^{-1/2}$$

for every finite T and a.s. finite constant ζ, both independent of N, where ζ will in general depend on T [94, 47]. Here, $\|\cdot\|$ refers to any norm on $\mathbb{R}^d$. For the first order moment of X_t^N/N it holds that

$$\mathbb{E}\left[X_t^N/N\right] = C_t + \mathcal{O}(N^{-1})$$

for all $t \geq 0$ [91, 92]. A higher order approximation, which is additionally capable of reproducing fluctuations and thus approximating the second order moment, is given by the SDE

$$\mathrm{d}C_t = \sum_{k=1}^{K} \tilde{\alpha}_k(C_t)\,\nu_k\,\mathrm{d}t + \sum_{k=1}^{K} \frac{1}{\sqrt{N}} \sqrt{\tilde{\alpha}_k(C_t)}\,\mathrm{d}W_k(t)\,\nu_k \tag{2.16}$$

with $C_0 = \lim_{N\to\infty} \frac{X_0^N}{N}$. Here $W_k(t)$, $k = 1, \ldots, K$, denote independent Wiener processes [93]. We refer to (2.16) as *SDE limit model*, which is also known as *chemical Langevin equation* in the context of chemical reaction kinetics [63]. The pathwise approximation error is given by

$$\sup_{t\in[0,T]} \|X_t^N/N - C_t\| \leq \zeta \frac{\log(N)}{N} \tag{2.17}$$

for finite T and a.s. finite constant ζ, both independent of N [93, 47]. For the first and second order moments it holds that

$$\begin{aligned} \mathbb{E}\left[X_t^N/N\right] &= C_t + \mathcal{O}(N^{-2}), \\ \mathrm{Var}\left(X_t^N/N\right) &= C_t + \mathcal{O}(N^{-2}) \end{aligned}$$

for all $t \geq 0$ [64]. Note again that the ODE and SDE limit models are approximations which only hold for finite time intervals $[0, T]$ that are independent of N. In many cases the SDE (2.16) is not discussed directly but in terms of a mean-field approximation. The corresponding Fokker–Planck equation is given by

$$\frac{\partial}{\partial t}\rho(c,t) = -\sum_{i=1}^{d} \frac{\partial}{\partial c_i}\left(b_i(c)\,\rho(c,t)\right) + \frac{1}{2N}\sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial^2}{\partial c_i \partial c_j}\left(\Sigma_{ij}(c)\,\rho(c,t)\right) \tag{2.18}$$

with

$$b(c) \coloneqq \sum_{k=1}^{K} \tilde{\alpha}_k(c)\,\nu_k \quad \text{and} \quad \Sigma(c) \coloneqq \sum_{k=1}^{K} \tilde{\alpha}_k(c)\,\nu_k\,\nu_k^\top$$

see, e.g., [74, 111]. The Perron–Frobenius operator $\mathcal{T}_N^t$ associated with the SDE limit model is given by $\mathcal{T}_N^t = \exp(t\mathcal{G}_N)$, where $\mathcal{G}_N$ is the operator on the right-hand side

of (2.18). Again, we emphasize its dependence on N. It acts on functions $f \in L^2_\pi([0,1]^d)$. Like the master equation (2.14) of the Markov jump process before, also the Fokker–Planck equation (2.18) cannot be solved analytically in general. Realizations of (2.16) can be generated using an Euler–Maruyama scheme, where the step size of the iterations are asymptotically independent of the population size N. Thus, for large N the SDE limit model (2.16) becomes more efficient than the ABM process (2.12).

Remark 2.2.1. *Linear noise approximation*, which is given by the leading-order term of the corresponding system size expansion, is another commonly used approach to approximate the distribution of the ABM process defined by the master equation (2.13) and to obtain insights into power spectra to detect oscillations as given, e.g., in predator-prey systems [112, 161]. However, estimates of first- and second-order moments by the linear noise approximation are less accurate than for the SDE limit model [64].

2.2.3 Extended Voter Model

The *(extended) voter model* is well-known not only in the field of opinion dynamics as noisy multi-state voter model but also for the description of foraging ant colonies or chemical systems, see, e.g., [69, 13, 128]. This model is characterized by its simplicity yet potential for complexity as ODE and SDE limit models for large numbers of agents can be derived analytically. The model as presented here follows the methodology in [126, 127].

Consider N agents, d opinions and two transitions and assume that agents are nodes in an interaction network, which can be of any shape at this point. To simplify notation, we introduce the use of tuples (i,j) instead of generic indices k. Given two agents with different opinions $S_i \neq S_j$, the first transition is given by *imitation* or *adaption*. It is a second-order transition of the form

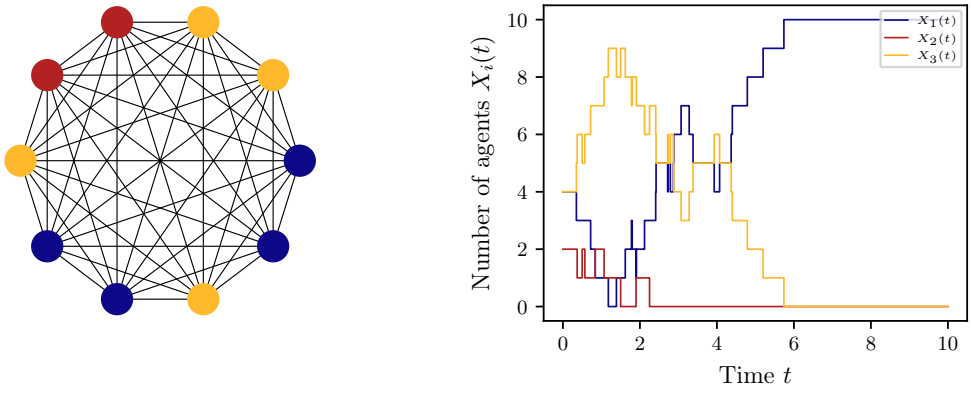
$$R_{ij}\colon \quad S_i + S_j \mapsto 2\,S_j,$$

meaning that an agents with opinion S_i adopts the opinion S_j of another agent. We interpret this, for instance, as persuasion of one agent by the other or adoption of a new technology. This is the "classical" idea of the original voter model as defined in [29, 71]. The second transition is given by *exploration* or *mutation*, i.e., a first-order transition of the form

$$R'_{ij}\colon \quad S_i \mapsto S_j,$$

where agents change their opinion independently of all other agents, i.e., it is a randomly occurring change of state. Although the transitions R_{ij} and R'_{ij} are differently motivated, their net change vectors are identical and given by $\nu_{ij} = e_j - e_i$, where $e_i \in \mathbb{N}_0^d$ denotes the unit vector with the ith entry equal to one and all others zero. Now, assuming a complete interaction network and positive transition rate constants $\gamma_{ij}, \gamma'_{ij} > 0$, the propensity functions for R_{ij} and R'_{ij} are given by

$$\alpha_{ij}(x) = \frac{\gamma_{ij}}{N} x_i\, x_j \quad \text{and} \quad \alpha'_{ij}(x) = \gamma'_{ij}\, x_i.$$



(a) **(b)**

Figure 2.1: (a) Interaction network with $N = 10$ agents and $d = 3$ types at time $t = 0$. (b) Realization of the ABM process. Reprinted from [126].

The scaling with N is justified by the fact that the probability for two specific agents to interact with each other is proportional to $1/N$.

Example 2.2.2. Assume that we have a complete interaction network with $N = 10$ agents, $d = 3$ opinions and that the rate constants are given by

$$\begin{aligned} \gamma_{12} = \gamma_{23} = \gamma_{31} &= 2, \\ \gamma_{32} = \gamma_{21} = \gamma_{13} &= 1, \\ \gamma'_{ij} &= 0.01 \end{aligned}$$

for $(i, j) = \{(1, 2), (2, 3), (3, 1), (3, 2), (2, 1), (1, 3)\}$. Figure 2.1 (a) shows the interaction network at time $t = 0$ with agents' opinions represented by blue, red, and yellow vertices. Figure 2.1 (b) shows a realization of the ABM process obtained by Gillespie's algorithm. △

Remark 2.2.3. The framework introduced in Section 2.2.1 is versatile and can also be used to model, for instance, infection kinetics defined by SIR (susceptible, infected, recovered) models. Setting notation to $S := S_1$ (susceptible), $I := S_2$ (infected) and $R := S_3$ (recovered) the transition rules for the SIR model are given by

$$\begin{aligned} S + I &\mapsto 2I, \quad \text{(infection)} \\ I &\mapsto R, \quad \text{(recovery)}. \end{aligned}$$

Various extension, e.g., with birth and death from any compartment or additional compartments as quarantined agents, are possible.

2.2.4 Predator-Prey Model

The second agent-based system is inspired by nature describing the interaction of predators and prey; thus *predator-prey model*. Our model is an agent-based version of the famous Lotka–Volterra equations, which model the dynamical behavior of two interacting species as differential equations. The model as presented here is taken from [126].

The predator-prey model consists of two different types of agents: predators and prey. Assume that the domain, in which the agents move freely, is continuous with periodic boundary conditions. Gaussian random walks with normally distributed step size determines the movement of the agents. That is, given its current position $z_i(k_0)$, the agent is located after k time steps at position

$$z_i(k_0 + k) = z_i(k_0) + \sum_{i=1}^{k} \zeta_i \quad \text{for} \quad \zeta_i \sim \sqrt{h}\,\mathcal{N}(0,1). \tag{2.19}$$

Unlike the voter model, the predator-prey model is formulated in discrete time. At each time step and according their type, each agent executes the following steps:

> **Prey.** Move and reproduce with probability p_{rep}. Place offspring randomly in the space.
>
> **Predator.** Move and search for prey within a radius of vision v. If prey is available in vision, choose one randomly, kill it and reproduce with probability p'_{rep}. Otherwise die with probability p_{death}.

Figure 2.2 describes the predator-prey model as flow chart in more detail. Note that there is neither interaction nor competition between the prey. For simplicity we assumed independence of resources for the predator-prey model, which results in an unlimited growth of the prey in the absence of predators. Note that the total population size is not constant but kept under control by the existence of predators.

Remark 2.2.4. The predator-prey model cannot be formulated directly using the formalism summarized Section 2.2.1 as a consequence of the spatial component. However, if we assume a *well-mixed* system and denote prey by S_1 and predators by S_2, the rules for the predator-prey model translate to

$$\begin{aligned} S_1 &\mapsto 2\,S_1, && \text{(reproduction of prey)} \\ S_1 + S_2 &\mapsto 2\,S_2, && \text{(reproduction of predators)} \\ S_2 &\mapsto \emptyset, && \text{(death of predators)} \end{aligned}$$

for some rate constants $\gamma_i > 0$, $i = 1, \ldots, 3$ [126].

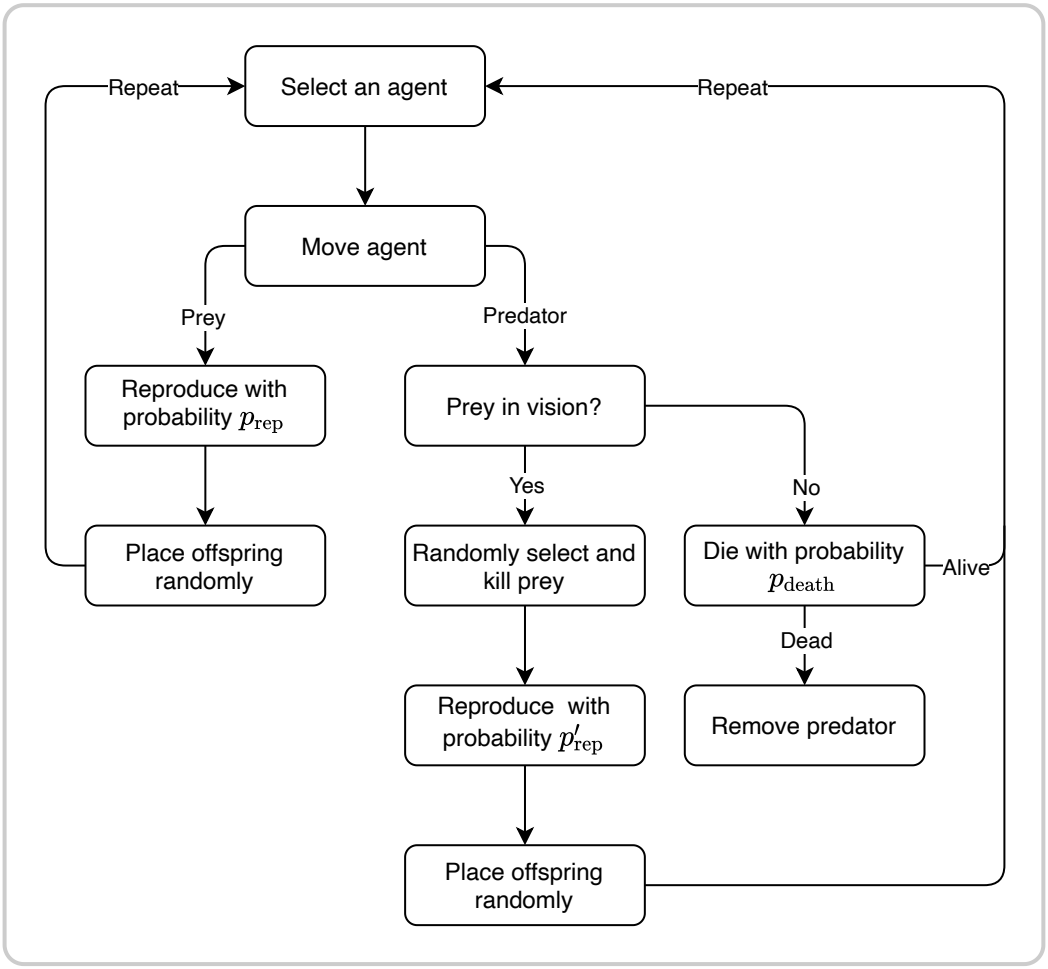
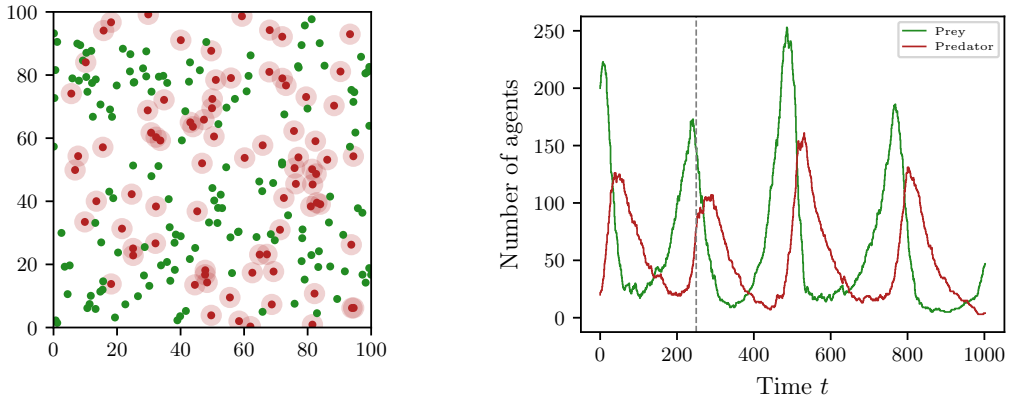


Figure 2.2: Flow chart of the predator-prey model. Reprinted from [126].

Table 2.1: Parameters used for realizations of the predator-prey model.

Parameter	Value
Space height × width	100 × 100
Step size variance h	1
Reproduction probability prey p_{rep}	0.03
Reproduction probability predator p'_{rep}	0.5
Probability of death p_{death}	0.02
Radius of vision v	3



(a) **(b)**

Figure 2.3: (a) Snapshot of the full state of the predator-prey model at time $t = 250$. (b) Realization of the population state for the parameters given in Table 2.1. The vertical gray dashed line marks the time at which the snapshot in (a) is taken. Reprinted from [126].

Example 2.2.5. Figure 2.3 (a) shows a snapshot of the full ABM state at time $t = 250$ of the predator-prey model for a realization using the parameters listed in Table 2.1. Green and red dots represent prey and predators, respectively. The search radius of the predators is indicated by the light-red area around the red dots. The population state is depicted in Figure 2.3 (b) and shows the respective numbers of prey and predators. The time at which the snapshot of the state in Figure 2.3 (a) was taken is marked by a gray dashed line. △

2.2.5 Civil Violence Model

The third agent-based systems models social insurgencies between two kinds of interacting agents in space. In its original version the *civil violence model* [45] is formulated in discrete-time and discrete-space, where agents move through random walks on a grid, temporarily occupying these positions for other agents. In this work, we present a slightly modified version, which allows a punctuated equilibrium and is based on the work in [51].

As for the predator-prey model we assume that the space is continuous with periodic boundary conditions and that agents perform Gaussian random walks of the form (2.19). The civil violence model is formulated in discrete time and consists of two different types of agents: citizens and officers. They are defined as follows:

> **Citizen.** Citizens are heterogeneously characterized by the two inherent parameters hardship H and risk aversion R, which are uniformly distributed in the interval $[0, 1]$. In each step citizens decide deterministically whether they become

rebellious or quiet, i.e., whether their state is active or inactive. The decision rule is given by:

$$\text{change state to} \quad \begin{cases} \text{active}, & \text{if } \; H(1-L) - RP_{\text{arr}} \geq T, \\ \text{inactive}, & \text{else.} \end{cases}$$

The parameters L and T are identically for all agents representing legitimacy of the authority and the activity threshold, respectively. The value P_{arr} represents a spatio-temporally varying arrest probability depending on the individual neighborhood of each agent. It is defined as sigmoid-shaped function $P_{\text{arr}} \colon \mathbb{N}_0^2 \to [0,1]$ given by

$$P_{\text{arr}}(A_v, C_v) \coloneqq 1 - \exp\left(-\kappa \frac{C_v}{A_v}\right) \sum_{i=0}^{15} \frac{\left(\kappa \frac{C_v}{A_v}\right)^i}{i!}, \tag{2.20}$$

where C_v and A_v denote the numbers of officers and active citizens within a neighborhood of radius v and κ a constant. For details on the choice of P_{arr} see [51].

Officer. Officers patrol (defined by (2.19)) and arrest the nearest active citizen in their radius of vision v'. They do not participate in insurgencies. To move quicker during an outburst of violence officers jump to the positions of active citizens. Arrested citizens are sent to jail for J iterations uniformly drawn over an integer distribution $\mathcal{U}(0, J_{\max})$ and released on a random position if they completed their prison sentence.

All parameters are fixed for the entire simulation. Figure 2.4 shows the civil violence model as flow chart in more detail. Note that differently from other implementations here a citizen first decides based on its current environment whether to become active or not before moving to a new position. This can be interpreted as an active decision to participate in a demonstration or revolution, such as the violent G20 demonstrations in Hamburg, Germany in July 2017. It is shown in [51] that the civil violence model with arrest probability function (2.20) allows to model repetitive outburst of violence, whereas this behavior cannot be observed in the original version. For further studies on the dynamical behavior of the civil violence model we refer to [45, 51, 177, 32, 100, 52, 70].

Example 2.2.6. Figure 2.5 (a) shows a snapshot of the full ABM at time $t = 119$ of the civil violence model for a typical realization using the parameters listed in Table 2.2. Inactive and active citizens are represented by green and red dots, respectively. Officers are indicated in blue. The population state is depicted in Figure 2.5 (b) in terms of the number of actives, jailed and inactive citizens. △

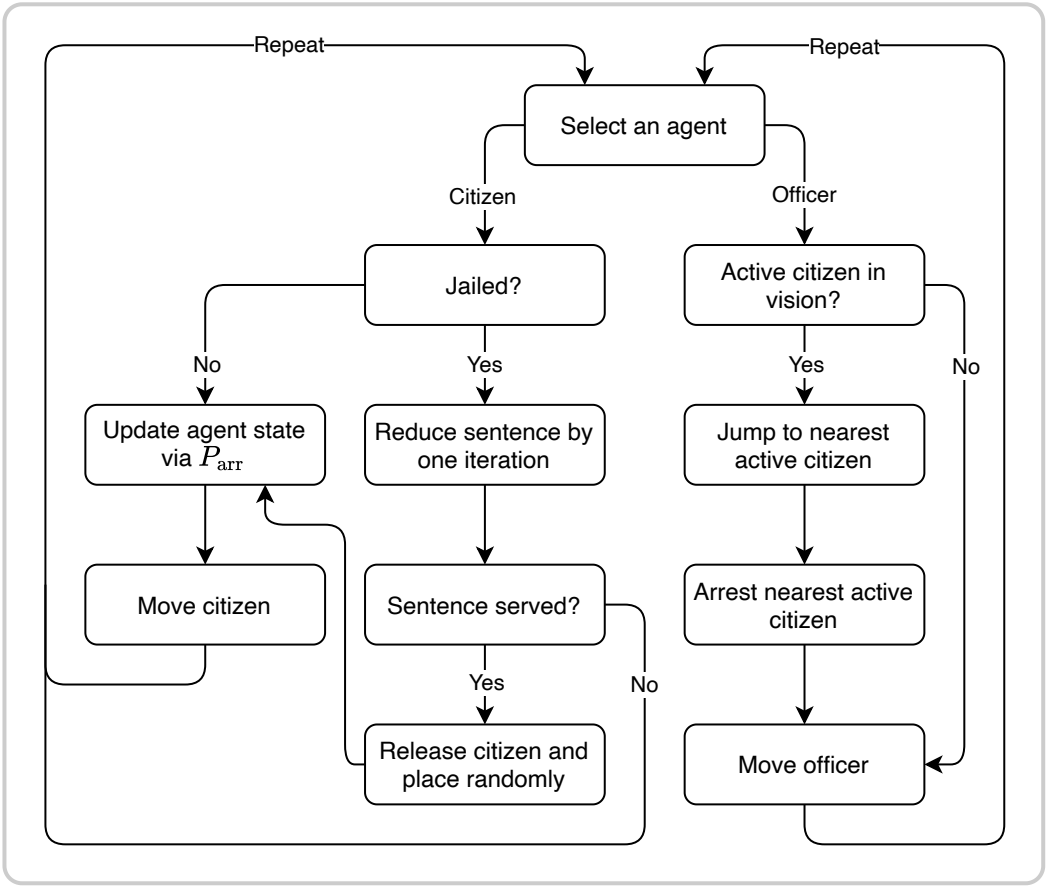
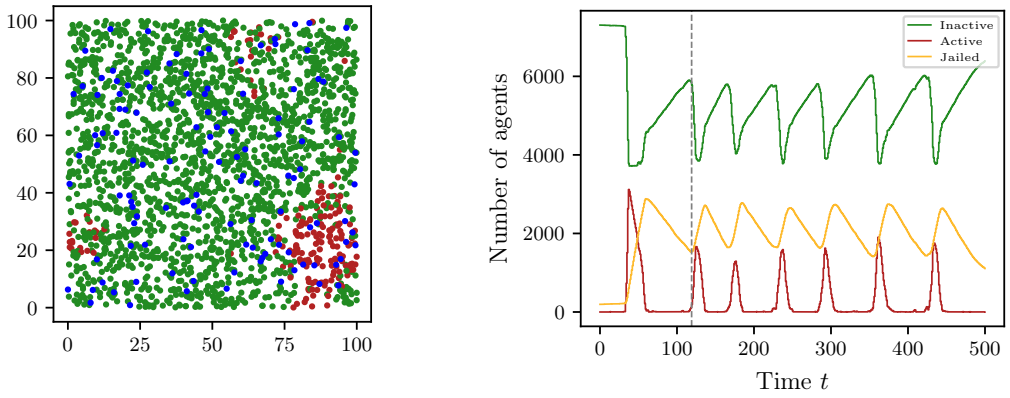


Figure 2.4: Flow chart of the civil violence model.

Table 2.2: Parameters used for realizations of the civil violence model.

Parameter	Value
Space height × width	100 × 100
Number of agents N	7500
Number of police officers	125
Citizens' step size variance h	1
Officers' step size variance h'	1
Citizens' radius of vision of v	10
Officers' radius of vision of v'	10
Activity threshold T	0.1
Legitimacy L	0.8
Maximum jail sentence J_{max}	120
Constant κ	62.6716



(a) **(b)**

Figure 2.5: (a) Snapshot of the full state of the civil violence model at time $t = 119$. (b) Realization of the population state for the parameters given in Table 2.2. The vertical gray dashed line marks the time at which the snapshot in (a) is taken.

3 Data-Driven Approximation of the Koopman Generator

This chapter concentrates on the data-driven approximation of the Koopman generator for both deterministic and stochastic dynamical systems. We present a general framework based on standard EDMD (cf. Section 2.1.4) to compute a matrix representation of the infinitesimal generator of the Koopman operator from data without requiring trajectory integration. Additionally, it can be used to approximate eigenvalues, eigenfunctions and modes of the generator. Due to duality, the framework extends to approximate the generator of the Perron–Frobenius operator. We show that the generator approximation can be used to identify the governing equations of deterministic and stochastic dynamical systems. We also show relationships to other methods for system identification such as the Koopman lifting technique [109], SINDy [20] and KRONIC [76]. The latter, however, focus mainly on ODEs. Lastly, we demonstrate the numerical efficacy and advantage of the resulting method with the help of suitable examples and benchmark problems. Looking ahead to Chapter 5, we will explore another powerful application of the Koopman generator approximation. We will show that the framework can be used to identify reduced models directly from high-dimensional (noisy) data provided by agent-based systems. This significantly reduces the computational effort and facilitates the analytical investigation.

The results presented in this chapter appeared in our publication [85], to which the author has made significant contributions.

3.1 Infinitesimal Generator EDMD

As the Koopman operator, so its infinitesimal generator can be used to extract important properties of dynamical systems. We now reformulate standard EDMD introduced in Section 2.1.4 to compute approximations of the infinitesimal generators of Koopman or Perron–Frobenius operator as well as the corresponding eigenvalues, eigenfunctions and modes. To emphasize the connection to standard EDMD, the proposed technique is called *generator EDMD* or shortly *gEDMD*.

3.1.1 Deterministic Dynamical Systems

Consider an ODE of the form (2.1) and assume that we have m measurements of the system's states $\{x_l\}_{l=1}^m$, and their corresponding pointwise time derivatives, given by $\{\dot{x}_l\}_{l=1}^m$, which can also be estimated from data, cf. [21]. The deterministic case has

also been derived from another perspective with different motivations in [76, 75]. We discuss their connection in detail in Section 3.2.

Generator Approximation

Just like EDMD, also generator EDMD requires a set of basis functions. Let $\{\psi_i\}_{i=1}^n$ denote the set of basis functions. Writing it in vector form as $\psi(x) = [\psi_1(x), \dots, \psi_n(x)]^\top$, we are able to define

$$\dot{\psi}_k(x) = (\mathcal{L}\psi_k)(x) = \sum_{i=1}^{d} b_i(x) \frac{\partial \psi_k}{\partial x_i}(x).$$

The partial derivatives of the basis functions required for $\dot{\psi}_k(x_l)$ do not impose a problem as they can be precomputed analytically or, alternatively, obtained via automatic differentiation or symbolic computing toolboxes. Further note that $b(x_l)$ is simply given by $\dot{x}_l$, which can be approximated using, e.g., finite differences if a direct measurement is not possible. We can set now up the matrices $\Psi_X, \dot{\Psi}_X \in \mathbb{R}^{n\times m}$ for all data points and basis functions, which are given by

$$\Psi_X = \begin{bmatrix} \psi_1(x_1) & \dots & \psi_1(x_m) \\ \vdots & \ddots & \vdots \\ \psi_n(x_1) & \dots & \psi_n(x_m) \end{bmatrix} \quad \text{and} \quad \dot{\Psi}_X = \begin{bmatrix} \dot{\psi}_1(x_1) & \dots & \dot{\psi}_1(x_m) \\ \vdots & \ddots & \vdots \\ \dot{\psi}_n(x_1) & \dots & \dot{\psi}_n(x_m) \end{bmatrix}. \tag{3.1}$$

Analogously to EDMD, we assume that there exists a matrix M such that $\dot{\Psi}_X = M\Psi_X$, which we solve in least-square sense by minimizing $\|\dot{\Psi}_X - M\Psi_X\|_F$ since in general this problem cannot be solved exactly. Then, the solution is given by

$$M = \dot{\Psi}_X \Psi_X^+ = \left(\dot{\Psi}_X \Psi_X^\top\right)\left(\Psi_X \Psi_X^\top\right)^+ = \widehat{A}\widehat{G}^+$$

with

$$\widehat{A} = \frac{1}{m}\sum_{l=1}^{m} \dot{\psi}(x_l)\,\psi(x_l)^\top \quad \text{and} \quad \widehat{G} = \frac{1}{m}\sum_{l=1}^{m} \psi(x_l)\,\psi(x_l)^\top.$$

Since the Koopman generator might be sparse even when the operator for the time-t map is not, its approximation can be advantageous.

As the deterministic case represents a special case of non-deterministic systems for $\sigma = 0$, we will show the proof of convergence to a Galerkin approximation in the infinite data limit, i.e., $m \to \infty$, in Section 3.1.2. The matrix $L = M^\top$ is an empirical estimate of the matrix representation of the infinitesimal generator $\mathcal{L}$. By duality, the matrix representation of the adjoint operator $\mathcal{L}^*$, i.e., the generator of the Perron–Frobenius operator, is given by $\widehat{L}^* = (M^*)^\top = (\widehat{A}^\top \widehat{G}^+)^\top$. If the gEDMD approximation converges for $n \to \infty$ to the Koopman generator as the standard EDMD approximation does for the Koopman operator, see [88] for details, is an open question and will be studied in future work.

Remark 3.1.1. In the case that the measurement data x_l are, e.g., corrupted by noise, or if the time derivatives $\dot{x}_l$ need to be approximated numerically, spurious nonzero entries might exists in the matrix M. In order to reduce these entries the same sparsification approach as proposed for SINDy, see [20], can be applied to gEDMD.

Example 3.1.2. Consider again the dynamical system defined in Example 2.1.3 for $\gamma = -0.8$ and $\delta = -0.7$ and use a set of basis functions comprising of monomials up to degree 8. We generated 1000 uniformly distributed training points in $\mathbb{X} = [-2, 2] \times [-2, 2]$ and applied gEDMD. We obtain the eigenvalues and (rescaled) eigenfunctions

$$\begin{aligned} \lambda_1 &\approx 0, & \varphi_1(x) &\approx 1, \\ \lambda_2 &\approx -0.7 = \delta, & \varphi_2(x) &= 1.286\, x_2 + 1.000\, x_1^2 \approx \tfrac{2\gamma-\delta}{\delta} x_2 + x_1^2, \\ \lambda_3 &\approx -0.8 = \gamma, & \varphi_3(x) &\approx x_1, \end{aligned}$$

which agree with the analytically computed ones in Example 2.1.3. Again, we obtain further eigenfunctions as products of the previous eigenfunctions, e.g., $\lambda_6 \approx -1.6 = 2\gamma$ with $\varphi_6(x) = 1.000\, x_1^2 \approx \varphi_3(x)^2$. The ordering of the eigenfunctions and eigenvalues depends on the values of γ and δ. The eigenvalues are typically sorted by decreasing values. △

System Identification

With gEDMD it is possible to reconstruct the governing equations of the underlying dynamical system using the full-state observable $g(x) = x$. Assume that the state space $\mathbb{X}$ is bounded so that the full-state observable is (component-wise) contained in $L^\infty(\mathbb{X})$ and that it can be represented by the basis functions ψ_i, which can trivially be assured by adding the observables $\{x_i\}_{i=1}^d$ to the set of basis function. Further, let ξ_ℓ denote the ℓth eigenvector of $\widehat{L}$ and $\Xi = [\xi_1, \dots, \xi_n]$ and assume that $B \in \mathbb{R}^{n \times d}$ is the matrix such that $g(x) = B^\top \psi(x)$. Define $\varphi(x) = [\varphi_1(x), \dots, \varphi_n(x)]^\top = \Xi^\top \psi(x)$. Then

$$g(x) = B^\top \psi(x) = B^\top \Xi^{-\top} \varphi(x),$$

where the column vectors of the matrix $V = B^\top \Xi^{-\top}$ are the Koopman modes v_ℓ for the full-state observable. The derivation of the modes works analogously to the standard EDMD case, see Section 2.1.4 or [82, 171] for further details. Applying the generator component-wise, we obtain a representation of the system in terms of eigenvalues, eigenfunctions, and modes of the generator, i.e.,

$$(\mathcal{L}g)(x) = b(x) \approx \sum_{\ell=1}^{n} \lambda_\ell \varphi_\ell(x) v_\ell,$$

which allows to decompose the system into different frequencies. Alternatively, the system can directly be represented in terms of the basis functions, that is

$$(\mathcal{L}g)(x) = b(x) \approx (\widehat{L}B)^\top \psi(x),$$

which is then equivalent to SINDy. We discuss it in more detail in Section 3.2. The following example demonstrates the procedure.

Example 3.1.3. In Example 3.1.2 we compute the eigenvalues λ_ℓ and corresponding eigenfunctions $\varphi_\ell(x)$. To reconstruct the dynamical system from Example 2.1.3 we still need to compute the Koopman modes. These are given by

$$\begin{aligned} v_2 &= [0,\, 0.778]^\top \approx [0,\, \frac{\delta}{2\gamma - \delta}]^\top, \\ v_3 &= [1,\, 0]^\top, \\ v_6 &= [0,\, -0.778]^\top \approx [0,\, -\frac{\delta}{2\gamma - \delta}]^\top. \end{aligned}$$

All other modes are numerically zero. Then, we obtain

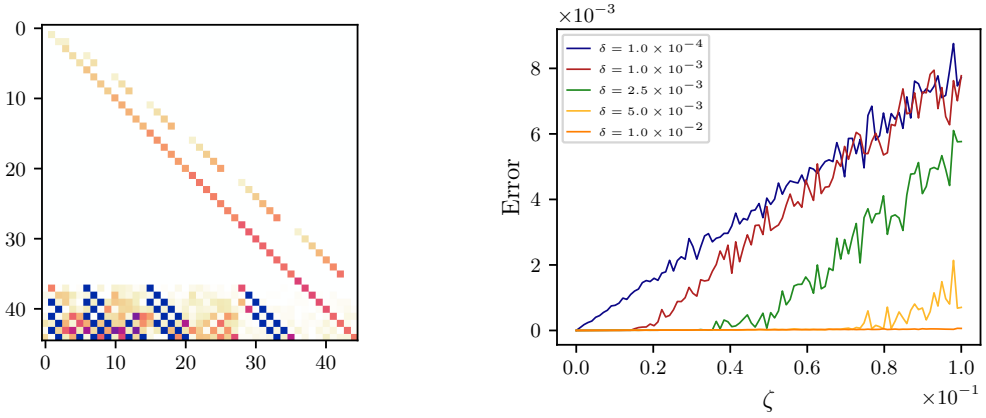
$$b(x) \approx \lambda_2 \varphi_2(x) v_2 + \lambda_3 \varphi_3(x) v_3 + \lambda_6 \varphi_6(x) v_6 \approx \begin{bmatrix} \gamma x_1 \\ \delta\,(x_2 - x_1^2) \end{bmatrix}.$$

Expressing it directly using the basis functions, this results in

$$b(x) \approx (\widehat{L}\,B)^\top \psi(x) = \begin{bmatrix} 0 & -0.8 & 0 & 0 & 0 & \dots \\ 0 & 0 & -0.7 & 0.7 & 0 & \dots \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \gamma x_1 \\ \delta\,(x_2 - x_1^2) \end{bmatrix}.$$

Hence, the governing equations are identified correctly in both cases. Furthermore, the matrix $\widehat{L}$ is sparse as shown in Figure 3.1 (a). Note that the nonzero entries in the last rows of the matrix are due to the fact that $\mathcal{L}\psi_k$ cannot be represented accurately with the chosen basis. For instance, for $\psi_{45}(x) = x_2^8$, we obtain $(\mathcal{L}\psi_{45})(x) = 8\delta x_2^7 (x_2 - x_1^2)$. However, the term $x_1^2 x_2^7$ is not contained in the set of basis functions since its degree is greater than 8. △

Remark 3.1.4. We assumed in the previous two examples that the derivatives for the training data are known or can be computed with sufficient accuracy. However, for inaccurate estimates or noise-corrupted data the resulting matrix representations of the operators often become nonsparse. To eliminate spurious nonzero entries we can apply further techniques like denoising, total-variation regularization, or iterative hard thresholding, see also [21] and references therein. Assume that the derivatives $b(x_l)$ are corrupted by noise, i.e., $b(x_l) + \eta$, where η is sampled from a Gaussian distribution with standard deviation ς. The iterative hard thresholding procedure proposed in [21] progressively eliminates all entries smaller than a given threshold δ and then recomputes the coefficients. In this way, applying hard thresholding to gEDMD we can remove undesired entries from the matrix representation $\widehat{L}$. However, attention is required as



(a) **(b)**

Figure 3.1: (a) Sparsity pattern of the Koopman generator approximation $\widehat{L}$ computed in Example 3.1.3. Darker colors represent entries with larger magnitude. (b) Recovery error depending on the standard deviation ς for different thresholds δ. The results reveal that in the case of inaccurate estimates of the derivatives further techniques are necessary to obtain suitable representations of the system. For a well-chosen cut-off value, iterative hard thresholding allows to recover the correct dynamics in the presence of noise. Without thresholding the results agree with the $\delta = 1 \times 10^{-4}$ case. Figure (b) reproduced from [85].

the sparsification results depend strongly on the chosen threshold as Figure 3.1 (b) suggests. We define the recovery error as the average difference between the true and the estimated coefficients after 10 iterations of the hard thresholding procedure. On the one hand, for a smaller signal-to-noise ratio, a larger threshold is needed to eliminate spurious nonzero entries. On the other hand, if the threshold is chosen too large, it also removes actual coefficients, which then deteriorates the approximation accuracy.

Conservation Laws

For a given dynamical system, a function $E\colon \mathbb{R}^d \to \mathbb{R}$ is said to be a *conserved quantity* if it remains constant for all t and all initial values, i.e., $\frac{\mathrm{d}}{\mathrm{d}t}E = \nabla E \cdot b = 0$. This immediately implies that E is an eigenfunction of the Koopman generator associated with eigenvalue $\lambda = 0$. Analogously, invariant densities are given by eigenfunctions of the Perron–Frobenius generator corresponding to eigenvalue $\lambda = 0$. Note that conserved quantities are not unique, i.e., any linear combination is a conserved quantity as well. Further, not all systems have conserved quantities, for instance, mechanical systems with damping. As early as 1931 Koopman considered conserved quantities in his original paper [87]. Conservation laws and conserved quantities play an important role in physics and engineering. However, their discovery is in principle difficult so that their numerical

approximation is helpful. With the aid of gEDMD it is possible to identify non-trivial eigenfunctions associated $\lambda = 0$, that is, conserved quantities, from data. Similarly, in [76, 75] the connection between conservation laws and Koopman eigenfunctions is used to find conserved quantities from data. The relationship of both methods will be discussed in detail in Section 3.2.

Example 3.1.5. Let us compute the conserved quantities of the following two systems with the help of gEDMD. Assume that the derivatives $\{\dot{x}_l\}_{l=1}^m$ are known. For both systems we generated 1000 uniformly distributed training points in $\mathbb{X} = [0,1] \times [0,1]$.

(i) Let us consider the undamped Duffing oscillator, i.e., for $\alpha, \beta \in \mathbb{R}$ we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\alpha x_1 - \beta x_1^3 \end{bmatrix}.$$

Choosing $\alpha = -1.1$, $\beta = 1.1$ and a dictionary comprising of monomials up to degree 5, the multiplicity of eigenvalue $\lambda = 0$ is two and we obtain a conserved quantity of the form

$$E(x) = -0.5500\,x_1^2 + 0.2750\,x_1^4 + 0.5\,x_2^2 - 0.0148 \approx \frac{\alpha}{2}x_1^2 + \frac{\beta}{4}x_1^4 + \frac{1}{2}x_2^2 + c,$$

where $c \in \mathbb{R}$ is an arbitrary constant.

(ii) Consider the mathematical pendulum, i.e., for $\alpha \in \mathbb{R}$ we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\alpha \sin(x_1) \end{bmatrix}.$$

For this example we choose a dictionary that consists of monomials and trigonometric functions. Again, for $\alpha = 1$ the multiplicity of eigenvalue $\lambda = 0$ is two and we obtain a conserved quantity of the form

$$E(x) = 0.5\,x_2^2 - 1.000\,\cos(x_1) - 1.0712 \approx 0.5\,x_2^2 - \alpha\,\cos(x_1) + c,$$

where $c \in \mathbb{R}$ is an arbitrary constant. △

3.1.2 Non-deterministic Dynamical Systems

Let us now generalize gEDMD to SDEs of the form (2.2). Assume that we have a set of m measurements of the system state $\{x_l\}_{l=1}^m$, the drift $\{b(x_l)\}_{l=1}^m$ as well as the diffusion $\{\sigma(x_l)\}_{l=1}^m$. Drift and diffusion are assumed to be known or that they can be estimated pointwise. The latter will be the case in Chapter 5, where gEDMD is applied to obtain coarse-grained representations of agent-based systems. Again, let $\{\psi_i\}_{i=1}^n$ denote the set of basis functions.

Generator Approximation

Let

$$\mathrm{d}\psi_k(x) = (\mathcal{L}\psi_k)(x) = \sum_{i=1}^{d} b_i(x)\,\frac{\partial \psi_k}{\partial x_i}(x) + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} a_{ij}(x)\,\frac{\partial^2 \psi_k}{\partial x_i \partial x_j}(x) \tag{3.2}$$

and for all data points and basis functions, we assemble the matrix $\mathrm{d}\Psi_X \in \mathbb{R}^{n\times m}$ with

$$\mathrm{d}\Psi_X = \begin{bmatrix} \mathrm{d}\psi_1(x_1) & \dots & \mathrm{d}\psi_1(x_m) \\ \vdots & \ddots & \vdots \\ \mathrm{d}\psi_n(x_1) & \dots & \mathrm{d}\psi_n(x_m) \end{bmatrix}.$$

Again, the partial derivatives of the basis functions can be precomputed analytically. Differently from the deterministic case, here we also need the second derivatives. As before, we assume that there exists a matrix M such that $\mathrm{d}\Psi_X = M\Psi_X$, which leads to the minimization problem $\|\mathrm{d}\Psi_X - M\Psi_X\|_F$. Solving it in least-square sense results in

$$M = \mathrm{d}\Psi_X \Psi_X^{+} = \big(\mathrm{d}\Psi_X \Psi_X^{\top}\big)\big(\Psi_X \Psi_X^{\top}\big)^{+} = \widehat{A}\widehat{G}^{+}$$

with

$$\widehat{A} = \frac{1}{m}\sum_{l=1}^{m} \mathrm{d}\psi(x_l)\,\psi(x_l)^{\top} \quad \text{and} \quad \widehat{G} = \frac{1}{m}\sum_{l=1}^{m} \psi(x_l)\,\psi(x_l)^{\top}.$$

As in the deterministic case, the solution $M = \widehat{L}^{\top} = \widehat{A}\widehat{G}^{+}$ is an empirical estimate of the matrix representation of the generator $\mathcal{L}$ and $M^* = (\widehat{L}^*)^{\top} = \widehat{A}^{\top}\widehat{G}^{+}$ an estimate of the adjoint operator $\mathcal{L}^*$. Proposition 3.1.6, which appeared along with the corresponding proof in our publication [85], summarizes the convergence results of gEDMD.

Proposition 3.1.6 (Proposition 3.5 in [85])**.** *For an infinitely large set of training data, i.e., $m \to \infty$, gEDMD converges to the Galerkin projection* $\mathbf{L}$ *of the generator $\mathcal{L}$ onto the space spanned by the basis functions $\{\psi_i\}_{i=1}^{n}$.*

Proof. Let $m \to \infty$, we obtain

$$\widehat{A}_{ij} = \frac{1}{m}\sum_{l=1}^{m} \mathrm{d}\psi_i(x_l)\,\psi_j(x_l) \underset{m\to\infty}{\longrightarrow} \int (\mathcal{L}\psi_i)(x)\,\psi_j(x)\,\mathrm{d}\mu(x) = \langle \mathcal{L}\psi_i, \psi_j\rangle_\mu = A_{ij},$$

$$\widehat{G}_{ij} = \frac{1}{m}\sum_{l=1}^{m} \psi_i(x_l)\,\psi_j(x_l) \underset{m\to\infty}{\longrightarrow} \int \psi_i(x)\,\psi_j(x)\,\mathrm{d}\mu(x) = \langle \psi_i, \psi_j\rangle_\mu = G_{ij},$$

where $x_l \sim \mu$. This means that the matrices $\widehat{A}$ and $\widehat{G}$ are empirical estimates of the matrices A and G, respectively. □

For the deterministic case, i.e., $\sigma = 0$, replace $\mathrm{d}\psi_i(x_l)$ by $\dot{\psi}_i(x_l)$. The proof is analogously to standard EDMD.

Remark 3.1.7. As before in the deterministic case we assumed that the drift and diffusion coefficients of the SDE (2.2) are known. However, both can be estimated pointwise from data, e.g., by finite differences via the Kramers–Moyal formulae. We will discuss this case in detail in Chapter 5. If a single ergodic simulation is available, the definition of $\mathrm{d}\psi_k$ in (3.2) can alternatively be replaced by

$$\mathrm{d}\psi_k(x_l) = \frac{1}{t}(x_{l+1} - x_l) \cdot \nabla\psi_k(x_l) + \frac{1}{2t}\left[(x_{l+1} - x_l)(x_{l+1} - x_l)^\top\right] : \nabla^2\psi_k(x_l).$$

Remark 3.1.8. If the stochastic dynamical system (2.2) is reversible with respect to the measure μ, it is sufficient to use only the first-order derivatives of the basis functions as the Galerkin matrix A in (2.7) can be expressed as

$$A_{ij} = \langle \mathcal{L}\psi_i, \psi_j \rangle_\mu = -\frac{1}{2}\int \nabla\psi_i \sigma\sigma^\top \nabla\psi_j^\top \,\mathrm{d}\mu.$$

The drift coefficient enters implicitly via the invariant measure μ, see [175] for details. The empirical estimate $\widehat{A}$ for A is defined as

$$\widehat{A} = -\frac{1}{2m}\sum_{l=1}^{m} \mathrm{d}\psi(x_l)\,\mathrm{d}\psi(x_l)^\top,$$

where $\mathrm{d}\psi(x_l) = \nabla\Psi(x_l)\sigma(x_l)$. Here, $\nabla\Psi \in \mathbb{R}^{n\times d}$ denotes the gradient matrix, where each row corresponds to the gradient of a basis function.

Example 3.1.9. Let us compute the eigenfunctions of the Koopman and the Perron–Frobenius generator for three examples. Assume that $\{b(x_l)\}_{l=1}^m$ and $\{\sigma(x_l)\}_{l=1}^m$ are known and not estimated from data.

(i) Consider again the Ornstein–Uhlenbeck process defined in Example 2.1.5 and set $\alpha = 1$ and $\beta = 4$. We choose monomials up to order 10 and generated 100 uniformly distributed training points in $\mathbb{X} = [-2, 2]$. The gEDMD result is shown in Figure 3.2 (a). The numerically computed Koopman generator eigenfunctions are practically indistinguishable from the analytical solutions. For this small amount of training data the same computation using standard EDMD usually gives a less accurate approximation of the dominant eigenfunctions. As already mentioned in Remark 2.1.10, the choice of basis functions is a crucial step for the approximation accuracy. This can be observed in Figure 3.2 (b). Using the same set of basis functions as for the Koopman generator, the dominant eigenfunctions of the Perron–Frobenius generator cannot be approximated that well. If we select Gaussian functions instead of monomials, we can significantly improve the results as it is shown in Figure 3.2 (c). Here, we choose 30 Gaussian functions with bandwidth $\sigma = 0.1$. Unfortunately, the most suitable set of basis functions is in general unknown in advance. Let us consider now the sparsity patters of the generator approximations by gEDMD and EDMD corresponding to the Ornstein–Uhlenbeck. The ideal resulting model is parsimonious since sparse representations minimize model complexity while simultaneously enabling accurate predictions without overfitting. The

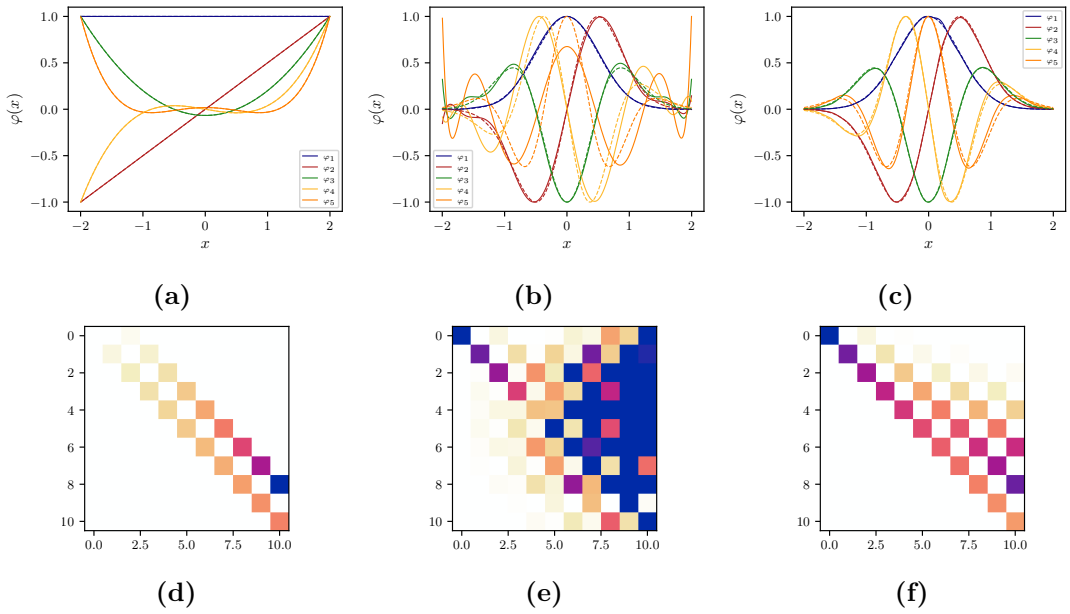


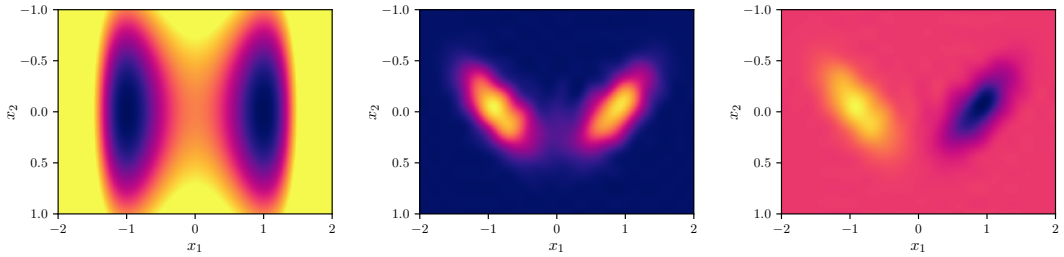
Figure 3.2: Eigenfunctions of (a) the Koopman generator and (b) the Perron–Frobenius generator associated with the Ornstein–Uhlenbeck process computed via gEDMD and monomials up to degree 10 compared to the analytically computed eigenfunctions (dashed). (c) Eigenfunctions of the Perron–Frobenius generator but with 30 Gaussian functions as dictionary. (d-f) Sparsity patterns of the generator approximation $\widehat{L}$ computed with gEDMD, operator approximation $\widehat{K}^\tau$ computed with EDMD, and operator approximations $\exp(\tau\widehat{L})$ obtained from $\widehat{L}$ using the same lag time τ as for $\widehat{K}^\tau$. Darker colors represent entries with larger magnitude. Reproduced from [85].

Figures 3.2 (d–f) show the sparsity patters of $\widehat{L}$ for the generator computed via gEDMD and twice $\widehat{K}^\tau$ for the operator, which is once computed directly via EDMD and once via $\exp(\tau\widehat{L})$ using the matrix exponential and the same lag time $\tau = 0.1$ as used for EDMD. The result reveals that gEDMD leads to a sparser representation with less spurious nonzero entries. Comparing Figure 3.2 (d) with the analytically computed generator representation in Example 2.1.8 demonstrates the efficiency of gEDMD.

(ii) We consider now a more complex example given by an SDE of the form (2.2) with state-dependent, non-isotropic diffusion term, i.e.,

$$b(x) = -\nabla V(x) = \begin{bmatrix} 4x_1 - 4x_1^3 \\ -2x_2 \end{bmatrix} \quad \text{and} \quad \sigma(x) = \begin{bmatrix} 0.7 & x_1 \\ 0 & 0.5 \end{bmatrix},$$

where $V(x) = (x_1^2 - 1)^2 + x_2^2$ represents the double-well potential giving raise to metastable behavior in terms of rare transitions between the two wells. We can detect metastable states with the help of the generator eigenvalues. More precisely, the metastable sets can be identified by the eigenfunctions corresponding to the dominant



(a) **(b)** **(c)**

Figure 3.3: (a) Double-well potential. (b) First and (c) second eigenfunction of the Perron–Frobenius generator showing two clearly separated wells, which are tilted due to the non-isotropic noise. In all plots, blue corresponds to small and yellow to large values. Reproduced from [85].

eigenvalues close to the largest eigenvalue $\lambda = 0$. Moreover, the number of metastable sets is equal to the number of dominant eigenvalues close to $\lambda = 0$ (including $\lambda = 0$ and counting multiplicity). Figure 3.3 (a) shows the potential V and (b-c) two dominant eigenfunctions of the Perron–Frobenius generator computed via gEDMD. The two metastable states can be well identified via the first two eigenfunctions of the respective Perron–Frobenius generator. Both sets are well separated by the second eigenfunction as shown in Figure 3.3 (c). The corresponding two dominant eigenvalues and the third are approximately given by

$$\lambda_1 \approx -0.0194, \quad \lambda_2 \approx -0.2042 \quad \text{and} \quad \lambda_3 \approx -2.0230.$$

Note the clear gap between the second and third eigenvalue. Here, we used 30000 training points in $\mathbb{X} = [-2, 2] \times [-1, 1]$ with a basis consisting of 300 Gaussian functions with bandwidth $\sigma = 0.2$. The basis functions are centered at the midpoints of a regular box discretization. △

(iii) Let us change V to a quadruple-well potential, i.e., $V(x) = (x_1^2-1)^2+(x_2^2-1)^2$, and consider an SDE of the form (2.2), again with state-dependent, non-isotropic diffusion term, that is,

$$b(x) = -\nabla V(x) = \begin{bmatrix} 4x_1 - 4x_1^3 \\ 4x_2 - 4x_2^3 \end{bmatrix} \quad \text{and} \quad \sigma(x) = \begin{bmatrix} 1 & x_1 x_2 \\ 0 & 1 \end{bmatrix}.$$

Figure 3.4 (a) shows the potential V and (b-e) the four dominant eigenfunctions of the Perron–Frobenius generator computed via gEDMD using 400 radial functions with bandwidth $\sigma = 0.15$. Here, we generated 400000 training data points in $\mathbb{X} = [-2, 2] \times [-2, 2]$. As before, the Perron–Frobenius eigenfunctions can be used to detect the four metastable states. The corresponding four dominant eigenvalues are approximately given by

$$\lambda_1 \approx -0.1534, \quad \lambda_2 \approx -0.4163, \quad \lambda_3 \approx -0.6532 \quad \text{and} \quad \lambda_4 \approx -0.8984.$$

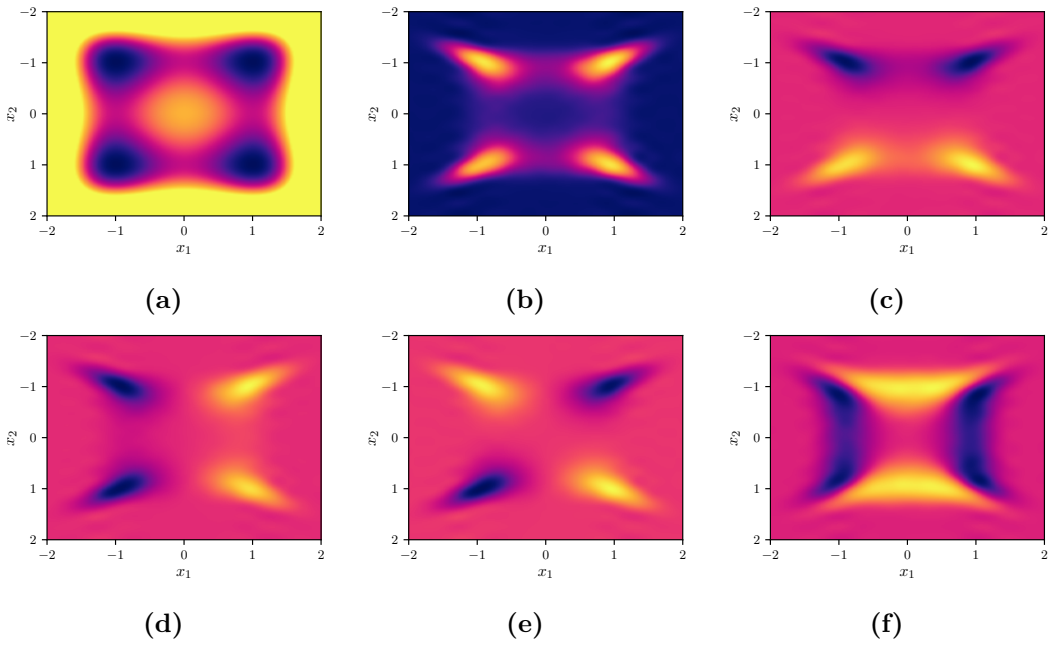


Figure 3.4: (a) Quadruple-well potential. (b-e) First four dominant eigenfunction of the Perron–Frobenius generator. As for the double-well problem, the eigenfunctions clearly separate the wells from each other. Again, the non-isotropic noise causes all wells to be tilted. (f) Non-dominant eigenfunction associated with eigenvalue λ_5 corresponding to a faster process. In all plots, blue corresponds to small and yellow to large values.

The fifths eigenvalue, which is given by $\lambda_5 \approx -4.1267$, shows a clear offset. The corresponding eigenfunction is depicted in Figure 3.4 (f). Note that also the Koopman eigenfunctions encode this information. Figure 3.5 shows the first three nontrivial eigenfunctions of the Koopman generator indicating a possible partition of the state space $\mathbb{X}$. We computed them for Figure 3.5 (a-c) using a dictionary of monomials up to and including order 4 and for Figure 3.5 (d-f) using 400 radial basis functions as for the Perron–Frobenius generator as dictionary. Additionally, we observe that the eigenfunctions in Figure 3.5 (d-f) are coarser than in Figure 3.5 (a-c). This is as expected since we only use smooth basis functions in the first case, which results in smooth representations of the eigenfunctions. △

Remark 3.1.10. Due to the connection between operator and generator, we can also use the eigenfunctions of the operator to detect metastable states. In this case the number of metastable sets is given by the number of dominant eigenvalues close to the maximal eigenvalue $\lambda^t = 1$ (including $\lambda^t = 1$ and counting multiplicity). The dominant eigenfunctions φ_ℓ show different sign combinations when weighted with the invariant measure, i.e., $\varphi_\ell = \frac{\varphi_\ell}{\mu}$, see [153]. Considering their zeros, we can decompose the state

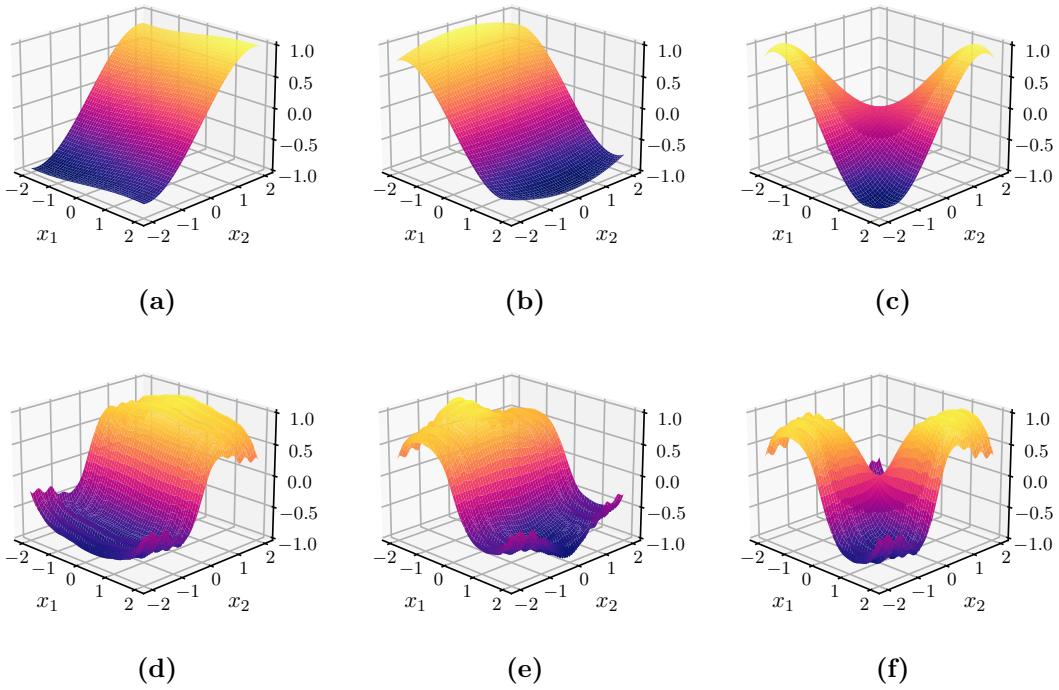


Figure 3.5: First three nontrivial eigenfunctions of the Koopman generator for the quadruple-well problem computed using a dictionary consisting in (a-c) of monomials up to and including order 4 and in (d-f) of 400 radial basis functions with bandwidth $\sigma = 0.15$.

space into metastable sets with associated asymptotic exit rates given approximately by the inverse dominant time scales, see [73].

System Identification

As already in the deterministic case, gEDMD can also be used in the non-deterministic case for system identification. Making the same assumptions on $\mathbb{X}$, we discover the drift term b of the underlying dynamics with the aid of the full-state observable g. To identify the diffusion term, note that for $\psi_k(x) = x_i x_j$, it holds that

$$a_{ij}(x) = (\mathcal{L}\psi_k)(x) - b_i(x)\,x_j - b_j(x)\,x_i, \tag{3.3}$$

where $a = \sigma\sigma^\top$. Note that the drift term b has to be identified in advance. Additionally, we assume that both b_i and b_j as well as $b_i(x)\,x_j$ and $b_j(x)\,x_i$ are representable in the space spanned by the basis functions $\{\psi_i\}_{i=1}^n$. Practically, this means that if, for instance, b contains monomials of degree p, then the dictionary must also include monomials of degree $p+1$. This requirement applies to all types of basis functions.

Remark 3.1.11. Note that using (3.3), we do not identify σ but $a = \sigma\sigma^\top$. If the drift term σ itself is needed, for instance, to generate new trajectories of the identified system, we can obtained it by a Cholesky decomposition of a, see also [175]. Further note that σ is not defined uniquely.

Example 3.1.12. Let us review Example 3.1.9 to illustrate the recovery of the drift and diffusion terms from the generator representation.

(i) For the Ornstein–Uhlenbeck process, we identify

$$b(x) = (\mathcal{L}\psi_2)(x) = -\alpha x \quad \text{and} \quad a(x) = (\mathcal{L}\psi_3)(x) - 2b(x)x = 2\beta^{-1},$$

which gives $\sigma(x) = \sqrt{2\beta^{-1}}$. Comparing it to the matrix representation of the generator analytically computed in Example 2.1.8, shows that the system is identified correctly.

(ii) To compute a representation of the Koopman generator for the double-well problem, we choose a dictionary consisting of monomials up to and including order 4 and generated 8000 random training points $x \in \mathbb{X} = [-2, 2] \times [-1, 1]$ and use exact values for $b(x)$ and $\sigma(x)$. The first six columns of the generator approximation are given by

	1	x_1	x_2	x_1^2	$x_1 x_2$	x_2^2
1	0	0	0	0.49	0	0.25
x_1	0	4	0	0	0.5	0
x_2	0	0	−2	0	0	0
x_1^2	0	0	0	9	0	0
$x_1 x_2$	0	0	0	0	2	0
x_2^2	0	0	0	0	0	−4
x_1^3	0	−4	0	0	0	0
$x_1^2 x_2$	0	0	0	0	0	0
$x_1 x_2^2$	0	0	0	0	0	0
x_2^3	0	0	0	0	0	0
x_1^4	0	0	0	−8	0	0
$x_1^3 x_2$	0	0	0	0	−4	0
$x_1^2 x_2^2$	0	0	0	0	0	0
$x_1 x_2^3$	0	0	0	0	0	0
x_2^4	0	0	0	0	0	0

Using columns two and three, we correctly recover the drift term b. Following (3.3), the entries of matrix a are given by

$$\begin{aligned}
a_{11}(x) &= (\mathcal{L}\psi_4)(x) - 2b_1(x)x_1 &&= 0.49 + x_1^2,\\
a_{12}(x) &= (\mathcal{L}\psi_5)(x) - b_1(x)x_2 - b_2(x)x_1 &&= 0.5x_1,\\
a_{22}(x) &= (\mathcal{L}\psi_6)(x) - 2b_2(x)x_2 &&= 0.25
\end{aligned}$$

which in fact is $\sigma\sigma^\top$. △

(iii) The representation of the Koopman generator for the quadruple-well problem follows the same way as for the double-well problem. Using 8000 random training points $x \in \mathbb{X} = [-2, 2] \times [-1, 1]$ and monomials up to and including order 4, we obtain the drift term b and matrix a with entries

$$\begin{aligned} a_{11}(x) &= (\mathcal{L}\psi_4)(x) - 2\,b_1(x)\,x_1 &&= 1 + x_1^2\,x_2^2, \\ a_{12}(x) &= (\mathcal{L}\psi_5)(x) - b_1(x)\,x_2 - b_2(x)\,x_1 &&= x_1\,x_2, \\ a_{22}(x) &= (\mathcal{L}\psi_6)(x) - 2\,b_2(x)\,x_2 &&= 1, \end{aligned}$$

which is $\sigma\,\sigma^\top$. Thus, the system is correctly identified. The full results can be found in Appendix A.1. Note that in the examples (ii) and (iii) monomials of order up to 3 are not sufficient to recover a. △

Remark 3.1.13. The following must still be mentioned:

(i) For the sake of illustration the considered systems are mainly composed of monomials. However, any dictionary containing twice continuously differentiable functions is possible.

(ii) The results depend on accurate estimates of the drift and diffusion terms. Noisy data leads to nonsparse solutions. Just as in the deterministic case, iterative hard thresholding can improve the results. As in Remark 3.1.4, we now add Gaussian noise with variance $\varsigma = 0.1$ to the drift and diffusion terms and apply hard thresholding with threshold $\delta = 0.1$. Then, we obtain the drift

$$b(x) = \begin{bmatrix} 4.00057\,x_1 - 4.00012\,x_1^3 \\ -1.99998\,x_2 \end{bmatrix}$$

and diffusion

$$\begin{aligned} a_{11}(x) &= 0.50035 + 0.99901\,x_1^2 - 0.00016\,x_1^4, \\ a_{12}(x) &= 0.49729\,x_1 - 0.00250\,x_1\,x_2 + 0.00097\,x_1^3\,x_2, \\ a_{22}(x) &= 0.25648 + 0.00720\,x_2^2 \end{aligned}$$

from the matrix approximation of the Koopman generator. Although the noise is also included in the diffusion term, which, as a consequence, might lead to an overestimation of the same, the obtained coefficients are still close to the analytical solution. Alternatively, iterative hard thresholding can directly be applied to the coefficients of $a_{ij}(x)$ to find a sparse representation of $a(x)$.

Just like SINDy, the method to discover the drift and diffusion terms of SDEs is subject to the same two difficulties that crucially affect the validity of the learned model:

(i) The dictionary needs to be rich enough such that b and a can be written in terms of the basis functions.

(ii) The drift and diffusion estimates must be available with sufficient accuracy.

Failure to meet either (i) or (ii), i.e., lack of expressivity of the dictionary or too noisy data, typically leads to nonsparse solutions. Using a larger set of training data or including further basis functions might mitigate these problems; however, the latter carrying the risk of overfitting. Notwithstanding, assuming that the model includes only a few simple terms, we are able to recover the governing equations of stochastic dynamical systems with the aid of the presented method. Furthermore, the approximation of the generator represents an important task itself since its eigenvalues and eigenfunctions provide information about time scales and metastable sets. Moreover, they can be used for model reduction and control. For further discussions on this topic we refer to [85, 89, 136] and references therein.

Conservation Laws

Also in the non-deterministic case, generator EDMD can also be used to approximate conserved quantities from data. If E is a conserved quantity of a non-deterministic system, then by the definition of the Koopman operator (2.3) and the partial differential equation $\frac{\partial u}{\partial t} = \mathcal{L}u$ it holds that $\mathcal{L}E = 0$. Thus, we can approximate conserved quantities by extracting non-trivial eigenfunctions associated with $\lambda = 0$.

To motivate the subsequent, assume that the stochastic model is driven by noise and that the noise term of the stochastic dynamical system has a small but positive correlation time, i.e., its autocorrelation function is not a delta function but a sharp peak. This implies that $\mathrm{d}W_t$ is regular and that the dynamical system has a well-defined solution. If the correlation time tends to zero, the SDE converges to a Stratonovich SDE given by

$$\mathrm{d}X_t = b(X_t)\,\mathrm{d}t + \sigma(X_t) \circ \mathrm{d}W_t, \tag{3.4}$$

where we use $\circ$ to distinguish (3.4) from SDEs of the form (2.2) in the sense of Itô [57, 131]. The difference between Stratonovich and Itô SDEs lies in the evaluation of the corresponding stochastic integral equation

$$X_t = X_0 + \int_0^t b(X_t)\,\mathrm{d}t + \int_0^t \sigma(X_t)\,\mathrm{d}W_t, \quad t \in [0,T].$$

The second integral cannot be defined uniquely as Riemann-Stieltjes integral since the Brownian motion has an unbounded variation. For an equidistant partition on $[0,T]$ with step size Δt such that $0 = t_0 < t_1 < \ldots < t_K = T$ we define

$$\int_0^t \sigma(X_t)\,\mathrm{d}W_t \coloneqq \lim_{K\to\infty} \sum_{k=0}^{K-1} \sigma\left((1-\alpha)\,X_k + \alpha\,X_{k+1}\right)(W_{k+1} - W_k), \quad \alpha \in [0,1].$$

For $\alpha = 0$ we call it *Itô stochastic integral* and we refer to an Itô SDE. For $\alpha = 0.5$ we obtain the *Stratonovich stochastic integral* and consequently a Stratonovich SDE. Other values of α exist. With the aid of the *drift correction formula* we convert a Stratonovich

SDE to an Itô SDE by correcting the noise-induced drift. It is defined component-wise as

$$c_i(x) = \sum_{j=1}^{d} \sum_{k=1}^{s} \frac{\partial \sigma_{ik}}{\partial x_j}(x)\, \sigma_{jk}(x), \quad i = 1, \ldots, d.$$

The corresponding Itô SDE reads

$$\mathrm{d}X_t = (b(X_t) + \frac{1}{2} c(X_t))\,\mathrm{d}t + \sigma(X_t)\,\mathrm{d}W_t,$$

see [160, 131] for further details. We will need the drift correction formula in Example 3.1.15 since the previous work has been carried for Itô SDEs.

Remark 3.1.14. Stratonovich SDEs allow an analogous formalism of conserved quantities as in the deterministic case. For a system of the form (3.4) a sufficient condition for E to be a conserved quantity is

$$\nabla E^\top \left[b + \sum_{i=1}^{s} \sigma_i \right] = 0,$$

where σ_i denotes the ith column of σ. This result follows directly from the chain rule of Stratonovich calculus, see [119, 49, 176].

Example 3.1.15. Let us consider the noisy Duffing oscillator, i.e., for $\alpha, \beta, \varepsilon \in \mathbb{R}$ we have a Stratonovich SDE of the form (3.4) with

$$b(x) = \begin{bmatrix} x_2 \\ -\alpha x_1 - \beta x_1^3 \end{bmatrix} \quad \text{and} \quad \sigma(x) = \varepsilon\, b(x).$$

Before we can apply gEDMD, we need to convert it to an Itô SDE by correcting the noise-induced drift given by

$$c(x) = \varepsilon^2 \begin{bmatrix} b_2(x) \\ (-\alpha - 3\beta x_1^2)\, b_1(x) \end{bmatrix}.$$

We set $\alpha = -1.1$, $\beta = 1.1$, $\varepsilon = 0.05$ and choose a dictionary that contains monomials. Then, applying gEDMD, the multiplicity of the eigenvalue $\lambda = 0$ is two and we obtain a conserved quantity of the form $E(x) \approx \frac{\alpha}{2} x_1^2 + \frac{\beta}{4} x_1^4 + \frac{1}{2} x_2^2 + c$, where $c \in \mathbb{R}$ is an arbitrary constant. Note that due to the Stratonovich formalism E has the same structure as in Example 3.1.5. △

3.2 Relationships between gEDMD and Other Methods

In the following, similarities and differences between the presented methods and other well-known approaches to system identification and generator approximation are highlighted.

3.2.1 SINDy

SINDy [20], which is an acronym for *sparse identification of nonlinear dynamics*, was developed to learn ODEs from data and marks a milestone for data-driven discovery of dynamical systems. Since then, various extensions have been developed such as parameter identification of stochastic systems [17] or the data-driven discovery of partial differential equations [146, 113], just to name a few applications. Just as gEDMD, the SINDy applies to a set of d-dimensional data vectors given by the state x_i and the corresponding time derivative $\dot{x}_i$. Note that unlike gEDMD, SINDy does not identify the transfer operator, but directly determines the governing equations of the dynamical system. Defining the matrix $\dot{X}$ as

$$\dot{X} = \begin{bmatrix} \dot{x}_1 & \cdots & \dot{x}_m \end{bmatrix},$$

we assume again that the evolution of the dynamical system can be described by a linear operator M such that $\dot{X} = M\Psi_X$. Omitting sparsification terms and solving the minimization problem $\|\dot{X} - M\Psi_X\|_F$ in least-square sense, we obtain the solution

$$M = \dot{X}\Psi_X^+ = (\dot{X}\Psi_X^\top)(\Psi_X\Psi_X^\top)^+ = \widehat{A}\widehat{G}^+$$

for matrices

$$\widehat{A} = \frac{1}{m}\sum_{l=1}^{m} \dot{x}_l\,\psi(x_l)^\top \quad \text{and} \quad \widehat{G} = \frac{1}{m}\sum_{l=1}^{m} \psi(x_l)\,\psi(x_l)^\top.$$

SINDy can be considered as a special case of gEDMD for deterministic systems due to the close connection between the vector field of a deterministic dynamical system and its Koopman generator. Assume that the full-state observable, which is given by $g(x) = B^\top\psi(x)$, is contained in the dictionary and let $\psi_k = x_j$. Then

$$\dot{\psi}_k = \sum_{i=1}^{d} b_i \frac{\partial \psi_k}{\partial x_i} = \sum_{i=1}^{d} b_i \frac{\partial x_j}{\partial x_i} = \sum_{i=1}^{d} b_i\,\delta_{ij} = b_j = \dot{x}_j$$

since all partial derivatives are zero except for $i = j$. Assuming $\psi = [g(x),\, \psi_{d+1}, \ldots, \psi_k]^\top$, the matrix $\dot{\Psi}_X$ can be written as

$$\dot{\Psi}_X = \begin{bmatrix} \dot{\Psi}_1 \\ \dot{\Psi}_2 \end{bmatrix} = \begin{bmatrix} \dot{X} \\ \dot{\Psi}_2 \end{bmatrix}$$

and it follows that

$$\dot{x} = B^\top\dot{\psi}(x) \approx B^\top M\psi(x) = \underbrace{B^\top\dot{\Psi}_X}_{\dot{X}}\Psi_X^+\psi(x) = \underbrace{\dot{X}\Psi_X^+}_{M_S}\psi(x) = M_S\,\psi(x).$$

3.2.2 KRONIC

KRONIC [75, 76], which stands for *Koopman reduced order nonlinear identification and control*, is derived from SINDy for the data-driven discovery of Koopman eigenfunctions, which are then used for control and the detection of conservation laws. For an a priori known Koopman eigenvalue (and using the notation of Section 3.1), KRONIC results in the eigenvalue problem

$$\left(\lambda_\ell \Psi_X^\top - \dot{\Psi}_X^\top\right) \xi_\ell = 0.$$

Alternatively, KRONIC can learn eigenvalues and the corresponding eigenfunctions simultaneously. If we multiply from the left by Ψ_X and assume regularity of $\Psi_X \Psi_X^\top$, this coincides with the deterministic gEDMD eigenvalue problem. Consequently, despite their different derivations, gEDMD and KRONIC are closely related for deterministic systems, which is also mentioned in [76].

3.2.3 Koopman Lifting Technique

The *Koopman lifting technique* [109, 110] is another methods that makes use of the infinitesimal generator $\mathcal{L}$ to discover the governing equations of dynamical systems. Although mainly developed for ODEs, extensions to SDEs with isotropic noise exist. The Koopman lifting technique works in two steps. First, standard EDMD is used to obtain an approximate representation of the Koopman operator $\widehat{K}^\tau$ from trajectory data for a fixed lag time τ and set of basis functions $\{\psi_i\}_{i=1}^n$. In the second step, the matrix logarithm is applied to obtain an approximation of the generator, that is

$$\widehat{L} = \frac{1}{\tau} \log \widehat{K}^\tau.$$

The system is then identified in the same way as demonstrated in Example 3.1.3 for gEDMD. As the Koopman lifting technique builds upon standard EDMD, neither time-derivatives of the states nor the partial derivatives of the basis functions are required. Instead, only pairs of τ-lagged data is needed. However, due to the non-uniqueness of the matrix logarithm, only a sufficiently small sampling time τ can guarantee that the (possibly complex) eigenvalues lie in the strip $\{z \in \mathbb{C} : |\Im(z)| < \pi\}$, where $\Im$ denotes the imaginary part. Roughly speaking, to capture the entire spectrum of frequencies an infinite sampling rate is necessary [110]. However, the independence of time-derivatives can be a useful alternative, e.g., when only trajectory data is accessible, since the presented estimation of the diffusion term can be transferred to the Koopman lifting technique. Practically this means that if time-derivatives have to be approximated from trajectory data, then the order of the finite-difference approximation and the step size crucially influence the accuracy of gEDMD, while the accuracy of the Koopman lifting technique depends mostly on the lag time (and the implementation of the matrix logarithm). For exact time-derivatives gEDMD has in general a higher accuracy than the Koopman lifting technique. Table 3.1 summarizes and compares the accuracies of

Table 3.1: Root mean square error for the system identification via gEDMD and the Koopman lifting technique (KLT) as a function of different lag times respectively step sizes τ and the order of finite-difference approximation for the dynamical systems as they appear in the Examples 2.1.3 and 3.1.5. For each system the dictionary consists of monomials up to and including degree 3 as well as trigonometric functions. We generated 5000 training points uniformly distributed in $\mathbb{X} = [-1, 1] \times [-1, 1]$ for both methods. For exact derivatives the error for gEDMD is numerically zero.

			Lag time resp. step size		
Dynamical system	Method	Order	$\tau = 0.1$	$\tau = 0.01$	$\tau = 0.001$
Simple system	KLT		6.97×10^{-4}	4.53×10^{-4}	6.75×10^{-6}
	gEDMD	2	$5.92 \times 10^{+0}$	3.39×10^{-2}	2.84×10^{-5}
		4	3.49×10^{-2}	4.45×10^{-4}	7.01×10^{-6}
		6	1.89×10^{-3}	4.60×10^{-4}	6.98×10^{-6}
Duffing oscillator	KLT		3.44×10^{-3}	2.65×10^{-4}	1.24×10^{-4}
	gEDMD	2	9.51×10^{-1}	2.54×10^{-2}	1.36×10^{-4}
		4	9.77×10^{-2}	2.68×10^{-4}	1.27×10^{-4}
		6	2.07×10^{-2}	2.83×10^{-4}	1.20×10^{-4}
Mathematical pendulum	KLT		2.80×10^{-3}	2.18×10^{-4}	1.30×10^{-5}
	gEDMD	2	1.59×10^{-1}	4.67×10^{-3}	1.38×10^{-5}
		4	3.33×10^{-3}	1.35×10^{-4}	1.03×10^{-5}
		6	3.48×10^{-4}	1.32×10^{-4}	1.08×10^{-5}

both approaches using the *root mean square error* (RMSE), which is defined by

$$\text{err} := \sqrt{\frac{1}{l}\sum_{i=1}^{l}(\widehat{y}_i - y_i)^2}, \tag{3.5}$$

where y_i and $\widehat{y}_i$ denote the measured quantity and its prediction, respectively.

4 Population Limits and Large Time Scales

In this chapter, we focus on agent-based systems formulated as continuous-time stochastic processes and their pathwise approximations by SDEs. We consider the case when the number of agents or the time scale of interest or both become large. By means of transfer operators, more specifically the Perron–Frobenius operator, we study for both the ABM and its SDE approximation the long-term behavior and the existence of metastable sets associated with it. Under certain conditions, the transfer operator approach bridges the pathwise approximation by differential equations for large population sizes on *finite* time scales and methods to characterize the long-term behavior on *exponentially long* time scales. We demonstrate that using the transfer operator approach it is possible to study the behavior on exponentially long time scales for medium or large population sizes by means of many *short* trajectories. Due to the pathwise closeness of the ABM and the SDE approximation, the structures determined using transfer operators are close for a sufficiently large number of agents, which can reduce the computational cost of analyzing ABMs. For reasons of comprehensibility, the considerations in this chapter are limited to ABMs acting on complete networks. The extended voter model introduced in Section 2.2.3 is used as guiding example.

The results presented in this chapter appeared in our publication [127], to which the author has made significant contributions.

4.1 Transfer Operator Approach

Metastability is a weak form of stability. A dynamical system is in a metastable state when it is in an apparently stable state, although the system can transition to a more stable state. Figuratively speaking, a bowling pin is in a metastable state if it wobbles without falling over. Obviously, if it fell over, it reached a more stable state.

Consider the extended voter model acting on a complete interaction network. For suitable chosen rate constants one opinion prevails in the population for a long period of time before another opinion emerges, replaces the old one and eventually dominates. Thus, the system exhibits a metastable behavior. In most cases the SDE limit model is capable to reproduce such metastable behavior for moderate to large population sizes. Clearly, the ODE limit model cannot reproduce it and fails in most cases. However, we will see that the SDE limit model, as it requires finite time scales (especially independent of the population size), may also be insufficient to fully explain this behavior.

To motivate the subsequent, we consider the Fokker–Planck equation (2.18). Assuming technical conditions regarding the growth of b and Σ at infinity, a unique invariant measure (stationary density) exists. However, it may exist only on the accessible part of

the state space. The SDE limit model is reversible, i.e., it satisfies the detailed balance condition with respect to this measure, if and only if there exists a smooth function V such that

$$b(c) = -\Sigma(c) \cdot \nabla V(c) + \frac{1}{2N} \nabla \cdot \Sigma(c).$$

This can be written more explicitly as

$$\sum_{k=1}^{K} \tilde{\alpha}_k(c) \left(\nu_k^\top \cdot \left[\nabla V - \frac{1}{2N} \nabla \log \tilde{\alpha}_k(c) \right] + 1 \right) \nu_k = 0. \tag{4.1}$$

If there exists a solution V of (4.1) with sufficient growth at infinity, the invariant measure $\hat{\mu}$ of the SDE limit model (precisely the density associated with it) is given by

$$\hat{\mu}(c) = \frac{1}{Z} \exp(-2NV(c)),$$

where Z is a normalization factor, see [175] for details. For further illustration, let us calculate the invariant measure for an agent-based system.

Example 4.1.1. We consider the voter model with N agents and two types of agents. Due to conservation, this is essentially a one-dimensional system and the solution of the SDE limit model (2.16) satisfies $C(t) = [c, 1-c]^\top$ with

$$\begin{aligned} \mathrm{d}c = &\Big[(\gamma_{21} - \gamma_{12})\, c(1-c) - \gamma'_{12}\, c + \gamma'_{21}(1-c) \Big] \mathrm{d}t \\ &+ \frac{1}{\sqrt{N}} \Big[- \sqrt{\gamma_{12}\, c(1-c)}\, \mathrm{d}W_1(t) + \sqrt{\gamma_{21}\, c(1-c)}\, \mathrm{d}W_2(t) \Big] \\ &+ \frac{1}{\sqrt{N}} \Big[- \sqrt{\gamma'_{12}\, c}\, \mathrm{d}W_3(t) + \sqrt{\gamma'_{21}(1-c)}\, \mathrm{d}W_4(t) \Big]. \end{aligned}$$

Using condition (4.1), it can be shown that this one-dimensional SDE is reversible. Assuming symmetric rate constants, that is, $\gamma_{12} = \gamma_{21} = \gamma > 0$, $\gamma'_{12} = \gamma'_{21} = \gamma' > 0$, and setting $\kappa := \frac{\gamma}{\gamma'}$, there exists a smooth function V satisfying (4.1), i.e.,

$$V(c) = \frac{1}{2N} \log(2\gamma\, c(1-c) + \gamma') - \frac{1}{2\kappa} \log(2\kappa\, c(1-c) + 1).$$

The invariant measure $\hat{\mu}$, depending on N, is then given by

$$\hat{\mu}(c) = \frac{1}{Z} \exp(-2NV(c)) = \frac{1}{Z\gamma'} \left(2\kappa\, c(1-c) + 1\right)^{\frac{N}{\kappa} - 1}.$$

Figure 4.1 shows the invariant measure for different numbers of agents. For small population sizes, precisely $N = 60$, it takes maxima for $c = 0$ and $c = 1$, while for large population sizes, i.e., $N = 1000$ and $N = 5000$, there exists a unique maximum at

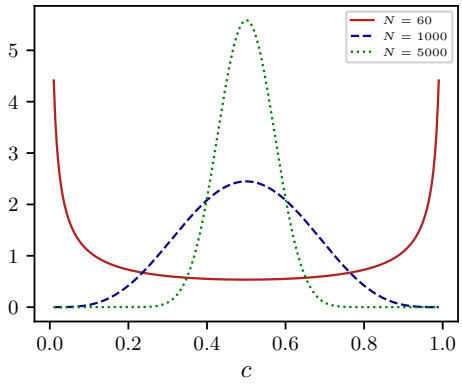


Figure 4.1: Invariant measure of the SDE limit model for $\gamma_{12} = \gamma_{21} = 1$, $\gamma'_{12} = \gamma'_{21} = 0.005$, and $\kappa = 200$ depending on different values of N (red/solid: $N = 60$, blue/dashed: $N = 1000$, green/dotted: $N = 5000$). Reproduced from [127].

$c = 0.5$. For larger N the concentration around the maximum gets sharper. In contrast to this, the ODE limit model, which is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}c = -2\gamma' c + \gamma',$$

has a unique stable fixed point at $c^* = 0.5$ to which it asymptotically converges. This means that for $N \to \infty$ both the ABM and the SDE limit vary around c^* and get close in finite time. The existence of the asymptotic stable fixed point of the ODE explains why the invariant measure of the SDE takes a unique maximum at c^* for large N. It *does not* explain the behavior for small N, i.e., why there exist maxima elsewhere for $N = 60$. Analogously, this holds for the ABM dynamics as well. We will observe that the full behavior of the dynamics can be explained by the existence of metastable sets. △

4.1.1 Transfer Operators and Metastability

For the subsequent considerations, we recall the transfer operators $\widetilde{\mathcal{P}}_N^t$ and $\mathcal{T}_N^t$ of the (rescaled) ABM process and the associated SDE limit model, respectively, introduced in Section 2.2.2. Let φ_ℓ denote the eigenfunctions of $\widetilde{\mathcal{P}}_N^t$ with corresponding eigenvalues λ_ℓ^t and analogously $\hat{\varphi}_\ell$ and $\hat{\lambda}_\ell^t$ for $\mathcal{T}_N^t$. Note that in this chapter eigenvalues are used according to Remark 2.1.2. We assume that both processes are geometrically ergodic on the respective space and that the respective measures μ and $\hat{\mu}$ exist. For the sake of comprehensibility we assume additionally reversibility of both dynamics, which, however, is not necessary. Under this assumption the transfer operators become self-adjoint in the Hilbert space L^2_π (with $\pi = \mu$ or $\pi = \hat{\mu}$, respectively) and hence have a real-valued spectral decomposition. This makes the following shorter and easier to

follow compared to the setting where reversibility is not assumed. In this case the more complicated complex alternatives like the Schur decomposition [40] or the singular value decomposition have to be used. According to Remark 2.1.4, $\lambda_1^t = 1$ is an isolated and time-independent eigenvalue. Further, it is the only eigenvalue with absolute value 1 and the corresponding eigenfunction is equal to the invariant measure, i.e., $\varphi_1 = \mu$. This holds for both transfer operators $\widetilde{\mathcal{P}}_N^t$ and $\mathcal{T}_N^t$.

We are now able to revisit metastability and recall Remark 3.1.10, which said that the dominant eigenvalues, in this chapter eigenvalues that are close to the maximal eigenvalue $\lambda = 1$, and the corresponding eigenfunctions can be used to identify metastable sets in the respective dynamics. Another important role of the dominant eigenfunctions lies in the theoretical justification of so-called *Markov state models* (MSMs), which are very low-dimensional reduced models of the full transfer operators with almost the same dominant eigenvalues [19, 153]. Given $\tilde{d} \ll d$ dominant eigenvalues of the transfer operator $\widetilde{\mathcal{P}}_N^t$, the Markov states of an MSM are constructed by non-negative ansatz functions ψ_k, $k = 1, \ldots, \tilde{d}$, which satisfy $\widetilde{\mathcal{P}}_N^t \psi_k \approx \psi_k$ as closely as possible. Thus, we can interpret them as macro-states, which are almost invariant under the dynamics. The ansatz functions ψ_k are chosen such that they form a partition of unity, i.e., $\sum_k \psi_k = \mathbb{1}$, and further, that the eigenvalues of the projected transfer operator $Q\widetilde{\mathcal{P}}_N^t Q$ are as close as possible to the dominant eigenvalues of the transfer operator $\widetilde{\mathcal{P}}_N^t$. The MSM is represented by a $\tilde{d} \times \tilde{d}$ matrix. Several algorithms for the computation of the almost-invariant ansatz functions ψ_k based on the dominant eigenfunctions of $\widetilde{\mathcal{P}}_N^t$ have been developed in the last years, such as set-oriented algorithms like milestoning [152] or PCCA+ [143], which approximate a basis of the dominant eigenspace.

All previous considerations can be applied to both, the ABM and SDE process independently of the number of agents N. Furthermore, we will see in the next two applications that for sufficiently large N the respective results of the MSM analysis are close for the ABM and SDE process and converge for $N \to \infty$.

Application: Metastable Behavior for Two Agent Types – Part 1

Let us revisit the voter model and the corresponding SDE limit model with two types of agents and set the transition rates to $\gamma_{12} = \gamma_{21} = 1$ and $\gamma'_{12} = \gamma'_{21} = 0.005$. Figure 4.2 (a) and (b) show typical long trajectories of the two processes for $N = 60$ agents. Both the ABM and the SDE process show a metastable behavior with at least two metastable areas which are located around $c = 0$ and $c = 1$. For increasing N and with respect to the same time interval, these two metastable areas vanish and the trajectories vary around the deterministic solution of the ODE limit model, see Figure 4.2 (c) and (d) for $N = 5000$. Short escapes from a narrow cylinder surrounding the ODE fixed point become increasingly rare. We observe this behavioral dependence of N also in Example 4.1.1.

We now want to explain this behavior with the existence of metastable sets. Therefore, we approximate the Perron–Frobenius transfer operators $\widetilde{\mathcal{P}}_N^\tau$ and $\mathcal{T}_N^\tau$ of the ABM and SDE processes for different numbers of agents, here $N = \{60, 100, 200, 1000\}$, and

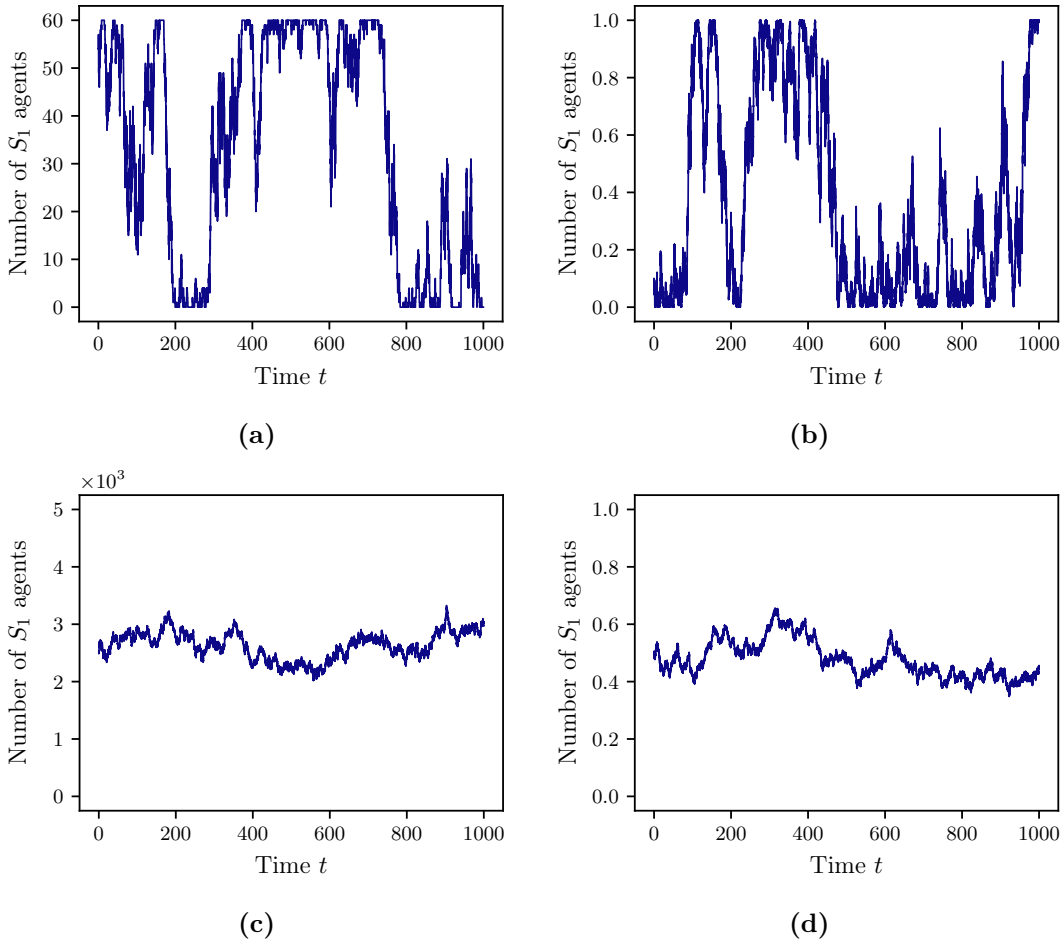


Figure 4.2: Realizations of (a) the ABM and (b) the SDE process for two types S_1 and S_2 of agents and $N = 60$ agents. (c) and (d) Realizations for $N = 5000$ agents. The rate constants are chosen as $\gamma_{12} = \gamma_{21} = 1$ and $\gamma'_{12} = \gamma'_{21} = 0.005$. The asymptotically stable fixed point of the ODE limit model is given by $c^* = [0.5, 0.5]^\top$. In contrast to the SDE limit model, the ODE limit model cannot approximate the ABM process as it cannot reproduce the typical metastable behavior. Figures (a) and (b) reprinted from [127].

time $\tau = 5$. To calculate the projected transfer operators $Q\widetilde{\mathcal{P}}_N^\tau Q$ and $Q\mathcal{T}_N^\tau Q$, we use Ulam's method and discretize the one-dimensional state space into n uniform boxes setting $n = 30$ for $N = 60$ and $n = 100$ for larger N. The entries of the respective stochastic discretization matrices are then approximated by (2.10) for $m_0 = 5000$ trajectories of length $\tau = 5$ per box for each process. Figure 4.3 (a) and 4.4 (a) show the invariant measures of the respective stochastic discretization matrices. Compared to the analytical results calculated in Example 4.1.1 for the SDE limit model, we observe

Table 4.1: Second and third eigenvalues for different numbers of agents N for the ABM and SDE limit model. Error estimates indicate that the error is about ± 0.01 in all cases. Reproduced from [127].

	ABM		SDE	
Number of agents N	λ_2^τ	λ_3^τ	$\hat{\lambda}_2^\tau$	$\hat{\lambda}_3^\tau$
60	0.9332	0.7204	0.9292	0.7002
100	0.9506	0.8131	0.9446	0.7932
200	0.9503	0.8591	0.9519	0.8579
1000	0.9510	0.8961	0.9492	0.8913

a very good approximation of the analytical result.

In the next step, we compute the dominant eigenvalues and eigenfunctions. For the discretized matrix associated with the ABM process and $N = 60$ agents the leading five eigenvalues are given by

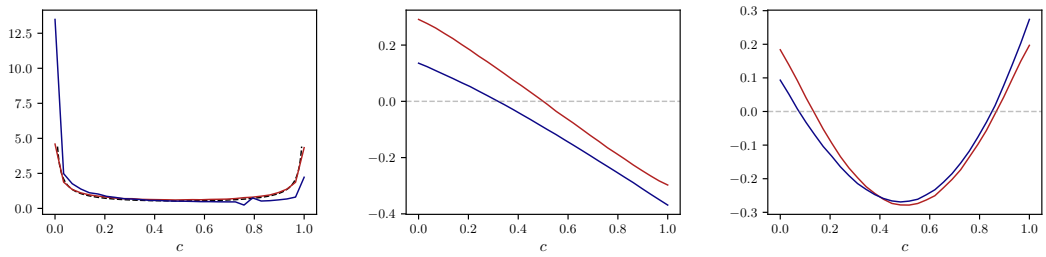
$$\lambda_1^\tau \approx 1, \quad \lambda_2^\tau \approx 0.93, \quad \lambda_3^\tau \approx 0.72, \quad \lambda_4^\tau \approx 0.46, \quad \lambda_5^\tau \approx 0.25.$$

For the SDE limit model we obtain

$$\hat{\lambda}_1^\tau \approx 1, \quad \hat{\lambda}_2^\tau \approx 0.93, \quad \hat{\lambda}_3^\tau \approx 0.70, \quad \hat{\lambda}_4^\tau \approx 0.43, \quad \hat{\lambda}_5^\tau \approx 0.21.$$

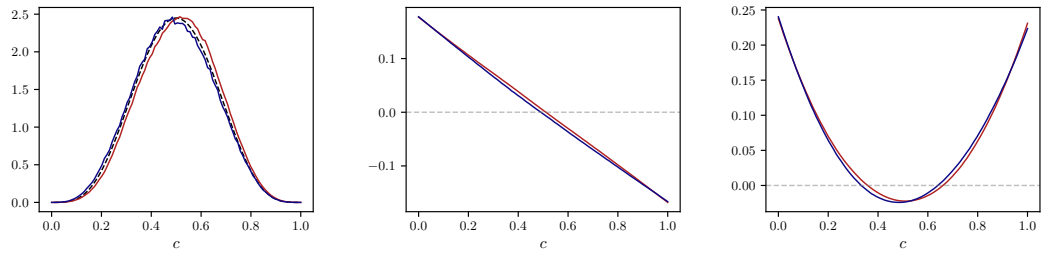
The corresponding eigenfunctions for the second and third eigenvalues are shown in Figure 4.3 (b) and (c). We observe that for both the ABM and SDE process the second eigenfunctions φ_2 and $\hat{\varphi}_2$ are qualitatively similar. Further, their respective zeros decompose the state space into the two metastable sets given by $A = [0, 0.5)$, and $B = [0.5, 1]$. The associated metastable time scale of $T_2 = -\tau / \log(\lambda_2^\tau) \approx 100$ seems to correspond to the typical transition behavior shown in Figure 4.2 (a) and (b). For the third eigenfunctions φ_3 and $\hat{\varphi}_3$ we observe again qualitative agreement. Their zeros reveal that also the approximate set $[0.2, 0.8]$ around the fixed point $c^* = 0.5$ of the ODE limit model is metastable with associated metastable time scale $T_3 \approx 14$.

The dominant eigenvalues for further values of N are summarized in Table 4.1. We observe that as N increases, the second eigenvalues remain (almost) the same, while the third eigenvalues increase with N. As a consequence, the associated metastable time scales T_2 does not change significantly, while the metastable time scale corresponding to the third eigenvalue increases from $T_3 \approx 14$ for $N = 60$ to $T_3 \approx 45$ for $N = 1000$. This means that the set around the fixed point $c^* = 0.5$ of the ODE process is increasingly metastable, i.e., rare events happen less frequently. The same observations can be made with the second and third eigenfunctions for the ABM and SDE process, which are shown in Figure 4.4 (b) and (c). For $N = 1000$ agents these are now very similar. The zeros of the second eigenfunctions still decompose the state space into the same metastable sets as for $N = 60$ agents. However, for $N = 1000$ the metastable set around the fixed point $c = 0.5$ has become significantly smaller, i.e., approximately $[0.35, 0.65]$.



(a) **(b)** **(c)**

Figure 4.3: (a) Invariant measures of the ABM and SDE process, respectively. The analytical result is indicated in black/dashed. (b) and (c) weighted eigenfunctions corresponding to the second and third eigenvalue, respectively. The eigenfunctions associated with the ABM process are shown in blue and for the SDE limit model in red. Rate constants are chosen as $\gamma_{12} = \gamma_{21} = 1$ and $\gamma'_{12} = \gamma'_{21} = 0.005$, for $N = 60$ agents. Figures (b) and (c) reprinted from [127].



(a) **(b)** **(c)**

Figure 4.4: (a) Invariant measures of the ABM and SDE process, respectively. The analytical result is indicated in black/dashed. (b) and (c) Weighted eigenfunctions corresponding to the second and third eigenvalue, respectively. The eigenfunctions associated with the ABM process are shown in blue and for the SDE limit model in red. Rate constants are chosen as $\gamma_{12} = \gamma_{21} = 1$ and $\gamma'_{12} = \gamma'_{21} = 0.005$, for $N = 1000$ agents. Figures (b) and (c) reprinted from [127].

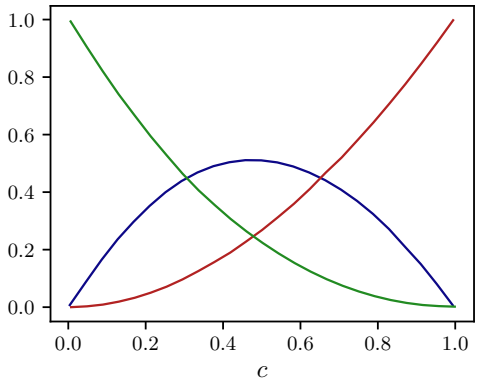


Figure 4.5: Three c-dependent quasi-stationary ansatz functions ψ_k, $k = 1, 2, 3$, used for constructing the MSM for the ABM process for $N = 1000$ agents. Reproduced from [127].

For further analysis of the three metastable sets, we build an MSM for the ABM process for $N = 1000$ agents via PCCA+ [143]. For three dominant eigenvalues, the matrix representation of the projected transfer operator is given by

$$Q\widetilde{\mathcal{P}}_N^{\tau}Q = \begin{bmatrix} 0.9506 & 0.0006 & 0.0488 \\ 0.0005 & 0.9525 & 0.0470 \\ 0.0271 & 0.0288 & 0.9441 \end{bmatrix}.$$

The eigenvalues $\lambda = \{1, 0.95, 0.89\}$ of matrix $Q\widetilde{\mathcal{P}}_N^{\tau}Q$ agree with the dominant eigenvalues up to two digits. Figure 4.5 shows the computed almost invariant ansatz functions ψ_k, $k = 1, 2, 3$. As the non-negative ψ_k form a partition of unity, they can be interpreted as almost *quasi-stationary distributions* that are almost invariant under the dynamics. However, they are not close to indicator functions of any of the sets[1] meaning that we cannot decompose the state space completely into three dominant metastable sets. The metastability of the three sets is too weak. Therefore, we apply further techniques in the following paragraph and application to increase the resolution of the underlying rare events.

Mean First Exit Times

Next, we want to determine the *expected mean first exit time* $\eta_N(i)$ for both processes to get further information about their behavior. The vector $\eta_N = (\eta_N(i))_{i \in I}$ of the expected mean first exit times $\eta_N(i)$ for starting in some box B_i, where $i \in I$ with $I = \{i = 1, \ldots, n : B_i \cap A = \emptyset\}$ and entering some box A can be obtained by solving

$$\frac{1}{\tau}(\mathrm{Id} - P_I^{\tau})\,\eta_N = \mathbf{1}_I. \tag{4.2}$$

[1] In many other scenarios, however, this is the case, see [153] for details.

Here P_I^τ denotes the submatrix of the $m \times m$ discretization matrix P^τ of the projected transfer operator $Q\widetilde{\mathcal{P}}_N^\tau Q$ (respectively $Q\mathcal{T}_N^\tau Q$) for indices $i \in I$, and $\mathbf{1}_I$ the vector of length $|I|$ with all entries equal to one. Note that the sets A and B_i do not need to be metastable sets or correlated with them. Equation (4.2) can be derived from the Galerkin discretization of the respective equation of the full transfer operator. The vector $\Phi_N = (\Phi_N(i))_{i\in I}$ of the associated rates can be computed via

$$\Phi_N(i) = \frac{1}{N}\log(\eta_N(i)). \tag{4.3}$$

We will calculate and compare the mean first exit times for both processes in the second part of the application.

Application: Metastable Behavior for Two Agent Types – Part 2

We again consider the voter model and its corresponding SDE limit model with two types of agents, but this time with asymmetric rates $\gamma_{12} = 1$, $\gamma_{21} = 1.1$, $\gamma'_{12} = 0.03$ and $\gamma'_{21} = 0.005$. The ODE limit model exhibits one stable fixed point $c^* \approx 0.7195$ in $[0, 1]$. The deviation of the fixed points can be found in Appendix A.2. As in the previous case with symmetric rate constants, both the ABM and the SDE process remain for large N and finite time intervals in a tight cylinder around the deterministic ODE solution.

Again, we approximate the transfer operators $\widetilde{\mathcal{P}}_N^\tau$ and $\mathcal{T}_N^\tau$ of the ABM and SDE processes for N in $\{60, 100, 200, 1000\}$ and time $\tau = 5$ via a box discretization of the state space into $n = 30$ for $N = 60$ and $n = 100$ boxes for larger N from $m_0 = 5000$ trajectories of the respective process per box. The first and second eigenfunctions of the projected transfer operators are shown in Figure 4.6 for $N = 1000$ agents. The invariant measures, i.e., first eigenfunctions, of both processes agree and are concentrated on the region around the fixed point c^*. Additionally, we observe that both eigenfunctions decompose the state space into two main metastable sets, where one is located on the left and the other on right of the fixed point c^*.

Let us now determine the expected mean first exit time $\eta_N(i)$ for starting in some box B_i for $i \in I$ and $I = \{i = 1, \dots, n : B_i \cap A = \emptyset\}$ and entering some set $A = [0.9, 1]$, which gives us also the expected mean first exit times $\eta_N(i)$ from $[0, 0.1]$ to $[0.9, 1]$. We compute the associated rates $\Phi_N(i)$ via Equation (4.3). Table 4.2 summarizes the rates $\Phi_N(i)$ for different numbers of agents N. We note two things: First, the mean first exit times η_N associated with the ABM and the SDE dynamics are remarkably similar and second, they grow exponentially with N. How these two observations can be explained with the help of large deviation theory, is discussed in Section 4.2.

4.1.2 Advantages and Limits

Let us now take a closer look at the advantages and limitations of the transfer operator approach. As a result of the pathwise closeness (2.17) of the ABM and SDE process, the discretization matrices $P^t_{\text{SDE}} = Q\mathcal{T}_N^t Q$ and $P^t_{\text{ABM}} = Q\widetilde{\mathcal{P}}_N^t Q$ of the transfer operators

Table 4.2: Exit rates Φ_N of the mean first exit times from $[0, 0.1]$ to $[0.9, 1]$ for different numbers of agents N for the ABM and SDE process. The rates Φ_N are marked as red crosses in Figure 4.10. Reproduced from [127].

	ABM		SDE	
Number of agents N	η_N	Φ_N	$\hat{\eta}_N$	$\hat{\Phi}_N$
60	458	0.0866	440	0.0784
100	540	0.0534	530	0.0515
200	874	0.0298	885	0.0293
1000	22000	0.0136	22100	0.0126

corresponding to both processes are close for large N. This implies

$$P^t_{\mathrm{SDE}} = P^t_{\mathrm{ABM}} + \mathcal{O}(1/N)$$

entrywise [47, 153] for all times $t \in [0, \tau]$ and (not too large) finite $\tau < \infty$. Given an n-dimensional ansatz space for the discretization, there exists a finite constant $\zeta_0 < \infty$ such that

$$\|P^\tau_{\mathrm{SDE}} - P^\tau_{\mathrm{ABM}}\|_1 \leq \zeta_0 \frac{n}{N}$$

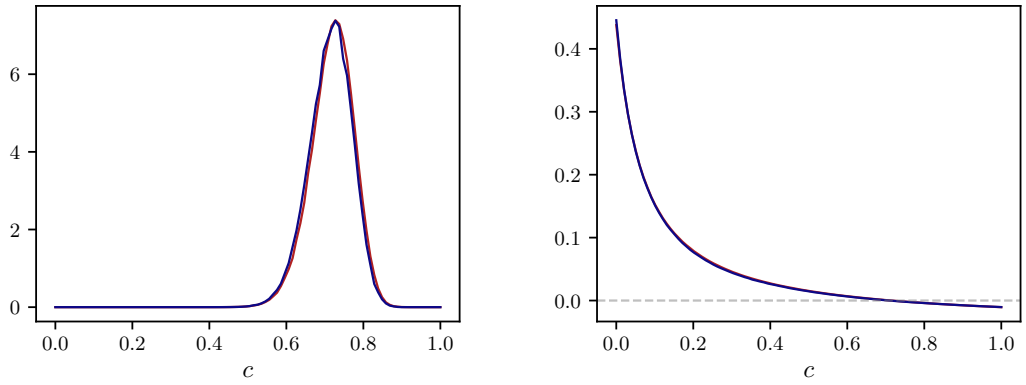
holds as an asymptotic result, i.e., there exists an $N_0 \in \mathbb{N}$ such that the statement holds for all $N > N_0$. This result implies that for sufficiently large N we may use the discretized SDE transfer operator to characterize the long-term behavior of the ABM process. Moreover, since only trajectories of length τ are needed, we can actually compute them in parallel for all boxes, which speeds up computation time.

However, as the discretizations P^t_{SDE} and P^t_{ABM} are computed using Ulam's method by means of trajectories via (2.10), we have to include an additional *sampling error* of order $\frac{1}{\sqrt{m_0}}$ in the number m_0 of trajectories. Moreover, as the ansatz space is finite-dimensional, the discretizations differ from the true transfer operators by the *discretization error* $\mathrm{err}_{\mathrm{discr}}$, see [147, 38]. Summarizing all results, we obtain in total

$$\|P^\tau_{\mathrm{SDE},m_0} - \widetilde{\mathcal{P}}^\tau_N\|_1 \leq \zeta_0 \frac{n}{N} + \zeta_1 \frac{n}{\sqrt{m_0}} + \mathrm{err}_{\mathrm{discr}}, \tag{4.4}$$

where $P^\tau_{\mathrm{SDE},m_0}$ denotes the numerical approximation of the discretized transfer operator P^τ_{SDE} given time τ. Figure 4.7 gives a graphical overview of the different errors appearing in Equation (4.4). Note that estimate (4.4) does not require any assumption of a spectral gap of $\widetilde{\mathcal{P}}^t_N$, but solely about time τ and the ansatz space, see [147, 148] for details on the approximation error of MSMs.

In summary, we conclude that even for large values of N, the resolution given by the underlying discretization limits the identification of metastable structures of the ABM process by studying the discretized transfer operator P^t_{SDE} associated with the SDE limit model. The resolution itself may be restricted by computational resources. Especially



(a) **(b)**

Figure 4.6: (a) Invariant measures (first eigenfunctions). (b) Weighted second eigenfunctions associated with the ABM (blue) and the SDE process (red) for $N = 1000$ agents. Rate constants are chosen as $\gamma_{12} = 1$, $\gamma_{21} = 1.1$, $\gamma'_{12} = 0.03$ and $\gamma'_{21} = 0.005$. Reprinted from [127].

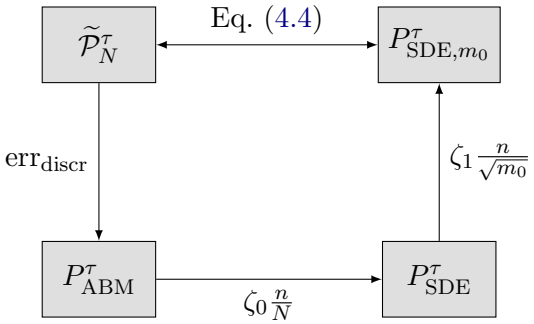


Figure 4.7: Connection between the true transfer operator $\widetilde{\mathcal{P}}_N^\tau$ of the ABM process, its discretization P^τ_{ABM}, the discretization P^τ_{SDE} associated with the SDE process and its numerical approximation $P^\tau_{\mathrm{SDE},m_0}$ for a given lag time τ.

if the metastable structures became finer and finer as N increases, the resolution needs to be increased to ensure a sufficient accuracy. While achieving a higher resolution by increasing the dimension n of the ansatz space with N, we may not be able to sufficiently reduce the first error term in Equation (4.4).

Remark 4.1.2. Equation (4.4), its consequences and in general this work assume that all types of agents are available in sufficient numbers. In the case of a sufficiently large number of agents (more generally, the system size), the SDE limit model is known to accurately approximate the chemical master equation [64]. However, if bi- or multistability stems from the discreteness of the system (that is, if the size of the system

is not large enough), the continuous approximation by the SDE process might fail to capture this noise-induced multi-stability of the discrete system, see [42, 67]. In these cases, hybrid modeling approaches may be used to approximate only parts of the process components by an SDE, see [172] for more details.

4.2 Large Deviations

In Section 4.1 we saw that the transfer operator approach reaches its limitations when rare events associated with metastability occur on time scales growing exponentially with the number of agents N. The necessary increase of the resolution by increasing the dimension n of the ansatz space is limited by computational resources and the approximation accuracy estimate (4.4). In order to avoid this problem, we now use large deviation theory which allows us to study these unlikely tail events. Using the voter model with two types of agents as example, we will see that under specific conditions there is an approximate agreement of the tail probabilities of the ABM process and the corresponding SDE limit model, although this is not necessary the case as shown in, e.g., [8]. As before in Section 4.1, we use the rescaled ABM process $C^{\varepsilon}_{\mathrm{ABM}}(t) \coloneqq \frac{1}{N} X^N(t)$ with smallness parameter $\varepsilon = 1/N$ with the associated master equation (2.14) for our analysis. Before giving a short introduction into large deviation theory, let us first take a look at an illustrative example.

Example 4.2.1. Consider the voter model with two types of agents and set the rate constants such that $\gamma'_{21} = 0$, $\delta = \gamma_{21} - \gamma_{12} > \gamma'_{12} > 0$. Then the ODE limit model exhibits two fixed points, where one is a stable fixed point at $c^*_{\mathrm{stable}} = 1 - \frac{\gamma'_{12}}{\delta}$ and the other an unstable fixed point at $c^*_{\mathrm{unstable}} = 0$. The second fixed point refers to the trapping state $c^*_{\mathrm{trap}} = [0, 1]$ of the ABM process. For every initial value $c_0 \in (0, 1]$ the trajectory of the ODE limit model converges asymptotically to the stable fixed point c^*_{stable}. For finite time intervals $[0, T]$ and sufficiently large N, the ABM and its SDE limit model remain in a narrow cylinder around the deterministic solution of the ODE process. However, for *very* large time intervals, i.e., T scales exponentially with N, both the ABM process and the associated SDE limit model will rarely deviate from the narrow cylinder around the stable fixed point, approach and, in the case of the ABM process, finally end up in the trapping state. This behavior can be seen in Figure 4.8 for the ABM. After more than 1.6×10^6 time steps the ABM process leaves the narrow cylinder around the stable fixed point and gets caught in the trapping state. △

4.2.1 Introduction to Large Deviation Theory

Large deviation theory studies the exponential decay of probabilities characterizing unlikely tail events for a large parameter; here the number of agents N. In the following, we give a brief introduction to large deviation theory tailored to the formulations of agent-based systems as introduced in Section 2.2.1. For a more general introduction to the topic we refer the reader to, e.g., [121].

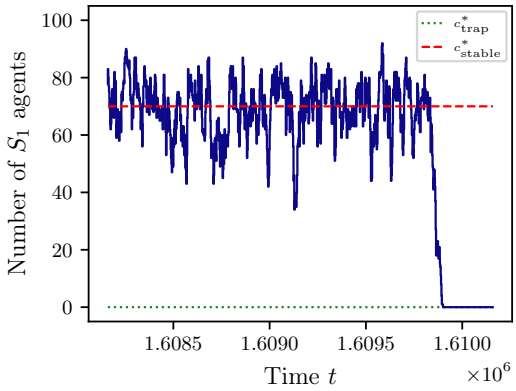


Figure 4.8: Realization of the voter model ending up in the trapping state c^*_{trap} (green/dotted) after staying close to the stable fixed point c^*_{stable} (red/dashed) for a very large time interval. The rate constants are set to $\gamma_{12} = 0.2280$, $\gamma_{21} = 0.3205$, $\gamma'_{12} = 0.0261$ and $\gamma'_{12} = 0$ and satisfy the condition in Example 4.2.1 such that the ODE process has two fixed points.

Let $C^\varepsilon(\cdot) = C^\varepsilon(t)$, $t \in [0,T]$ be an (arbitrary, random) path from a suitable path space $\mathcal{C}$ (e.g., $\mathcal{C} = H^1([0,T],\mathbb{R}^d)$) with initial value $C^\varepsilon(0) = c_0$ and $\varepsilon = 1/N$. Let the probability distribution generated by the master equation (2.14), respectively by the Markov jump process associated with it, or by the SDE limit equation (2.16) be denoted by $\mathbb{P}$. Given a *specific* path $c(\cdot) = c(t)$, $t \in [0,T]$ with initial value $c(0) = c_0$ from the path space $\mathcal{C}$ (e.g., the solution of the ODE limit model (2.15)) we call $\mathcal{I} : \mathcal{C} \to [0,\infty]$ the *large deviation rate function* associated with $\mathbb{P}$ if

$$\begin{aligned}
&\lim_{\delta\to 0} \liminf_{\varepsilon\to 0} \varepsilon \log \mathbb{P}\left(\sup_{t\in[0,T]} \|C^\varepsilon(t) - c(t)\| < \delta\right) \\
&= \lim_{\delta\to 0} \limsup_{\varepsilon\to 0} \varepsilon \log \mathbb{P}\left(\sup_{t\in[0,T]} \|C^\varepsilon(t) - c(t)\| < \delta\right) \\
&= -\mathcal{I}\left(c(t)\right)
\end{aligned}$$

for t in $[0,T]$, see [50]. This is written more intuitively as

$$\mathbb{P}\left(C^\varepsilon(\cdot) \approx c(\cdot)\right) \asymp \exp\left(-\frac{1}{\varepsilon}\mathcal{I}\left(c(\cdot)\right)\right), \tag{4.5}$$

and means that the probability to find an arbitrary random path $C^\varepsilon(\cdot)$ close to a specific path $c(\cdot)$ decays exponentially with asymptotic rate given by the product of smallness parameter $1/\varepsilon$ and rate function $\mathcal{I}$. Here, $\approx$ denotes δ-closeness of the paths $C^\varepsilon(\cdot)$ and $c(\cdot)$ and $\asymp$ *asymptotic equality* or *exponential equivalence*, that is

$$\varepsilon \log \mathbb{P}\left(C^\varepsilon(\cdot) \approx c(\cdot)\right) = -\mathcal{I}\left(c(\cdot)\right) + o(1),$$

where $x_\varepsilon = x + o(1)$ means that $(x_\varepsilon - x)/\varepsilon \to 0$ for $\varepsilon \to 0$. We call expression (4.5) a *large deviation principle*.

Alternatively, we can consider the following: the probability $p^\varepsilon(t_0, c_0, t_1, c_1)$ to go from c_0 at time t_0 to c_1 at time t_1 can be expressed via the path integral formalism by

$$p^\varepsilon(t_0, c_0, t_1, c_1) \asymp \int \exp\left(-\frac{1}{\varepsilon}\mathcal{I}\left(c(\cdot)\right)\right) \mathcal{D}c(\cdot),$$

where $\mathcal{D}c(\cdot)$ denotes integration over all paths $c(\cdot)$ with initial value $c(t_0) = c_0$ and final value $c(t_1) = c_1$ [18]. This means, that for small ε, the exponential factor $\exp(-\frac{1}{\varepsilon}\mathcal{I}\left(c(\cdot)\right))$ acts as probability density in $\mathcal{C}$ and that the rate function $\mathcal{I}$ is the path space measure introduced by the dynamics.

In many cases, we can obtain expressions for the rate function by constructing its pointwise form $I : \mathbb{R}^d \times [0, T] \to [0, \infty]$. In this case Equation (4.5) becomes

$$\mathbb{P}\left(C^\varepsilon(t) \approx c\right) \asymp \exp\left(-\frac{1}{\varepsilon} I(c, t)\right), \tag{4.6}$$

meaning that

$$\varepsilon \log \mathbb{P}\left(C^\varepsilon(t) \approx c\right) = -I(c, t) + o(1)$$

for all t in $[0, T]$. While $\mathcal{I}$ characterizes the associated probability distribution in the path space $\mathcal{C}$, the pointwise rate function I describes the asymptotic behavior of the probability distribution $\mathbb{P}$ induced by the dynamics in the state space $\mathbb{R}^d$ [127].

4.2.2 Large Deviation Rate Functions for the ABM and SDE Process

Next, we compute the pointwise rate functions I for the master equation (2.14) of the rescaled ABM process and the corresponding SDE limit model (2.16). We focus on the essential results. In many cases, there exist explicit techniques as the Gärtner–Ellis [44] or the Feng–Kurtz [50] methods to find the rate functions, see [121]. Besides large deviation principle, other asymptotic approaches like WKB theory or eikonal approximations, cf. [43, 8], allow the investigation of exponentially small probabilities for Fokker–Planck and master equations in the sense of (4.6).

The following results are based on our publication [127].

Large Deviation Rate Functions for the ABM Process

It was shown in [8] that the solution of the master equation (2.14) satisfies

$$\rho^\varepsilon_{\mathrm{ABM}}(c, t) \asymp \exp\left(-\frac{1}{\varepsilon} I_{\mathrm{ABM}}(c, t)\right)$$

for a rate function I_{ABM}. We can derive a Hamilton–Jacobi equation for I_{ABM} of the form

$$\partial_t I_{\mathrm{ABM}}(c, t) + \mathcal{H}_{\mathrm{ABM}}(c, \nabla I) = 0$$

with the Hamiltonian function

$$\mathcal{H}_{\mathrm{ABM}}(c,\xi) \coloneqq \sum_k \tilde{\alpha}_k(c)\left(\exp(\nu_k^\top \cdot \xi) - 1\right) \tag{4.7}$$

for $\xi \in \mathbb{R}^d$. The rate function I_{ABM} can be characterized by

$$\mathcal{I}_{\mathrm{ABM}}(c(t)) = I_{\mathrm{ABM}}(c_0, 0) + \int_0^T \mathcal{L}_{\mathrm{ABM}}(c(t), \dot{c}(t))\,\mathrm{d}t, \qquad t \in [0,T]$$

for $c(0) = c_0$ using the Lagrangian

$$\mathcal{L}_{\mathrm{ABM}}(c,v) \coloneqq \sup_\xi \left[v^\top \cdot \xi - \mathcal{H}_{\mathrm{ABM}}(c,\xi)\right].$$

However, in general the Lagrangian $\mathcal{L}_{\mathrm{ABM}}$ does not have an explicit form.

Mean First Exit Times. The infinitesimal generator $\widetilde{\mathcal{G}}^\varepsilon = \widetilde{\mathcal{G}}_N$ of the rescaled ABM process underlying the master equation (2.14) is given by

$$(\widetilde{\mathcal{G}}^\varepsilon f)(c) \coloneqq \frac{1}{\varepsilon}\sum_k \tilde{\alpha}_k(c)\left[f(c+\varepsilon\nu_k) - f(c)\right].$$

Let $\tau_{c_0}^\varepsilon$ denote the exit time of the ABM process from a bounded domain $D \subset \mathbb{R}^d$ with boundary ∂D starting in $c_0 \in D$. Then, for the mean first exit time $\eta^\varepsilon(c_0) = \mathbb{E}(\tau_{c_0}^\varepsilon)$ it holds that $\widetilde{\mathcal{G}}^\varepsilon \eta^\varepsilon = -1$ with boundary conditions $\eta^\varepsilon = 0$ on ∂D. We are interested in large deviations of the mean first exit time $\eta^\varepsilon(c_0)$ from a bounded domain D after starting in a state $c_0 \in D$. Therefore, we set

$$\eta^\varepsilon(c_0) \asymp \exp\left(\frac{1}{\varepsilon}\Phi_{\mathrm{ABM}}(c_0)\right)$$

for a rate function $\Phi_{\mathrm{ABM}} : \mathbb{R}^d \to [0,\infty]$, which has to be interpreted as

$$\liminf_{\varepsilon\to 0} \varepsilon \log \eta^\varepsilon(c_0) = \limsup_{\varepsilon\to 0} \varepsilon \log \eta^\varepsilon(c_0) = \Phi_{\mathrm{ABM}}(c_0).$$

The large deviation theory then allows us to obtain

$$\sum_k \tilde{\alpha}_k(c)\left(\exp(\nu_k^\top \cdot \nabla\Phi_{\mathrm{ABM}}(c_0)) - 1\right) = 0$$

as an expression for the rate function Φ_{ABM} [121]. We can further write it using the Hamiltonian $\mathcal{H}_{\mathrm{ABM}}$ defined in (4.7), i.e.,

$$\mathcal{H}_{\mathrm{ABM}}(c_0, \nabla\Phi_{\mathrm{ABM}}(c_0)) = 0.$$

This implies that we can determine the rate function Φ_{ABM} for the mean first exit time via the curves $\mathcal{H}_{\mathrm{ABM}} = 0$ in the phase portrait of the Hamiltonian system associated with the master equation (2.14). Therefore, the Hamiltonian $\mathcal{H}_{\mathrm{ABM}}$ is crucial for the characterization of both the large deviation rate function and the (exponentially large) mean first exit time. Before computing the rate function Φ_{ABM} explicitly in Section 4.2.3 using the Hamiltonian $\mathcal{H}_{\mathrm{ABM}}$, we repeat the previous considerations for the SDE process.

Large Deviation Rate Functions for the SDE Process

We consider a rate function I_{SDE} for the SDE process (2.16) with the density $\rho^{\varepsilon}_{\mathrm{SDE}}(c,t)$ such that

$$\rho^{\varepsilon}_{\mathrm{SDE}}(c,t) \asymp \exp\left(-\frac{1}{\varepsilon} I_{\mathrm{SDE}}(c,t)\right).$$

Again, there exists a Hamiltonian function $\mathcal{H}_{\mathrm{SDE}}$ such that we have an equation of the form [8, 121]

$$\partial_t I_{\mathrm{SDE}}(c,t) + \mathcal{H}_{\mathrm{SDE}}(c, \nabla I_{\mathrm{SDE}}) = 0.$$

Both rate functions I_{ABM} and I_{SDE} have the same minimum curve, which is given by the solution of the ODE limit model. Moreover, for small ξ, it holds that

$$\mathcal{H}_{\mathrm{ABM}} = \mathcal{H}_{\mathrm{SDE}} + \mathcal{O}(\|\xi\|^3), \tag{4.8}$$

which implies that the associated Hamiltonian of the SDE limit model is the second-order accurate approximation of the Hamiltonian of the ABM process near the ODE limit curve.

If the matrix Σ defined in the Fokker–Planck equation (2.18) is invertible, we can directly compute the associated Lagrangian [53, 18]. In this case, large deviation rate function $\mathcal{I}_{\mathrm{SDE}}$ on path space of the SDE system can be expressed as

$$\mathcal{I}_{\mathrm{SDE}}\left(c(t)\right) = \frac{1}{2}\int_0^T \left(\dot{c}(t) - b(c(t))\right)^{\top} \Sigma(c(t))^{-1} \left(\dot{c}(t) - b(c(t))\right) \mathrm{d}t$$

for paths $c(t) \in H^1([0,t], \mathbb{R}^d)$ with $t \in [0,T]$ that start in the initial state c_0. We set $I_{\mathrm{SDE}}(c_0, 0) = 0$ as initial condition.

Mean First Exit Times. As before for the ABM process, we consider the rate function Φ_{SDE} for the mean first exit time from the given domain of the SDE process. The infinitesimal generator $\mathcal{G}^{\varepsilon} = \mathcal{G}_N$ corresponding to the SDE process (2.16) is given by

$$(\mathcal{G}^{\varepsilon} f)(c) = -\sum_{i=1}^{d} b_i(c) \frac{\partial}{\partial c_i} f(c) + \frac{\varepsilon}{2} \sum_{i=1}^{d} \sum_{j=1}^{d} \Sigma_{ij}(c) \frac{\partial^2}{\partial c_i \partial c_j} f(c)$$

and in this case we get [18]

$$b(c)^{\top} \cdot \nabla \Phi_{\mathrm{SDE}}(c) + \frac{1}{2} \nabla \Phi_{\mathrm{SDE}}(c)^{\top} \cdot \Sigma(c) \nabla \Phi_{\mathrm{SDE}}(c) = 0.$$

Again, the Hamiltonian function $\mathcal{H}_{\mathrm{SDE}}$ can be expressed by

$$\mathcal{H}_{\mathrm{SDE}}(c, \nabla \Phi_{\mathrm{SDE}}(c)) = 0.$$

4.2.3 Application: Explicit Rate Functions for Specific Propensities

After these preparations we derive explicit rate functions for the voter model. We consider again two types of agents. The total (rescaled) propensity function for the extended voter model is given by

$$a_{ij}(c) \coloneqq \alpha_{ij}(c) + \alpha'_{ij}(c) = \gamma_{ij}\, c_i\, c_j + \gamma'_{ij}\, c_i$$

for $(i,j) = \{(1,2),(2,1)\}$ with net change vectors $\nu_{12} = -\nu_{21} = [-1,1]^\top$.

Explicit Rate Function for the ABM Process

The Hamiltonian associated with the master equation (2.14) is given by

$$\mathcal{H}_{\mathrm{ABM}}(c,\xi) = a_{12}(c)\,(\exp(\xi_2 - \xi_1) - 1) + a_{21}(c)\,(\exp(\xi_1 - \xi_2) - 1)$$

and the Lagrangian is defined as

$$\mathcal{L}_{\mathrm{ABM}}(c,v) = \sup_{\xi} \left[v^\top \cdot \xi + a_{12}(c)\,(1 - \exp(\xi_2 - \xi_1)) + a_{21}(c)\,(1 - \exp(\xi_1 - \xi_2)) \right].$$

Due to the conservation property $c_1 + c_2 = 1$ for all solutions of the master equation, it holds for the Lagrangian that

$$\mathcal{L}_{\mathrm{ABM}}(c,v) = \begin{cases} \mathcal{L}_{\mathrm{ABM},1}(c_1, v_1), & v_1 + v_2 = 0, \quad c_2 = 1 - c_1 \\ \infty, & \text{otherwise,} \end{cases}$$

where $\mathcal{L}_{\mathrm{ABM},1}$ denotes the reduced Lagrangian and is given by

$$\mathcal{L}_{\mathrm{ABM},1}(c,v) = \sup_{\Delta\xi} \left[\Delta\xi v + a_{12}(c)\,(1 - \exp(-\Delta\xi)) + a_{21}(c)\,(1 - \exp(\Delta\xi)) \right]$$

for $\Delta\xi \coloneqq \xi_1 - \xi_2$, c and v one-dimensional, and $a_{ij}(c) = a_{ij}((c, 1-c))$. The reduced Lagrangian can be computed explicitly, which results in

$$\begin{aligned} \mathcal{L}_{\mathrm{ABM},1}(c,v) = v \log \left[\frac{1}{2\,a_{21}(c)} (v + \sqrt{v^2 + 4\,a_{12}(c)\,a_{21}(c)}) \right] \\ + a_{12}(c) + a_{21}(c) - \sqrt{v^2 + 4\,a_{12}(c)\,a_{21}(c)}. \end{aligned}$$

Along the trajectory of the reduced ODE limit model, i.e.,

$$v = \dot{c} = (\gamma_{21} - \gamma_{12})\,c(1-c) + \gamma'_{21}(1-c) - \gamma'_{12}\,c \tag{4.9}$$

it holds that $\mathcal{L}_{\mathrm{ABM},1} = 0$. The corresponding reduced Hamiltonian is given by

$$\mathcal{H}_{\mathrm{ABM},1}(c,\Delta\xi) = a_{12}(c)\,(\exp(-\Delta\xi) - 1) + a_{21}(c)\,(\exp(\Delta\xi) - 1)$$

and we obtain the curves for $\mathcal{H}_{\mathrm{ABM},1}(c,\Delta\xi) = 0$:

$$\Delta\xi = 0, \qquad \Delta\xi(c) = \log \frac{a_{12}(c)}{a_{21}(c)}. \tag{4.10}$$

Figure 4.9 (a) shows the phase portrait of $\mathcal{H}_{\mathrm{ABM},1}$ and the two curves (4.10) marked in green and red.

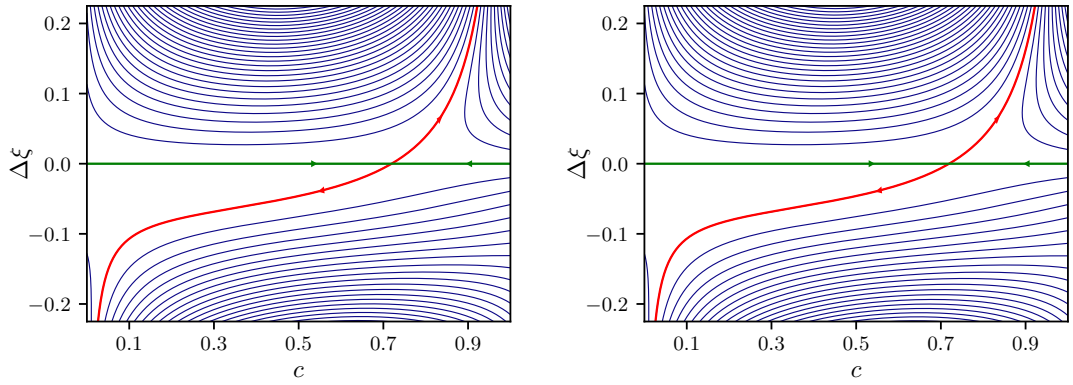


Figure 4.9: (a) Phase portrait of the reduced Hamiltonian $\mathcal{H}_{\mathrm{ABM},1}$ with curves (4.10) (green and red) for the ABM process. (b) Phase portrait of the reduced Hamiltonian $\mathcal{H}_{\mathrm{SDE},1}$ with curves (4.11) (green and red) for the SDE limit model. The arrows indicate the direction of the temporal evolution. Reprinted from [127].

Explicit Rate Function for the SDE Process

As a consequence of the conservation property, for the SDE process in this application its holds that matrix Σ, which is given by

$$\Sigma = (a_{12} + a_{21}) \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix},$$

is not positive. In fact, it has an eigenvalue 0. This implies that we have to use a reduced Lagrangian $\mathcal{L}_{\mathrm{SDE},1}$ given by

$$\begin{aligned} \mathcal{L}_{\mathrm{SDE},1}(c, v) &= \max_{\Delta\xi} \left(v\Delta\xi + (a_{12}(c) - a_{21}(c))\Delta\xi - \frac{1}{2}\Delta\xi^2 \right) \\ &= \frac{1}{2} \left(a_{12}(c) - a_{21}(c) + v \right)^2 . \end{aligned}$$

Again, along the trajectory of the reduced ODE limit equation (4.9) it holds that $\mathcal{L}_{\mathrm{SDE},1} = 0$. The associated reduced Hamiltonian is given by

$$\mathcal{H}_{\mathrm{SDE},1}(c, \Delta\xi) = (a_{21}(c) - a_{12}(c))\,\Delta\xi + \frac{1}{2}(a_{12}(c) + a_{21}(c))\,\Delta\xi^2$$

and the curves for $\mathcal{H}_{\mathrm{SDE},1}(c, \Delta\xi) = 0$ are given by

$$\Delta\xi = 0, \qquad \Delta\xi(c) = 2\frac{a_{12}(c) - a_{21}(c)}{a_{12}(c) + a_{21}(c)}. \tag{4.11}$$

Additionally, we observe that $\mathcal{H}_{\mathrm{SDE},1} = \mathcal{H}_{\mathrm{ABM},1} + \mathcal{O}(\Delta\xi^3)$ in accordance with Equation (4.8). Figure 4.9 (b) shows the phase portrait of $\mathcal{H}_{\mathrm{SDE},1}$ and the two curves (4.11) marked in green and red. Comparing both phase portraits and the zero-curves shown in Figure 4.9 (a) and (b), we see that they are visually indistinguishable, which implies that the respective large deviation rates for the mean first exit times are almost identical.

Example 4.2.2. Consider again the voter model with two types of agents and the same asymmetric rate constants $\gamma_{12} = 1$, $\gamma_{21} = 1.1$, $\gamma'_{12} = 0.03$ and $\gamma'_{21} = 0.005$ as in Section 4.1.1 (Application, Part 2). In this setting we can explicitly compute the large deviation rate functions. For instance, to calculate the rate of the mean first exit time for passing from $c_0 = 0.1$ to $c_1 = 0.9$, i.e., facing a growth from a minority of 10 % of the agents of type S_1 to a majority of 90 %, we have to do the following: Starting in state c_0 we follow the green curve $\Delta\xi = 0$ corresponding to the ODE limit model until the fixed point $c^* \approx 0.7195$. Afterwards, we have to act against the ODE limit process and follow the red curve. As the considered model is one-dimensional, the rate function is given by

$$\Phi(c_0 \to c_1) = \int_{c_0}^{c^*} \Delta\xi \,\mathrm{d}c + \int_{c^*}^{c_1} \Delta\xi \,\mathrm{d}c \approx 0 + 0.0102 = 0.0102,$$

which results from $\nabla\Phi = \Delta\xi$, see [18]. Inserting the zero-curves (4.10) and (4.11) for the ABM and SDE process, respectively, yields the same value for both rates. Comparing these results to the empirical rate $\varepsilon \log(\eta^\varepsilon(c_0 \to c_1))$ estimated from 1000 realizations of the ABM and the SDE process, we observe that the empirical rate converges to the large deviation rate $\Phi(c_0 \to c_1)$ for $\varepsilon \to 0$, i.e., $N \to \infty$. Additionally, the empirical rate of the ABM and the SDE process are almost identical even for small N. Moreover, we also observe quantitative agreement with the rates of the mean first exit times computed in Section 4.1 using the associated transfer operators, see Table 4.2. Figure 4.10 summarizes the results, where the values of Table 4.2 are indicated by red markers. △

Remark 4.2.3. Along the curves $\Delta\xi = 0$ the temporal derivative $\frac{\mathrm{d}}{\mathrm{d}t}c = \frac{\partial\mathcal{H}}{\partial\Delta\xi}$ is positive on the left hand side of the fixed point c^* and negative on the right hand side, while the temporal derivative $\frac{\mathrm{d}}{\mathrm{d}t}\Delta\xi = -\frac{\partial\mathcal{H}}{\partial c}$ is constant 0. Thus, with the ODE limit model, we always end up at the fixed point c^*, no matter which side we approach from. Along the red curves for, e.g., $\Delta\xi > 0$, both derivatives $\frac{\mathrm{d}}{\mathrm{d}t}\Delta\xi$ and $\frac{\mathrm{d}}{\mathrm{d}t}c$ are positive and we follow the red curve upwards to c_1. In Figure 4.9 the direction of the temporal evolution is indicated by arrows.

4.2.4 Deviations of the SDE Rates from the ABM Rates

For the voter model analyzed in Section 4.2.3 the Hamiltonian's phase portraits of the ABM process and its SDE limit model are visually indistinguishable, see Figure 4.9, so that the resulting mean first exit rates are almost identical. However, as the

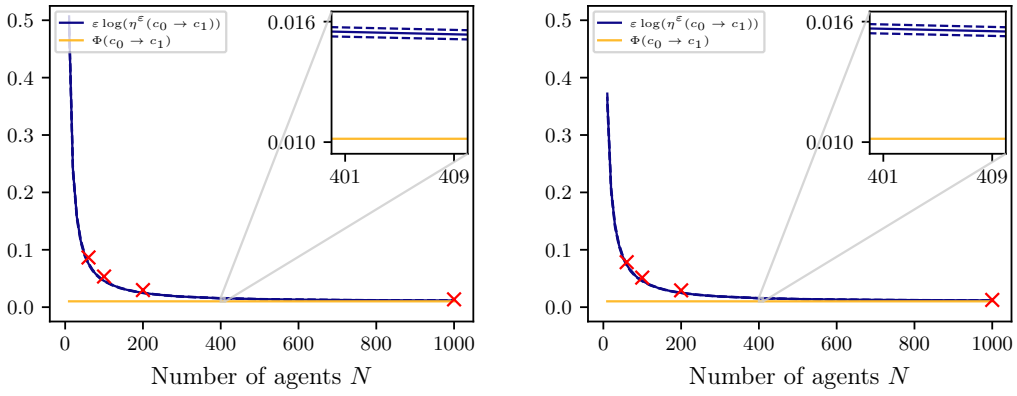


Figure 4.10: Mean time $\eta^{\varepsilon}(c_0 \to c_1)$ to pass from $c_0 = 0.1$ to $c_1 = 0.9$ for (a) the ABM process and (b) the SDE limit model compared to the value $\Phi(c_0 \to c_1) = 0.0102$ of the rate function. The dashed lines mark the confidence interval for confidence level 0.999. The values of Table 4.2 are indicated by red markers. Reprinted from [127].

Hamiltonian of the SDE limit model is only a second order accurate approximation in $\Delta\xi = \xi_1 - \xi_2$ of the ABM process, there may be deviations for large $\Delta\xi$, which lead to exponential deviations in the exit times. In these cases, the SDE process fails to quantitatively capture the dynamics of the ABM process. Therefore, characterization of metastable behavior via the SDE process is in general insufficient to quantitatively understand the metastability of the ABM process.

5 Data-Driven Model Reduction of Agent-Based Systems

In Chapter 4 we saw that the long-term characteristic behavior of agent-based systems can be determined by simulating (many) short trajectories of the corresponding SDE limit models instead. In this chapter, our goal is to illustrate how we can use the Koopman generator to obtain reduced models represented by ordinary or stochastic differential equations and infer the physical laws of complex agent-based system from aggregated trajectory data that represents the collective behavior of larger groups or the entire population. The approach is based on the methods presented in Chapter 3, with the difference that here we *directly* learn reduced models from (highly noisy) data generated by an ABM. Since we know the resulting limit processes for agent-based systems defined on complete networks such as the voter model, we can compare the numerical results obtained for finitely many agents with the theoretical results and demonstrate that under appropriate conditions the estimated models are in good agreement with known limit cases. We show that the obtained reduced models are in good agreement with the corresponding SDE limit models for large population sizes and can be used for both system identification and predictions of the temporal evolution. Furthermore, we demonstrate that the proposed approach also yields good reduced models that allow prediction in some other cases where the limit process is unknown or the system is even far from a limit case. This is shown for the voter model on incomplete, clustered interaction networks as well as for agent-based systems that do not have a network-based formulation such as the predator-prey model or civil violence model. In general, this approach requires a large amount of data, which, however, is not a problem in simulation studies, where a surrogate model is required for the optimization or control of the full-complexity ABM.

The results presented in this chapter appeared in our publication [126].

5.1 Learning Reduced Models from Noisy Data

Although the presented approach is based on Chapter 3 and our publication [85], learning a reduced model for an agent-based system from (noisy) aggregated trajectory data still remains a non-trivial task. Firstly, we need to estimate the drift and diffusion terms *pointwise*, cf. Section 3.1. Secondly, we obtain *global* descriptions of the drift and diffusion terms using the Koopman generator, cf. Section 3.1.1. The necessarily large number of agents and the often counterintuitive choice of basis function pose further problems. The proposed approach is sketched in Figure 5.1. We will now go through

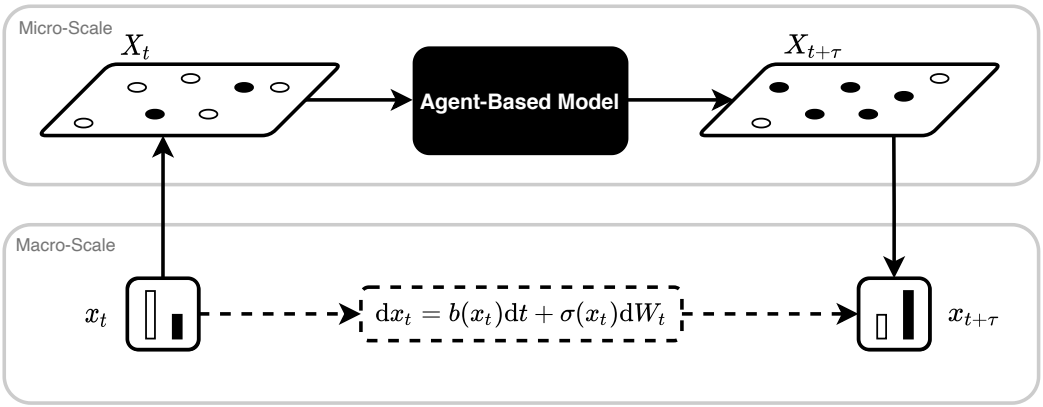


Figure 5.1: On the micro-scale level the agent-based system computes state $X_{t+\tau}$ given an initial state X_t. On the macro-scale level we need to determine a reduced model represented by an SDE such that the aggregate state $x_{t+\tau}$ associated with $X_{t+\tau}$ can be computed from x_t (associated with X_t).

the main steps necessary to learn the Koopman generator from data generated by an ABM.

Measurements. We consider an ABM and assume that we have access to m measurements of an aggregate state variable representing agents sharing, e.g., the same type S_i or belonging to some group. Let us denote these measurements by $\{x_l\}_{l=1}^m$. To ensure a good coverage of the whole aggregate space, we choose the measurements x_l such that they are uniformly distributed in the (aggregate) state space $\mathbb{X}$. Note that $\mathbb{X}$ is not necessarily the population state space (2.11). To achieve a good coverage of the aggregate state space $\mathbb{X}$, one option is to construct an appropriate map from the macroscopic (aggregate) state $x_t = x_l(t)$ to the microscopic ABM state $X_t = X_l(t)$, $l = 1, \ldots, m$. By *appropriate* we mean that the mapped macroscopic state x_t and a naturally evolved ABM state X_t with same aggregate variables agree in probability. In practice, this means that if, e.g., the agents follow a certain spatial distribution, this must be taken into account when constructing the map. Alternatively, we can gather the measurements "on the fly", i.e., by using the full ABM states belonging to aggregate trajectories, which can be obtained from the simulation of the ABM. The price to pay for this rather straightforward approach is that it might not result in a good coverage of the entire aggregate space. The mapping back from micro-scale to macro-scale consists in most cases only of determining the different group sizes. At this point we know the (macroscopic) aggregate states x_t and $x_{t+\tau}$ associated with the microscopic states X_t and $X_{t+\tau}$, respectively.

Pointwise Estimates. The overall goal is to find an SDE of the form (2.2), which represents a reduced model of the ABM. However, along with the simplicity of ABMs

comes that the drift and diffusion terms b and σ are inherently unknown such that we need accurate estimate of them. This can be achieved via pointwise finite difference approximations for each measurement $\{x_l\}_{l=1}^m$ using the *Kramers–Moyal formulae*

$$b(x) \coloneqq \lim_{\tau \to 0} \mathbb{E}\left[\frac{1}{\tau}(X_\tau - x) \;\middle|\; X_0 = x\right], \tag{5.1}$$

$$a(x) \coloneqq \lim_{\tau \to 0} \mathbb{E}\left[\frac{1}{\tau}(X_\tau - x)(X_\tau - x)^\top \;\middle|\; X_0 = x\right]. \tag{5.2}$$

The above formulae can be derived from the Kramers–Moyal expansion, see, e.g., [142]. We evaluate both expressions by Monte Carlo methods via multiple short trajectories of the ABM at each data point $\{x_l\}_{l=1}^m$ (using the map between macro-scale and micro-scale level). The simulation of several short realizations of the full ABM is similar to the so-called *equation-free approach*. We also applied it previously in Chapter 4 to approximate the Perron–Frobenius operator of an ABM and the corresponding SDE limit process. It is important to emphasize that these *pointwise* estimates of the drift and diffusion for each training data point form the first step in obtaining a *global* description of them. We now know the (macroscopic) aggregate states $\{x_l\}_{l=1}^m$ and the pointwise drift and diffusion estimates $\{b(x_l)\}_{l=1}^m$ and $\{a(x_l)\}_{l=1}^m$.

Conservation. If the aggregate state is subject to conservation, we have only $d-1$ degrees of freedom and the aggregated trajectory data belongs to a $d-1$ dimensional system. For instance, if the number of agents N is constant for all time $t \geq 0$, the aggregate state $x_j(t)$ (e.g., representing the number of agents of type S_j) can be expressed by

$$x_j(t) = N - \sum_{i \neq j} x_i(t).$$

Thus, without loss of generality, we reduce each measurement by keeping only the first $d-1$ components. This way we eliminate redundant representations of the system. Additionally, we can obtain a frequency representation $c_i(t) = x_i(t)/N$ by scaling the measurements by the number of agents N.

Basis Functions. Next, we need to select a set of basis functions $\{\psi_i\}_{i=1}^n$, which, again, is a non-trivial but crucial step, since in general it is not clear how the drift term b and diffusion term σ of the SDE (2.2) look like. Assuming that the SDE approximation of the ABM adopts to the model structure introduced in Section 2.2 and comprises at most pth order transitions, we can show that a dictionary consisting of monomials up to and including degree $p+1$ is sufficient to correctly identify the model of the form (2.16). To show this, we rewrite the SDE (2.16) such that it takes on the form of (2.2), i.e.,

the drift and diffusion terms $b(c)$ and $\sigma(c)$ are given by

$$b(c) = \sum_{k=1}^{K} \widetilde{\alpha}_k(c)\,\nu_k, \tag{5.3}$$

$$\sigma(c) = \frac{1}{\sqrt{N}} \begin{bmatrix} \sqrt{\widetilde{\alpha}_1(c)}\,\nu_1 & \dots & \sqrt{\widetilde{\alpha}_K(c)}\,\nu_K \end{bmatrix}. \tag{5.4}$$

First note that the maximum degree of all propensity functions coincides with the highest order transition. Thus, to identify the drift term (5.3), the set of basis functions needs to contain at least monomials up to degree p. Second, as gEDMD identifies $a = \sigma\sigma^\top$ and not the diffusion term (5.4) itself, we compute for $c = x/N$

$$a(c) \coloneqq \sigma(c)\,\sigma(c)^\top = \sum_{k=1}^{K} \frac{1}{N}\,\widetilde{\alpha}_k(c)\,\nu_k\nu_k^\top,$$

which implies that monomials are sufficient for the identification of the diffusion term as well. Moreover, due to diffusion identification using Equation (3.3), we can conclude that also monomials of degree $p+1$ are required. However, in Section 5.2.4 we see that monomials are not always sufficient to correctly identify the underlying dynamics.

Identification. We can now compute the matrices Ψ_X and $\mathrm{d}\Psi_X$ of Section 3.1.2 and set up the minimization problem $\|\mathrm{d}\Psi_X - M\Psi_X\|_F$. The solution $L = M^\top$ represents an empirical estimate of the infinitesimal generator $\mathcal{L}$ corresponding to the ABM. We identify the drift and diffusion terms b and σ using a suitable projection matrix B, which are now *global* descriptions (i.e., functions depending on x), cf. Section 3.1.1. The considerations above are not restricted to ABMs but applicable to other (stochastic) dynamical systems, where different scale regimes, estimation of drift and diffusion terms, or conversation play a role. Algorithm 5.1 summarizes the procedure.

Remark 5.1.1. A Python toolbox for the data-driven model reduction of agent-based systems via Algorithm 5.1 is available at `https://github.com/Henningston/ABMs`. The toolbox contains the following:

(i) The voter model implemented as Markov jump process on complete networks, the extended voter model implemented in discrete-time for arbitrary networks, and the predator-prey model. The respective models are contained in `VoterModel.py`, `ExtendedVoterModel.py` and `PredatorPreyModel.py`.

(ii) The demo scripts use the voter model to show how the toolbox can be used. The script `demo_data_generation.py` illustrates the generation of measurements of the aggregated state. In `demo_post_processing.py` drift and diffusion terms for each measurement are estimated pointwise using the Kramers–Moyal formulae. The data is normalized and the generator approximation matrix is computed. The reduced stochastic differential equation can be simulated with `demo_reduced_SDE.py`. Its accuracy can be evaluated with `demo_evaluation.py`.

Algorithm 5.1: Algorithm 4.1 in [126]

1. Generate m measurements $\{x_l\}_{l=1}^m$ of the aggregated state of the ABM.
2. Estimate the drift and diffusion terms $\{b(x_l)\}_{l=1}^m$ and $\{a(x_l)\}_{l=1}^m$ at the measurement points, e.g., via Monte Carlo simulations for short lag times τ using the Kramers–Moyal formulae (5.1) and (5.2).
3. If applicable, normalize the data:
   a) Reduce the training data by keeping only $d-1$ components of each measurement as well as its drift and diffusion estimates.
   b) Scale by the number of agents N, i.e., $c_i(t) = x_i(t)/N$.
4. Choose a suitable set of basis functions $\{\psi_i\}_{i=1}^n$ and compute the matrices Ψ_X and $\mathrm{d}\Psi_X$.
5. Minimize $\|\mathrm{d}\Psi_X - M\Psi_X\|_F$ and obtain a generator approximation $L = M^\top$ and identify the drift and diffusion terms using Formula (3.3).

5.2 Numerical Results

We now consider four benchmark problems to evaluate the efficacy of the procedure summarized in Algorithm 5.1. In Section 5.2.1, for varying numbers of agents N and numbers of Monte Carlo samples k for the pointwise drift and diffusion estimates we compare the data-driven results with the theoretical SDE limit model (2.16) for the voter model. If the number of agents is sufficiently large and the accuracy of the estimates is high enough, the data-driven SDE coincides with the SDE limit model associated with the voter model, which shows that these are two crucial parameters for the quality of the numerically obtained model. In Section 5.2.2, we extend the SDE limit model and consider the case where the network consists of clusters connected by a few edges only. We observe that with increasing connectivity between the clusters the fractions of agents sharing the same opinions synchronize. In the subsequent Sections 5.2.3 and 5.2.4, we go beyond network-based descriptions and consider the predator-prey model and the civil violence model. While the first model still allows the use of monomials as basis functions, this is not the case for the second. All results are compared using the root mean square error defined in (3.5).

5.2.1 Extended Voter Model on Complete Networks

We consider the voter model with $d = 3$ opinions as defined in Section 2.2.3 and assume a fully connected interaction network. Due to the conservation of agents, this is essentially a two-dimensional system and the state space is given by the two-dimensional simplex $\mathbb{X}_N$ as defined in (2.11). Thus, we reduce the limit SDE (2.16) by one equation to make it comparable with the data-driven SDE. We set the same rate constants as in

Example 2.2.2, i.e.,

$$\gamma_{12} = \gamma_{23} = \gamma_{31} = 2, \tag{5.5a}$$
$$\gamma_{32} = \gamma_{21} = \gamma_{13} = 1, \tag{5.5b}$$
$$\gamma'_{ij} = 0.01, \tag{5.5c}$$

for $(i,j) = \{(1,2),(2,3),(3,1),(3,2),(2,1),(1,3)\}$. We then scale each measurement by the number of agents N and obtain $c_j(t) = 1 - \sum_{i \neq j} c_i(t)$. Then, the drift and diffusion terms are given by $b\colon \mathbb{X} \to \mathbb{R}^2$ and $a\colon \mathbb{X} \to \mathbb{R}^{2\times 2}$, respectively. Their derivation can be found in Appendix A.3. Note that $a(c) = a(c)^\top = (a_{ij}(c))$.

For the dictionary we choose monomials up to degree 3 as the highest order transition, i.e., imitation, is of order 2. We can now analytically construct for any given number of agents $N \in \mathbb{N}$ the first columns of the generator approximation L_N using the coefficients of b and a. For example, for $N = 10$ agents the matrix entry l_{22} is given by the coefficient of c_1 in the first component b_1 of the drift term b, i.e., $l_{22} = \gamma_{31} - \gamma_{13} - \gamma'_{12} - \gamma'_{13} - \gamma'_{31}$, see Equation (A.2a) in Appendix A.3 for details. Then, the first six columns of matrix L_{10} are given by

$$L_{10} = \begin{array}{c} \\ 1 \\ c_1 \\ c_2 \\ c_1^2 \\ c_1c_2 \\ c_2^2 \\ c_1^3 \\ c_1^2c_2 \\ c_1c_2^2 \\ c_2^3 \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & c_1 & c_2 & c_1^2 & c_1c_2 & c_2^2 & \dots \end{array} \\ \left[\begin{array}{cccccccc} 0 & 0.01 & 0.01 & 0.001 & 0 & 0.001 & \dots \\ 0 & 0.97 & 0 & 0.321 & 0.009 & 0 & \dots \\ 0 & 0 & -1.03 & 0 & 0.009 & 0.321 & \dots \\ 0 & -1 & 0 & 1.64 & 0 & 0 & \dots \\ 0 & -2 & 2 & 0 & -0.36 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & -2.36 & \dots \\ 0 & 0 & 0 & -2 & 0 & 0 & \dots \\ 0 & 0 & 0 & -4 & 1 & 0 & \dots \\ 0 & 0 & 0 & 0 & -1 & 4 & \dots \\ 0 & 0 & 0 & 0 & 0 & 2 & \dots \end{array}\right] \end{array} \in \mathbb{R}^{10\times 10}.$$

Given the expressions derived for the drift and diffusion term of the SDE limit model, see Appendix A.3, we expect that the drift and diffusion terms of the data-driven system take on the following structure:

$$b_i(c) \coloneqq \beta_5^i c_1^2 + \beta_4^i c_2^2 + \beta_3^i c_1c_2 + \beta_2^i c_1 + \beta_1^i c_2 + \beta_0^i, \tag{5.6a}$$
$$a_{ij}(c) \coloneqq \kappa_5^{ij} c_1^2 + \kappa_4^{ij} c_2^2 + \kappa_3^{ij} c_1c_2 + \kappa_2^{ij} c_1 + \kappa_1^{ij} c_2 + \kappa_0^{ij}. \tag{5.6b}$$

We can immediately obtain the coefficients β_h^i appearing in (5.6a) from the second and third column of matrix L_N, i.e,

$$b_1(c) = (\mathcal{L}\psi_2)(c) = -c_1^2 - 2c_1c_2 + 0.97c_1 + 0.01,$$
$$b_2(c) = (\mathcal{L}\psi_3)(c) = c_2^2 + 2c_1c_2 - 1.03c_2 + 0.01.$$

The coefficients κ_h^{ij} appearing in (5.6b) are obtained from the columns four to six. For instance, the component $a_{12}(c)$ is given by

$$a_{12}(c) = (\mathcal{L}\psi_5)(c) - b_1(c)\,c_2 - b_2(c)\,c_1 = -0.3\,c_1c_2 - 0.001\,c_1 - 0.001\,c_2.$$

Under certain conditions we can recover the rate constants of the underlying Markov jump process. For the considered system we compare the drift and diffusion coefficients of the SDE limit model (2.16) with the data-driven system (5.6) and set up a system of linear equations $A\gamma = v$, where A is a suitable matrix and γ and v are given by

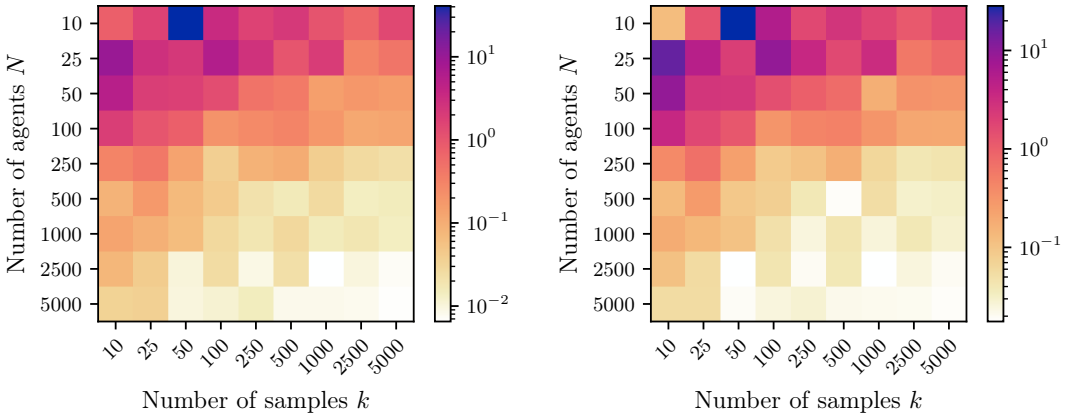
$$\gamma = [\gamma_{12}, \gamma_{13}, \dots, \gamma_{32}]^\top,$$
$$v = [\beta_5^1, \dots, \beta_0^2, \kappa_5^{11}, \dots, \kappa_0^{22}]^\top.$$

In this example the rate constants (5.5) are chosen such that the model is symmetric in the sense that imitation is possible in both ways (i.e., $\gamma_{ij} \neq 0$ for all $i \neq j$). As a consequence, we cannot solve the linear system exactly but only find dependencies for γ_{ij} and γ_{ji} that give the same coefficients of the drift and diffusion terms b and σ. However, this is only affecting the reconstruction of the underlying Markov jump process but not the data-driven model.

Evaluations

We compare the numerical results to the corresponding analytical counterparts, i.e., the columns of matrix L_N and the drift and diffusion terms of the SDE limit model. We set a maximum of 5000 for both the number of agents N and the number of Monte Carlo samples k for the pointwise estimation via the Kramers–Moyal formulae. As the state space $\mathbb{X}_N$ is discrete and the number of agents N assumed to be constant, the amount of distinct points is finite and depends on N and d. More precisely, the number of distinct points of a d-dimensional regular discrete simplex with $N+1$ points on each edge is given by $\binom{N+d}{d}$ for $d \leq N$ [33]. In this example, we consider three opinions and thus have a two-dimensional simplex with $\binom{N+2}{2}$ points. Choosing $m \leq \binom{N+d}{d}$ uniformly distributed measurements in $\mathbb{X}_N$ for different N as indicated in Table 5.1, we estimate the drift and diffusion term for each point via finite differences using the Kramers–Moyal formulae for k short trajectories of the Markov jump process with a lag time of $\tau = 0.01$. In total we used $m \cdot k$ training data points for the identification.

Figure 5.2 shows the identification error between the numerically obtained drift and diffusion coefficients and the analytical counterparts of the SDE limit model depending on the number of agents and Monte Carlo samples. With increasing N and k the error decreases by several orders of magnitude. However, the number of agents has a significantly larger influence than the number of samples. In particular, for small N, e.g., $N = 10$, we observe that higher values of k do not improve the error. This is consistent with Equation (2.17) since the SDE (2.16) approximation of the Markov jump process improves for larger N.



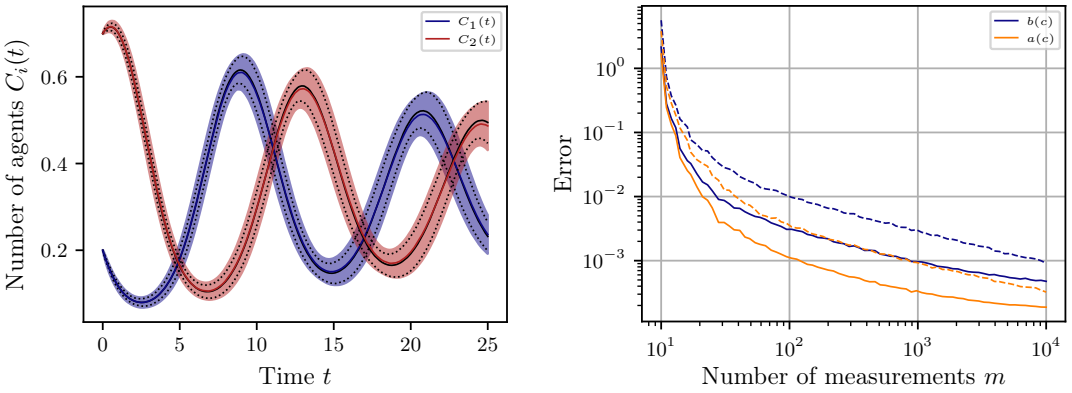
(a) **(b)**

Figure 5.2: Approximation error of the coefficients of (a) the drift and (b) diffusion estimates for the voter model in Section 2.2.3 compared to the exact SDE limit model (2.16) depending on the number of agents N and number of Monte Carlo samples k for the estimation via Kramers–Moyal formulae. For increasing N and k the approximation error decreases. The number of measurements m depends on N and is given in Table 5.1. Reproduced from [126].

Table 5.1: Measurement set sizes used for the computation of the system identification error of the voter model on complete networks for given N. The total training data set is of size $m \cdot k$.

Number of agents N	Measurements m
10	7
25	35
50	133
100	515
250	3163
$N \geq 500$	10000

Another important aspect next to system identification is how well the data-driven model approximates the dynamics of the ABM, e.g., to make predictions about the number of agents of a specific type, which then might be used for optimization or control. In Figure 5.3 (a) we see the long-term prediction of the data-driven model and the analytical SDE limit model (2.16). Expectation and standard deviation, which are estimated from 1000 Monte Carlo simulation, show a good agreement. In particular with respect to the numerical effort that makes the simulation required for Figure 5.3 (a) infeasible in many cases, the data-driven model provides valuable results. In addition,



(a) **(b)**

Figure 5.3: (a) Expectation (solid) and standard deviation (shaded) of the data-driven model for the dynamics of the voter model for $N = 5000$ agents and the corresponding SDE limit model (2.16) (black/dotted) estimated from 1000 Monte Carlo simulations. The initial state is set to $c(0) = [0.2, 0.7, 0.1]^\top \in \mathbb{X}$. (b) Approximation error of the drift and diffusion estimates compared to the analytical SDE limit model (2.16) depending on the number of measurements m for fixed $k_1 = 10$ (dashed), $k_2 = 100$ (solid). The error is averaged over 100 simulations and $N = 5000$ agents. Independently, for both parameters m and k, a smaller error can be expected for larger amounts of training data. Reproduced from [126].

the time required to simulate the data-driven model is significantly less than the original ABM.

Next, we compute the dependency of the error on the number of measurements m given a fixed k. Figure 5.3 (b) shows the error depending on m averaged over 100 simulations for 5000 agents for two fixed k, namely $k_1 = 10$ (dashed) and $k_2 = 100$ (solid). As expected, with increasing m the error decreases by several orders of magnitude independently of k. If we consider the total number of training data points $m \cdot k$, the impact of increasing m is larger than the one of increasing k. We observe that for small $m \cdot k$, the error is smaller for $k_1 = 10$ than for $k_2 = 100$ since the state space is covered more densely by the measurements. For instance, for $k_1 = 10$ and $m = 100$ the error is less than for $k_2 = 100$ and $m = 10$. This can easily be seen by shifting the solid lines by one decade to the right. Thus, we have two parameters to tune the amount of training data to be used.

5.2.2 Extended Voter Model on Clustered Networks

In the previous section we assumed that the network is fully connected. However, since the assumption that any agent can interact with all other agents at any time is rather the exception in ABMs, we consider now the case where the network consists of Q

(not necessarily equally-sized) clusters. Within a cluster we assume that each agent is connected to all other agents, i.e., each cluster q is a complete subgraph of size N_q. Between distinct clusters two agents are connected with probability p such that for sufficiently small p the clusters are connected only by a few edges. The corresponding adjacency matrix can then be represented as block matrix with dense and sparse submatrices. As before, the agents act according to the rules of the voter model. However, as the network is not fully connected anymore, transition propensities depend on the size of the individual neighborhood of each agent and therefore might differ among agents. Additionally, since now aggregation without respecting the network structure leads to errors, we augment the population state to include subpopulations for each cluster, i.e., we aggregate per cluster. This cluster-wise aggregated data is then used to learn a reduced, data-driven model of the agent dynamics.

An SDE Limit Model for Clustered Networks

We now extend the SDE limit model to describe the temporal evolution of the relative frequencies for each type per cluster. Assume that the interaction network consists of Q clusters and that the edges connecting two different clusters are drawn uniformly with probability p. The augmented system state is given by

$$C(t) = \left[c_1(t)^\top, \dots, c_Q(t)^\top\right]^\top \in \mathbb{R}^{dQ}.$$

Let $\widetilde{\alpha}_{q,k}$ denote the rescaled propensity function of transition k in cluster q and $\nu_{q,k} \in \mathbb{R}^{dQ}$ the corresponding net change vector. We obtain the augmented SDE limit model given by

$$\mathrm{d}C(t) = \sum_{q=1}^{Q} \left[\sum_{k=1}^{K_q} \widetilde{\alpha}_{q,k}(C(t))\, \nu_{q,k}\, \mathrm{d}t + \sum_{k=1}^{K_q} \frac{1}{\sqrt{N}} \sqrt{\widetilde{\alpha}_{q,k}(C(t))}\, \mathrm{d}W_{q,k}(t)\, \nu_{q,k} \right].$$

For simplicity, we assumed equally-sized clusters, i.e., N agents per cluster. However, clusters of different sizes are also possible as we see in the following example.

Example 5.2.1. Consider a network with two clusters Q_1 and Q_2, each with N_1 and N_2 agents. Let p denote the probability for an edge connecting two agents of cluster Q_1 and Q_2. The connection strength between cluster Q_1 and Q_2 is defined as the ratio between the number of edges E connecting both clusters and the total number of possible connecting edges $E_{\max} = N_1 N_2$. The expected connection strength is then given by

$$\mathbb{E}\left[\frac{E}{E_{\max}}\right] = \frac{\mathbb{E}[E]}{E_{\max}} = \frac{p N_1 N_2}{N_1 N_2} = p.$$

Let us consider the same transition rules as for the voter model in Section 2.2.3, i.e., imitation and exploration. Since the latter is a transition independent of all other agents, it is also independent of the network structure. In contrast, imitation depends on

the network as it is induced either inside or outside a cluster such that the corresponding transition rule is given by

$$R_{qq',ij}\colon S_{q,i} + S_{q',j} \to S_{q,j} + S_{q',j}.$$

If $q = q'$, we call $R_{qq',ij}$ an *intra-cluster* transitions, otherwise an *inter-cluster* transitions. Note that while the transition rules $R_{qq',ij}$ for $q = q'$ and R_{ij} (cf. Section 2.2.3) look the same, their propensity functions α_{ij} differ. For intra-cluster transitions they are given by

$$\alpha_{ij} = \frac{1}{N_q + pN_{q'}} \gamma_{q,ij}\, x_{q,i}\, x_{q,j}.$$

Note the difference in the denominator which is due to the fact that each agent expects $N_q + pN_{q'}$ possible partners for interaction. For inter-cluster transitions the propensity functions are given by

$$\alpha_{qq',ij} = \frac{p}{N_q + pN_{q'}} \beta_{q,ij}\, x_{q,i}\, x_{q',j}$$

with corresponding net change vector $\nu_{qq',ij} = \nu_{ij}$. This is because an inter-cluster transition $R_{qq',ij}$ has only influence cluster Q_q and thus, from the agents' point of view, $R_{qq',ij}$ has the same effect as R'_{ij}. For the augmented state $C(t) = \left[c_1(t)^\top, c_2(t)^\top\right]^\top \in \mathbb{R}^{2d}$, the SDE solution is given by

$$\begin{aligned}
\mathrm{d}c_{q,i}(t) = \Bigg[& \sum_{i\neq j} \frac{1}{(p+1)} (\gamma_{q,ji} - \gamma_{q,ij})\, c_{q,i}(t)\, c_{q,j}(t) && \text{(5.7a)}\\
& + \sum_{i\neq j} \left[\gamma'_{q,ji}\, c_{q,j}(t) - \gamma'_{q,ij}\, c_{q,i}(t)\right] && \text{(5.7b)}\\
& + \sum_{i\neq j} \frac{p}{(p+1)} \left[\beta_{q,ji}\, c_{q,j}(t)\, c_{q',i}(t) - \beta_{q,ij}\, c_{q,i}(t)\, c_{q',j}(t)\right] \Bigg] \mathrm{d}t && \text{(5.7c)}\\
+ \frac{1}{\sqrt{N}} \Bigg[& \sum_{i\neq j} \sqrt{\frac{1}{(p+1)} \gamma_{q,ji}\, c_{q,i}(t)\, c_{q,j}(t)}\, \mathrm{d}W^{\mathrm{im}}_{q,ji}(t) - \sqrt{\frac{1}{(p+1)} \gamma_{q,ij}\, c_{q,i}(t)\, c_{q,j}(t)}\, \mathrm{d}W^{\mathrm{im}}_{q,ij}(t) && \text{(5.7d)}\\
& + \sum_{i\neq j} \sqrt{\gamma'_{q,ji}\, c_{q,j}(t)}\, \mathrm{d}W^{\mathrm{ex}}_{q,ji}(t) - \sqrt{\gamma'_{q,ij}\, c_{q,i}(t)}\, \mathrm{d}W^{\mathrm{ex}}_{q,ij}(t) && \text{(5.7e)}\\
& + \sum_{i\neq j} \sqrt{\frac{p}{(p+1)} \beta_{q,ji}\, c_{q,j}(t)\, c_{q',i}(t)}\, \mathrm{d}W^{\mathrm{int}}_{q,ji}(t) - \sqrt{\frac{p}{(p+1)} \beta_{q,ij}\, c_{q,i}(t)\, c_{q',j}(t)}\, \mathrm{d}W^{\mathrm{int}}_{q,ij}(t) \Bigg]. && \text{(5.7f)}
\end{aligned}$$

The addends (5.7c) and (5.7f) correspond to inter-cluster transitions, while the others correspond to intra-cluster transitions. To simplify the notation, we assumed that both clusters are of the same size. We drop the index q whenever it is clear from the context. However, for unequally-sized clusters the following coefficients change:

$$\begin{aligned}
\tfrac{1}{p+1} \quad &\longrightarrow \quad \frac{1}{p\frac{N_{q'}}{N_q}+1}, && \text{(inter-cluster)}\\
\tfrac{p}{p+1} \quad &\longrightarrow \quad \frac{p}{p\frac{N_q}{N_{q'}}+1}, && \text{(intra-cluster).}
\end{aligned}$$

Additionally, $1/\sqrt{N}$ changes to $1/\sqrt{N_q}$. Then, for $N_q \gg N_{q'}$, the influence of cluster q' vanishes, while $N_q \ll N_{q'}$ the opposite is the case. △

Algorithm 5.2: Discrete-Time Extended Voter Model

forall *timesteps* **do**
  Randomly arrange all agents in a queue.
  forall *agents in queue* **do**
    Get number N of adjacent neighbors.
    Get number X_j of type S_j in neighborhood for all d types.
    Calculate transition probabilities $P = \exp(t_{\text{step}} G)$ based on neighbors for

$$(G_{ij})_{i,j=1,\dots,d} = \begin{cases} -\sum_{j=1}^{d} \frac{\gamma_{ij} X_j}{N} + \gamma'_{ij}, & \text{if} \quad i = j, \\ \frac{\gamma_{ij} X_j}{N} + \gamma'_{ij}, & \text{else.} \end{cases}$$

    Update agent's state according to previously calculated probabilities.

Evaluations

We consider a network consisting of two equally-sized clusters, each with $N = 50$ agents, and assume that the rate constants for intra- as well as inter-cluster transitions are equal in both clusters, i.e., $\gamma_{q,ij} = \gamma_{q',ij}$, $\gamma'_{q,ij} = \gamma'_{q',ij}$, and $\beta_{q,ij} = \beta_{q',ij} = \gamma_{q,ij}$ for all $(i,j) = \{(1,2),(2,3),(3,1),(3,2),(2,1),(1,3)\}$. For imitation we set the rate constants to (5.5a) and (5.5b). For exploration we set $\gamma'_{ij} = 0$ for all pairs (i,j). In this section we simulated the ABM in discrete time with step size $t_{\text{step}} = 0.01$ using Algorithm 5.2. We chose $m = 1000$ uniformly distributed initial states of the ABM and created $k = 1000$ samples for each of them to obtain the data-driven model via Algorithm 5.1 applied to the cluster-based aggregate states of the agent dynamics on the network.

The data-driven model is compared against the model defined in (5.7) for the two networks depicted in Figures 5.4 (a) and (b). The first network, being a subgraph of the second, has a connection strength of $p = 0.01$. The second network has a 20-times larger connectivity, i.e., $p = 0.2$. The prediction of the temporal evolution of the expectation for each type per cluster is shown in Figures 5.4 (c) and (d). Although both predictions start from the same initial value, they quickly differ in their evolution over time. We also find that at higher connectivity, i.e., larger p, both clusters synchronize so that the relative number of agents per type in each cluster is identical, see Figure 5.4 (d). In both cases, this is a direct consequence of the network structure. As for complete networks in Section 5.2.1, the results improve for larger values of N, m, and k.

Remark 5.2.2. Consider a network where two agents are randomly connected with a probability of 10 %. If the network is not connected, we consider each component individually. In any case, the resulting network is sparsely connected. We assume that the network (the component) consists of $N = 500$ agents such that the approximate average degree is 50. We compute a data-driven model for this setting using $m = k = 1000$ measurements and realizations for a lag time of $\tau = 0.01$ and compared it to the ABM. Figure 5.5 (a) shows the first moments of the data-driven model (solid)

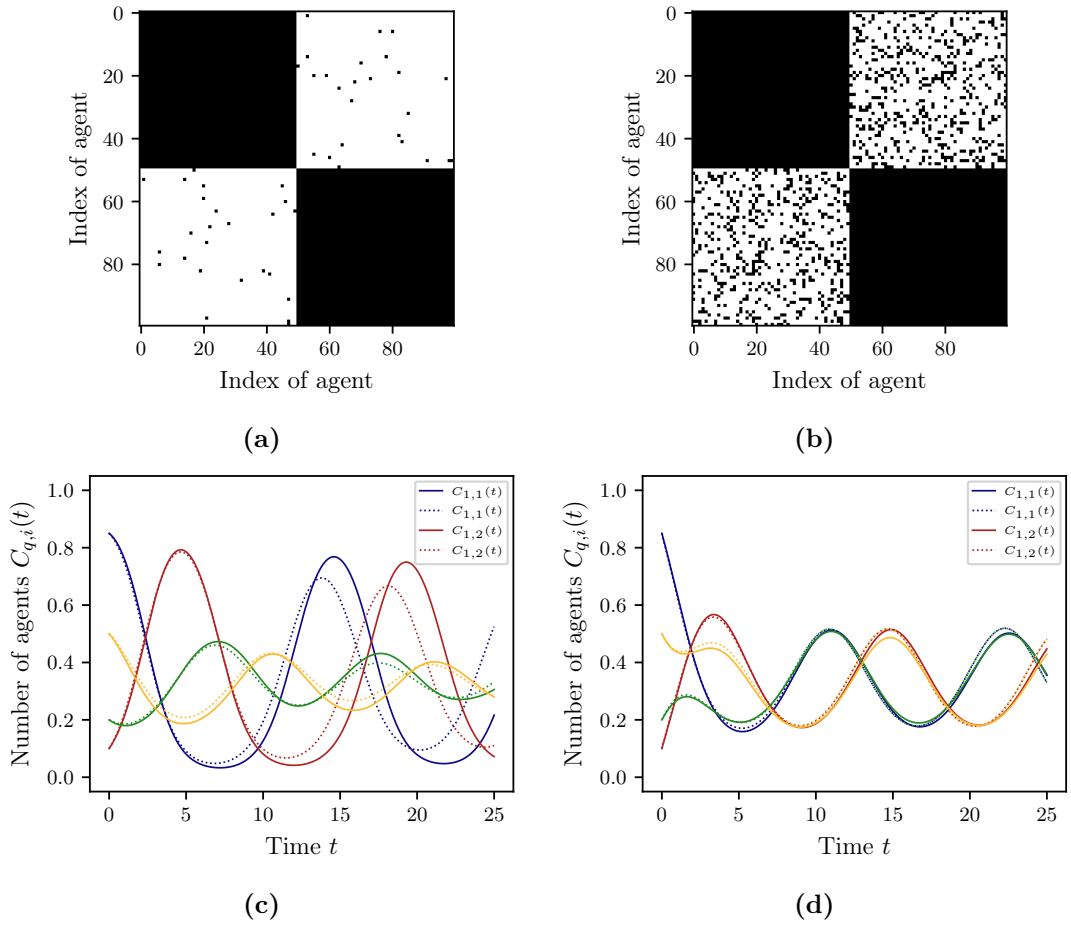
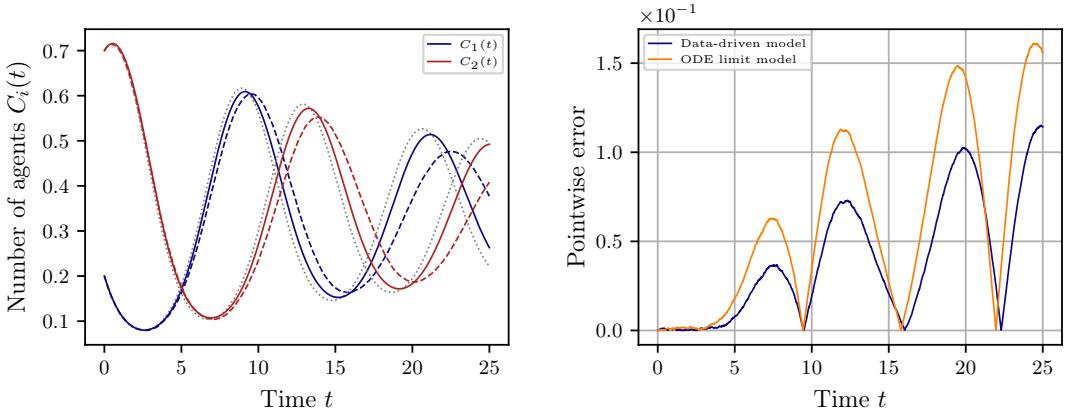


Figure 5.4: (a-b) Adjacency matrices of interaction networks with two clusters of connectivity $p = 0.01$ and $p = 0.2$, respectively. Each cluster has $N = 50$ agents, where black represents an edge and white no edge. (c-d) First-order moments of the data-driven model (solid) and the limit SDE (5.7) (dotted) for initial value $c(0) = [0.85, 0.1, 0.05, 0.2, 0.5, 0.3]^\top$. The data-driven model is estimated using $m = k = 1000$ measurements and realizations for a lag time of $\tau = 0.01$. Reproduced from [126].

and the ABM (dashed) estimated from 1000 Monte Carlo simulations. While for short times t the data-driven model agrees with the ABM, for larger time t the prediction deteriorates primarily due to the sparsity of the network. It should be noted that the lack of a reference model for sparse networks complicates the analysis. However, if we compare the pointwise difference between the expectations of the data-driven approximation (solid) or the SDE limit model (2.16) (gray/dotted) and the voter model (dashed), the data-driven model provides a better approximation than the limit SDE, see Figure 5.5 (b).

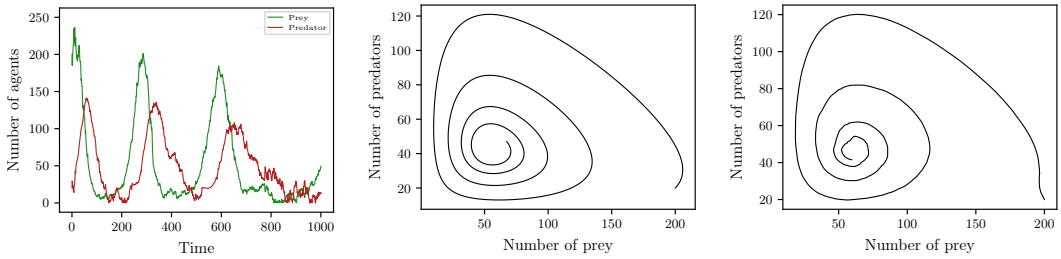


(a) **(b)**

Figure 5.5: (a) Expectation of the data-driven model (solid) compared to the voter model (dashed) for $N = 500$ agents on a random network with average degree of approximately 50 estimated from 1000 Monte Carlo simulations. The initial state is $c(0) = [0.2, 0.7, 0.1]^\top \in \mathbb{X}$. The data-driven model is estimated using $m = k = 1000$ measurements and realizations for a lag time of $\tau = 0.01$. For comparison, the expectation of the SDE limit model (2.16) is indicated in gray/dotted. (b) Pointwise error with respect to S_1 agents between the data-driven model or the expectation of the SDE limit model and the voter model. The error of the data-driven model is lower, indicating a better approximation. Figure (a) reprinted from [126].

5.2.3 Predator-Prey Model

Let us now move beyond network-based agent-based system and consider the predator-prey model introduced in Section 2.2.4 for the parameters listed in Table 2.1. Although all agents move relatively slowly with respect to the dimension of the space and search radius v of the predators (cf. Remark 2.2.4), we assume that the model we are looking for still resembles the classical Lotka–Volterra differential equations. Thus, we choose monomials up to degree 3 for the set of basis functions so that we can identify the coefficients of the drift and diffusion terms. The data-driven model is then learned from $m = k = 1000$ measurements and samples. As this ABM is formulated in discrete time, the lag time for the pointwise drift and diffusion estimates via (5.1) and (5.2), respectively, is set to one time step, i.e., $\tau = 1$. Note that step 3a) of Algorithm 5.1, i.e., reduction of the data, is not applicable here since the number of agents is not constant. Figure 5.6 (a) shows one possible trajectory of the data-driven SDE similar to that of the predator-prey model in Figure 2.3 (b). For a better comparison we computed the first-order moment of both models via 958 Monte Carlo simulations. In 42 out of 1000 realizations size of the prey population grew exponentially due to the extinction of the predators. Figure 5.6 (b) and (c) show the phase portraits of the first-order moment of



(a) **(b)** **(c)**

Figure 5.6: (a) Realization of the data-driven SDE learned from $m = k = 1000$ measurements and samples for the predator-prey model with parameters given in Table 2.1. Phase portraits of the first-order moment of (b) the data-driven SDE and (c) the predator-prey model estimated from 958 Monte Carlo simulations. Reprinted from [126].

the data-driven SDE and the predator-prey model. We observe that the reduced model is able to approximate the qualitative dynamical behavior of the ABM.

5.2.4 Civil Violence Model

We consider now the civil violence model introduced in Section 2.2.5 for the parameters listed in Table 2.2. This model is already so complex that a proper choice of basic functions is not obvious. For instance, a dictionary consisting exclusively of monomials is not sufficient in this case as we can see in Figure 5.7 (a), where the data-driven model is obtained using monomials up to order 4 and $m = k = 100$ measurements and samples for lag time $\tau = 1$. However, exploiting physical insights about the agent-based system, we are able to learn a reduced model that qualitatively agrees with the dynamical behavior of the full-complexity ABM.

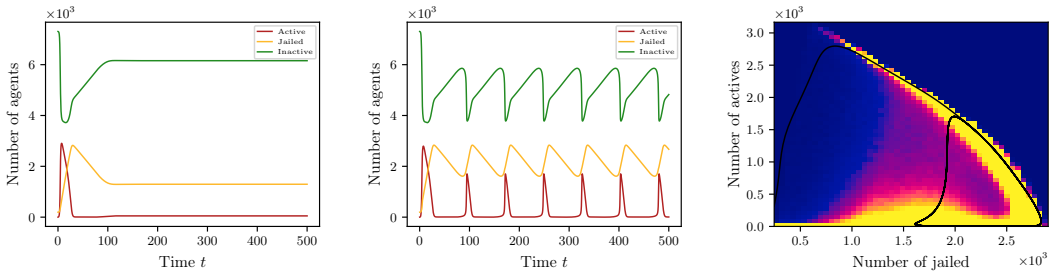
Let c_1 and c_2 denote the fraction of active and jailed agents, respectively. By conservation, the fraction of inactive agents is given by $c_3 = 1 - c_1 - c_2$. As shown in [51], the choice of arrest probability function plays a major role in the behavior of the model. Therefore, we augment the set of basis function $\{\psi_i\}_{i=1}^n$ by n additional functions defined by

$$\psi_{2i} \coloneqq \psi_i \cdot \widehat{P}, \quad i = 1, \dots, n, \tag{5.8}$$

where the function $\widehat{P}\colon \mathbb{R}^d \to [0, 1]$ is given by

$$\widehat{P}(c) \coloneqq \exp(-\kappa c_1) \sum_{i=0}^{15} \frac{(\kappa c_1)^i}{i!}.$$

The constant κ is the same as for the arrest probability function (2.20). Note that $\widehat{P}$ is related to the arrest probability function (2.20) via $P_{\text{arr}}(A_v, C_v) = 1 - \widehat{P}(C_v/A_v)$. The new set of basis function comprises $2n$ functions.



(a) **(b)** **(c)**

Figure 5.7: (a) Prediction of the temporal evolution of the data-driven model learned from the dynamics of the civil violence model with the parameters in Table 2.2 using $m = k = 100$ training points and samples and monomials as basis functions. (b) The same prediction, but using the augmented set of basis functions. (c) Empirically determined distribution of the civil violence model estimated from 10000 trajectories of length $t = 500$ and phase portrait of data-driven model. Blue corresponds to small and yellow to large values.

Table 5.2: Mean characteristics of the civil violence model and the data-driven model estimated from 10000 trajectories with length $t = 500$.

	Civil violence model	Data-driven model
Peak to peak distance	78.46	77.10
Peak height	1927	1704
Peak width at 0.98 relative height	20.52	21.12
Peak width at 0.5 relative height	9.409	6.767

Evaluations

Again, choosing monomials up to degree 4 and the functions defined in (5.8) as dictionary, we learn a data-driven model for the same set of training data. Figure 5.7 (b) shows a long-term prediction of the data-driven model, which highly resembles the trajectory shown in Figure 2.5 (b). The phase portrait shown in Figure 5.7 (c) fits the empirical obtained distribution of the ABM, which is estimated from 10000 Monte Carlo simulations with length $t = 500$. Furthermore, comparing the characteristic periodic outbursts of violence, we observe a remarkable agreement of the mean peak to peak distance and the mean peak width at 0.98 relative height between the data-driven model and the full ABM. Further characteristics are listed in Table 5.2. Thus, by using insider information about the agent-based system, we are able to augment set of basis functions such that it is possible to obtain a data-driven reduced model that qualitatively agrees with the agent dynamics.

6 Multi-Objective Optimization of Agent-Based Systems

In this chapter, we show how the reduced models computed in Chapter 5 can be used as surrogates to solve multi-objective optimization problems involving agent-based systems. Our goal is to find optimal tradeoffs between several possibly conflicting objectives, which are given by the dynamical behavior of the system. Surrogate modeling is of particular interest here, as objective functions defined by agent-based systems are often expensive or even infeasible to evaluate. We show that if the underlying dynamics are linear with respect to the parameters, it is possible to construct surrogate models with varying parameters using convex interpolation of Koopman generators. As an example we examine the long-term effect of a parameter change on the agent dynamics for two agent-based systems. In the case of the voter model, the goal is, for instance, to find or evaluate a campaign strategy to shift the opinion of the majority in a desired direction. We show that the surrogate model based on convex interpolation of Koopman generators has high agreement with an analytic derived surrogate for the expensive objective function under consideration. In the second example, the civil violence model, we consider the optimization of utilization of labor force. Both the number of officers deployed and the number of insurgents are to be minimized at the same time. In both scenarios, we show that the results obtained with the help of these surrogates actually approximate optimal points of the agent dynamics by comparing them to test points for which the objective is evaluated using the full-complexity agent-based systems.

6.1 Introduction to Multi-Objective Optimization

Whether in industry, economy, society, or even everyday life, a common problem is achieving multiple goals simultaneously. Often these contradict each other and share no optimum. For many industrial goods, for example, quality cannot be improved if the selling price is to be reduced at the same time. Another example with current reference concerns the length of curfews due to the corona virus pandemic. The minimization of socio-economic costs and limiting virus spread are two conflicting goals. Therefore, a compromise must be found, i.e., an agreement reached by mutual concessions, often varying or blending the original goals. An optimal compromise cannot be improved further without worsening at least one of the other goals.

Multi-objective optimization concerns the simultaneous optimization of k objective functions $f_1, \dots, f_k \colon \mathbb{R}^n \to \mathbb{R}$. Roughly speaking, multi-objective optimization first tries to find all the optimal compromises before a decision is taken. This opens up more

options for decision makers, as better compromises can easily be overlooked in single-objective optimization, e.g., due to an a priori prioritization of conflicting objectives. The question of what is "the best" optimal compromise is answered in decision-making theory and beyond the scope of this chapter.

In the following, we consider multi-objective optimization problems formulated as

$$\begin{aligned} \underset{y\in\mathbb{R}^n}{\text{minimize}} \quad & F(y) = [f_1(y), \dots, f_k(y)]^\top \\ \text{subject to} \quad & g_i(y) \leq 0, \quad i = 1, \dots, q \\ & h_j(y) = 0, \quad j = 1, \dots, p, \end{aligned} \tag{6.1}$$

where $F\colon \mathbb{R}^n \to \mathbb{R}^k$, $g\colon \mathbb{R}^n \to \mathbb{R}^q$ and $h\colon \mathbb{R}^n \to \mathbb{R}^p$. The *decision space* is given by $\mathbb{R}^n$ and the *objective space* by the image of F. The *feasible decision space* or *feasible set* is given by

$$\mathcal{R} = \{y \in \mathbb{R}^n \mid g(y) \leq 0, h(y) = 0\}.$$

The following definition provides the necessary fundamentals.

Definition 6.1.1 ([155, 156])**.** Consider the multi-objective optimization problem (6.1). Let $v, w \in \mathbb{R}^n$ and $y, y^\star \in \mathcal{R}$. Then

(i) v is *less than* w, i.e., $v <_p w$, if $v_i < w_i$ for all $i = 1, \dots, n$. The relation $\leq_p$ is defined in an analogous way.

(ii) v *is dominated* by w (equivalently, we say w *dominates* v) if

$$F(v) \neq F(w) \quad \text{and} \quad F(w) \leq_p F(v).$$

(iii) $y^\star$ is called *Pareto optimal* or *Pareto point* if there is no y that dominates $y^\star$.

The set of all Pareto optimal points is called *Pareto set* after the Italian engineer, economist and sociologist of the same name. The image of the Pareto set under the objective function F is called the *Pareto front*.

There exists various methods and techniques to solve multi-objective optimization problems such as scalarization, ε-constraint methods, evolutionary and genetic algorithms, particle swarm, agent-based and multi-agent methods, see [118, 155, 30, 133, 2, 14] and references therein. All these methods are well-suited to compute (sets of) single Pareto points. Other methods such as continuation methods [155] use the fact that the Pareto set is a $k-1$-dimensional smooth manifold under certain conditions. Set-oriented methods such as subdivision techniques [37] compute outer approximations of the entire Pareto sets. Many approaches use box discretizations to cover the Pareto set. A suitable data structure helps to limit memory requirements. For a general overview on set-oriented methods see, e.g., [156].

In this chapter, we use a subdivision technique known as *sampling algorithm* [37] to obtain a covering of the Pareto set. This algorithm uses only evaluations of the objective function F and does not require its derivative, which is especially advantageous for

Algorithm 6.1: [37, 132]

Let $\mathcal{B}^0$ be a collection of finitely many subsets of $\mathcal{R}$ such that $\bigcup_{B\in\mathcal{B}^0} B = \mathcal{R}$. Then, obtain the new collection $\mathcal{B}^{s+1}$, $s \geq 0$, iteratively from $\mathcal{B}^s$ in two steps:

1. Construct a new collection $\hat{\mathcal{B}}^{s+1}$ from $\mathcal{B}^s$ by subdividing each subset $B \in \mathcal{B}^s$ such that

$$\bigcup_{B\in\hat{\mathcal{B}}^{s+1}} B = \bigcup_{B\in\mathcal{B}^s} B,$$
$$\operatorname{diam}(\hat{\mathcal{B}}^{s+1}) = \theta^{s+1}\operatorname{diam}(\mathcal{B}^s)$$

   for $0 < \theta_{\min} \leq \theta^{s+1} \leq \theta_{\max} < 1$.

2. Define the new collection $\mathcal{B}^{s+1}$ by

$$\mathcal{B}^{s+1} := \left\{ B \in \hat{\mathcal{B}}^{s+1} \mid \nexists\, \hat{B} \in \hat{\mathcal{B}}^{s+1} \text{ such that } \hat{B} \text{ dominates } B \right\}.$$

agent-based systems. Assume that every decision variable y is bounded component-wise, i.e., $a_i \leq y_i \leq b_i$ for $i = 1, \ldots, n$, such that the feasible decision space $\mathcal{R}$ is given by

$$\mathcal{R} = [a_1, b_1] \times \ldots \times [a_n, b_n] \subset \mathbb{R}^n. \tag{6.2}$$

Let $\mathcal{B}$ denote a collection of finitely many subsets of $\mathcal{R}$ such that $\bigcup_{B\in\mathcal{B}} B = \mathcal{R}$. Before summarizing the sampling algorithm given in [37], we need the definition of set-wise dominance.

Definition 6.1.2 (Remark 2.1.6 in [132])**.** A set $B^\star$ *dominates* a set B if for every $y \in B$ there exists at least one $y^\star \in B^\star$ such that $y^\star$ dominates y.

The sampling algorithm works iteratively in two steps. In the first step, each box $B \in \mathcal{B}^s$, $s \geq 0$, is subdivided with respect to the coordinates, which leads to a new collection of boxes $\hat{\mathcal{B}}^{s+1}$. In the second step, each box $B \in \hat{\mathcal{B}}^{s+1}$ is subject to a set-wise non-dominance test in which all boxes containing only dominated points are discarded. The remaining boxes form the new collection $\mathcal{B}^{s+1}$. Algorithm 6.1 summarizes the procedure following the notation in [132]. For further details see [37, 155, 156, 132].

Remark 6.1.3. The following points are worth mentioning:

(i) Set-wise dominance as in Definition 6.1.2 is difficult to check. Using a finite number of test points per box, we can perform the non-dominance test heuristically. An insufficient testing, however, might result in boxes being falsely discarded. By adding all neighboring boxes of the current collection and then discarding the dominated ones, falsely rejected boxes can be recovered. See [156, 132] for further details.

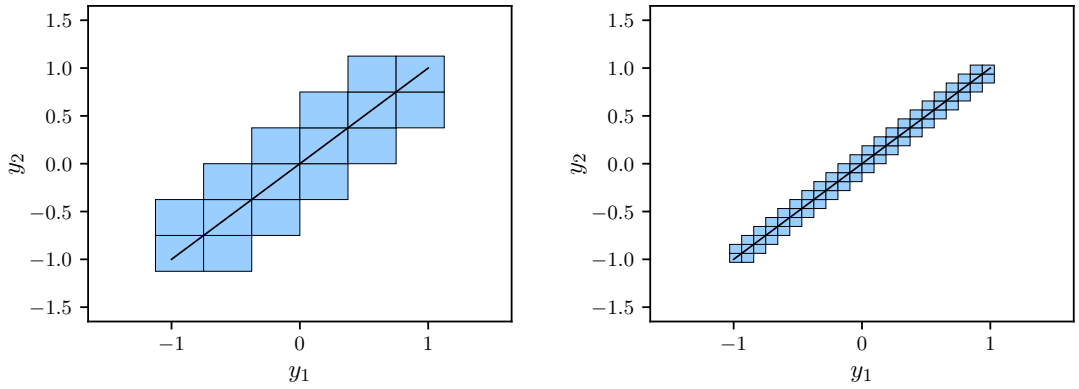


Figure 6.1: Box coverings after (a) 8 and (b) 12 iterations obtained using Algorithm 6.1. The black/solid line indicates the Pareto set.

(ii) As each box can be represented by a center and a radius, all collections can be stored efficiently in a binary tree such that the memory consumption grows linearly with n. GAIO [36], a toolbox written for MATLAB, provides binary tree data structures and algorithms for set-oriented calculations and can be used for the implementation of Algorithm 6.1.

The following example illustrates the previous concepts and demonstrates the functionality of Algorithm 6.1.

Example 6.1.4. Consider a multi-objective optimization problem of the form (6.1) with objective functions $f_i \colon \mathcal{R} \to \mathbb{R}$ given by

$$f_1(y) = (y_1 - 1)^2 + (y_2 - 1)^2$$
$$f_2(y) = (y_1 + 1)^2 + (y_2 + 1)^2$$

and feasible decision space $\mathcal{R} = [-1.5, 1.5] \times [-1.5, 1.5]$. Obviously, the Pareto set is given by $\{y \in \mathcal{R} \mid -1 \leq y_1 = y_2 \leq 1\}$, i.e., a straight line connecting the minima $[1, 1]^\top$ and $[-1, -1]^\top$ of the two objective functions f_1 and f_2, respectively. Figure 6.1 shows the box coverings obtained via Algorithm 6.1 after 8 and 12 iterations. The black/solid line indicates the Pareto set. △

6.2 A Multi-Objective Optimization Ansatz for Agent-Based Systems

Agent-based systems often have a lot of different parameters influencing the dynamical behavior of the system. While some of them are just of technical nature (i.e., necessary

for the correct implementation), most of them can be given an interpretation (e.g., reproduction rates, radius of vision or range of motion). In this section, we want to take influence on these parameters with respect to some target behavior of the ABM. The question behind changing the parameters is which policies, strategy, action, sanction, program, operations or any other measure achieves best all desired targets. For the ease of readability we use the terms *policy* and *influence* throughout this section. Clearly, none of these are free of charge and thus, we assume that influencing the ABM parameters causes certain costs. The aim is to find the optimal tradeoffs between all desired goals and the required investment for the policy. In the sense of multi-objective optimization, for an objective function $F\colon \mathcal{R} \to \mathbb{R}^k$ given (partially) by an agent-based system, we want to compute the Pareto set for n decision variables (here parameters of an agent-based system). Assume that the objective functions $f_i\colon \mathcal{R} \to \mathbb{R}$, $i = 1, \dots, k$, can be written as

$$f_i(y) = \int_{t_0}^{t_1} r_i(y)\mathrm{d}t + s_i(y)$$

where $r_i\colon \mathcal{R} \to \mathbb{R}$ and $s_i\colon \mathcal{R} \to \mathbb{R}$ denote running and static costs, respectively. There are two issues:

(A) Most agent-based systems are high-dimensional, which makes them computationally expensive. Additionally, the ABM can be stochastic. Consequently, both have an effect on the objective functions f_i. Therefore, for an efficient computation of the multi-objective optimization problem either surrogates of the objective functions f_i are needed or a surrogate model for the entire ABM that is less expensive to evaluate.

(B) Most agent-based system cannot be expressed analytically such that (some) exact derivatives of the objectives f_i cannot be found. If they can be approximated, they might not be accurate enough. We hence need derivative-free or fault-tolerant methods to solve the multi-objective optimization problem.

The major of these two problems is (A) since we can circumvent (B) simply by using Algorithm 6.1, which only relies on function evaluations and does not require derivatives of the objective functions. To address the first problem, we consider surrogate models of entire agent-based systems based on Koopman generators.

6.2.1 Koopman Generator Interpolation

In Chapter 5, we have shown that under certain conditions it is possible to obtain reduced models using the Koopman generator that accurately approximate the aggregate dynamics of ABMs. However, these are fixed for the chosen set of parameters and computing a reduced model for every parameter during the optimization process would not only be expensive but also ineffective. It has been shown recently that it is possible to interpolate between Koopman generators of control-affine dynamical systems, i.e., systems which are linear with respect to the control, see Theorem 3.2 in [136]. We can deduce the following for parameter-affine non-deterministic dynamical systems:

Theorem 6.2.1. *Given a space of twice differentiable functions, if the dynamics are parameter-affine (i.e., linear with respect to the parameter), then the Koopman generators are parameter-affine.*

Proof. Consider a parameter-affine dynamical system

$$\mathrm{d}X_t = b(X_t, p)\,\mathrm{d}t + \sigma(X_t, p)\,\mathrm{d}W_t \tag{6.3}$$

with drift $b\colon \mathbb{R}^d \to \mathbb{R}^d$, diffusion $\sigma\colon \mathbb{R}^d \to \mathbb{R}^{d\times s}$ and parameter $p\colon \mathbb{R}^{d_p} \to \mathbb{R}^{d_p}$. Drift and diffusion terms of system (6.3) are give by

$$b(X_t, p) = b_0(X_t) + \sum_{i=1}^{d_p} p_i\, b_i(X_t) \tag{6.4}$$

$$a(X_t, p) = a_0(X_t) + \sum_{i=1}^{d_p} p_i\, a_i(X_t) \tag{6.5}$$

for $a = \sigma\sigma^\top$. Given some twice differentiable function f, the stochastic Koopman generator $\mathcal{L}_p$ depending on p applied to f yields

$$\begin{aligned}\mathcal{L}_p f &= b \cdot \nabla_x f + \frac{1}{2}\, a : \nabla_x^2 f \\ &= b_0 \cdot \nabla_x f + \sum_{i=1}^{d_p} p_i\, b_i \cdot \nabla_x f + \frac{1}{2}\, a_0 : \nabla_x^2 f + \frac{1}{2} \sum_{i=1}^{d_p} p_i\, a_i : \nabla_x^2 f,\end{aligned}$$

where ∇_x^2 denotes the Hessian. Setting $\mathcal{A}_p = \mathcal{L}_p - \mathcal{L}_0$, we obtain

$$\mathcal{A}_p f = \sum_{i=1}^{d_p} p_i \left[b_i \cdot \nabla_x f + \frac{1}{2}\, a_i : \nabla_x^2 f \right].$$

The operators $\mathcal{A}_p$ are linear with respect to parameter p. Moreover, the Koopman generators are linear with respect to parameter p. □

This means that for any linear combination of parameters $p = \sum_{i=1}^{d_p} \alpha_i\, p_i$ we obtain

$$\mathcal{L}_p = \mathcal{L}_0 + \sum_{i=1}^{d_p} \alpha_i\, \mathcal{L}_i.$$

In particular, if the ABM is linear with respect to the decision variable $y \in \mathcal{R}$ and the feasible decision space $\mathcal{R}$ is given by (6.2), then we can construct a surrogate model based on convex interpolation of Koopman generators $\mathcal{L}_i$ corresponding to the vertices of the convex polytope $\mathcal{R}$. The generators of the surrogate model for varying $y \in \mathcal{R}$ are obtained from

$$\mathcal{L} = \sum_{i=1}^{2^n} \alpha_i\, \mathcal{L}_i,$$

where $\sum_{i=1}^{2^n} \alpha_i = 1$ and $\alpha_i \in [0, 1]$.

6.2.2 Extended Voter Model

We consider the voter model with $d = 2$ opinions as defined in Section 2.2.3 with fully connected interaction network and set the transition rate to $\gamma_{12} = 1$, $\gamma_{21} = 2$ and $\gamma'_{12} = \gamma'_{21} = 0.1$. For instance, to win votes in election campaigns or to change individual mobility behavior to green technologies, we want to find a long-term policy that affects the transition rates γ_{12} and γ_{21} such that the expected number of agents of type S_1 is minimized. We are interested in the long-term effect of the policy on the ABM dynamics and assume only static costs. Additionally, we assume that the policy once applied to the model via $\gamma_{12} + y_1$ and $\gamma_{21} + y_2$, for $y = [y_1, y_2]^\top \in \mathcal{R} \subset \mathbb{R}^2$, will not change. In order to find the Pareto set, we set up a multi-objective optimization problem of the form (6.1) with feasible decision space $\mathcal{R} \coloneqq [-1, 5] \times [-2, 5]$ and objective function $F \colon \mathcal{R} \to \mathbb{R}^2$ with

$$f_1(y) = \mathbb{E}[X_1(t, y)/N], \tag{6.6a}$$

$$f_2(y) = y_1^2 + y_2^2, \tag{6.6b}$$

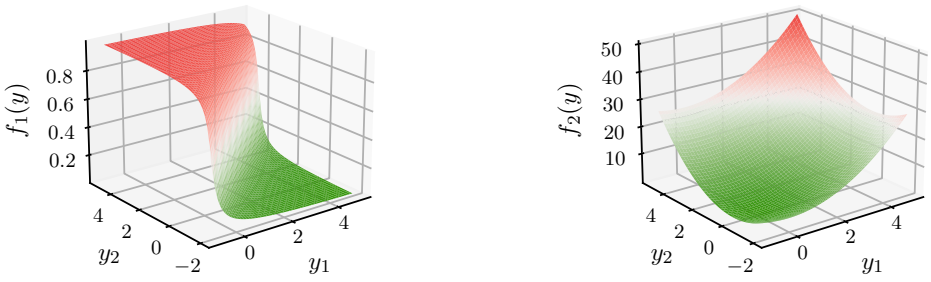
where $X_1(t, y)$ denotes the number of agents of type S_1 at time t depending on parameter $y \in \mathcal{R}$. As the first objective (6.6a) is expensive to evaluate, we construct a surrogate model based on Koopman generators and Theorem 6.2.1. In fact, drift and diffusion terms can be written as (6.4) and (6.5). Thus, for large numbers of agents N the ABM dynamics are parameter-affine. Using Algorithm 5.1, we learn four generator approximations corresponding to the four vertices of the feasible decision space $\mathcal{R}$ for $N = 500$ agents from 100 training points with each 100 samples. Having this, the objective function f_1 is given by

$$f_1(y) = \mathbb{E}[X_1(t, y)/N] \approx C_1(t, y),$$

where $C_1(t, y)$ denotes the data-driven ODE model associated with the ABM at time t depending on parameter $y \in \mathcal{R}$. Figure 6.2 shows both objective functions for all parameters $y \in \mathcal{R}$ and $t = 10$. Note that the SDE limit model might also be used, however, at a higher expense due to stochasticity. Also note that the second objective (6.6b) representing the costs of the policy is given by the modeler and problem. To demonstrate the procedure, we assume that any influence on the transitions rates γ_{12} and γ_{21} has a cost.

Before computing an outer approximation of the Pareto set, let us consider the following as an alternative to the surrogate model. Since we know the ground-truth model in this example (i.e., ODE (2.15) resp. SDE (2.16)) and since we are interested in the long-term behavior, i.e., $t \to \infty$, we can replace the objective (6.6a) by calculating the fixed points of $C_1(t, y)$ (in general solving ODE (2.15) resp. SDE (2.16) for a given number of agents N and time $t \geq 0$). Exploiting $c_2 = 1 - c_1$, the ODE limit process is given by

$$\frac{\mathrm{d}}{\mathrm{d}t} c_1 = c_1(1 - c_1)(\gamma_{21} + y_2 - \gamma_{12} - y_1) - c_1(\gamma'_{12} + \gamma'_{21}) + \gamma'_{21}.$$



(a) **(b)**

Figure 6.2: Objective functions f_1 and f_2 as defined in (6.6) for $N = 500$ agents depending on the policy $y = [y_1, y_2]^\top \in \mathcal{R} = [-1, 5] \times [-2, 5]$.

The fixed point as a function $\tilde{f}_1$ depending on y is given by

$$\tilde{f}_1(y) = \begin{cases} \frac{\gamma+y_2-y_1-(\gamma'_{12}+\gamma'_{21})+\sqrt{(\gamma'_{12}+\gamma'_{21}-(\gamma+y_2-y_1))^2+4\gamma'_{21}(\gamma+y_2-y_1)}}{2(\gamma+y_2-y_1)}, & \text{if} \quad \gamma + y_2 - y_1 \neq 0 \\ \frac{\gamma'_{21}}{\gamma'_{21}+\gamma'_{12}}, & \text{else,} \end{cases}$$

where $\gamma := \gamma_{21} - \gamma_{12}$, see Appendix A.2 for the deviation. Comparing the objective function surrogate based on the ODE fixed point equation with the surrogate model based on convex interpolation of Koopman generators, we see a good agreement of both surrogates for large time t (here $t = 10$). Figure 6.3 shows the error defined as pointwise difference between both surrogates.

After these preparations we can now apply Algorithm 6.1 to compute an outer approximation of the Pareto set. Figure 6.4 shows the computed coverings after 12 iterations plotted against objective function (6.6a) for both surrogate methods. As expected, we observe that the coverings are almost identical. For a refined feasible decision space, that is a subspace $\mathcal{R}^\star \subset \mathcal{R}$ with

$$\mathcal{R}^\star = [0.25, 0.75] \times [-0.75, -0.25] \subset \mathcal{R},$$

motivated, for instance, by a range of special interest, we compute another covering of the Pareto set. Figure 6.5 shows in (a) the computed covering of the Pareto set for $\mathcal{R}^\star$ and in (b) the image of $\mathcal{R}^\star$ under the objective function F (6.6) as light blue area as well as the approximated Pareto front (red/solid). To verify that our surrogate model approximated the dynamics of the ABM sufficiently well, we randomly choose some test points (blue/dots) for which we evaluate objective (6.6a) from 100 Monte Carlo simulations using the full-complexity agent dynamics. Figure 6.5 (c) and (d) show close-ups of two different kinds of test points. We observe on the one hand in Figure 6.5 (c) that test points covered by the boxes in Figure 6.5 (a) actually approximate the Pareto front of the ABM. These points are mapped on the Pareto front and thus are non-dominated. On the other hand, we observe that the marked test points (in general,

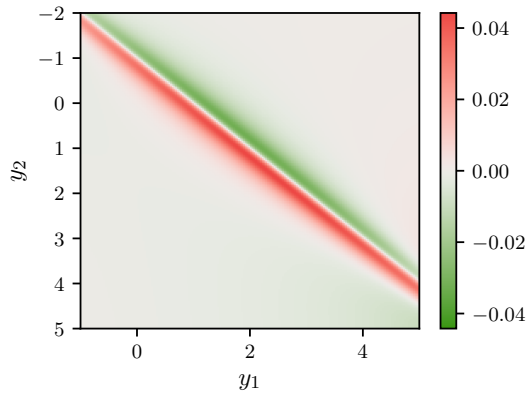


Figure 6.3: Pointwise difference between the surrogate model based on convex interpolation of Koopman generators and the objective function surrogate based on the fixed point equation of the ODE limit model.

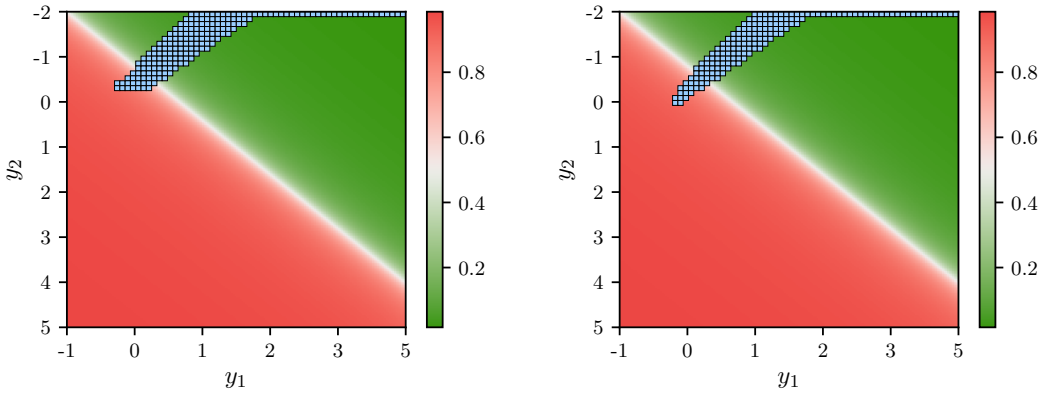


Figure 6.4: Pareto set coverings after 12 iterations obtained using (a) the surrogate model based on convex interpolation of Koopman generators and (b) the objective function surrogate based on the ODE fixed point equation plotted against objective function (6.6a) for $N = 500$ agents and feasible decision space $\mathcal{R} = [-1, 5] \times [-2, 5]$.

points that are not covered in Figure 6.5 (a)) are located on the right-hand side of the Pareto front and thus are dominated, see Figure 6.5 (d). The error bars indicate a 99.9 % confidence level.

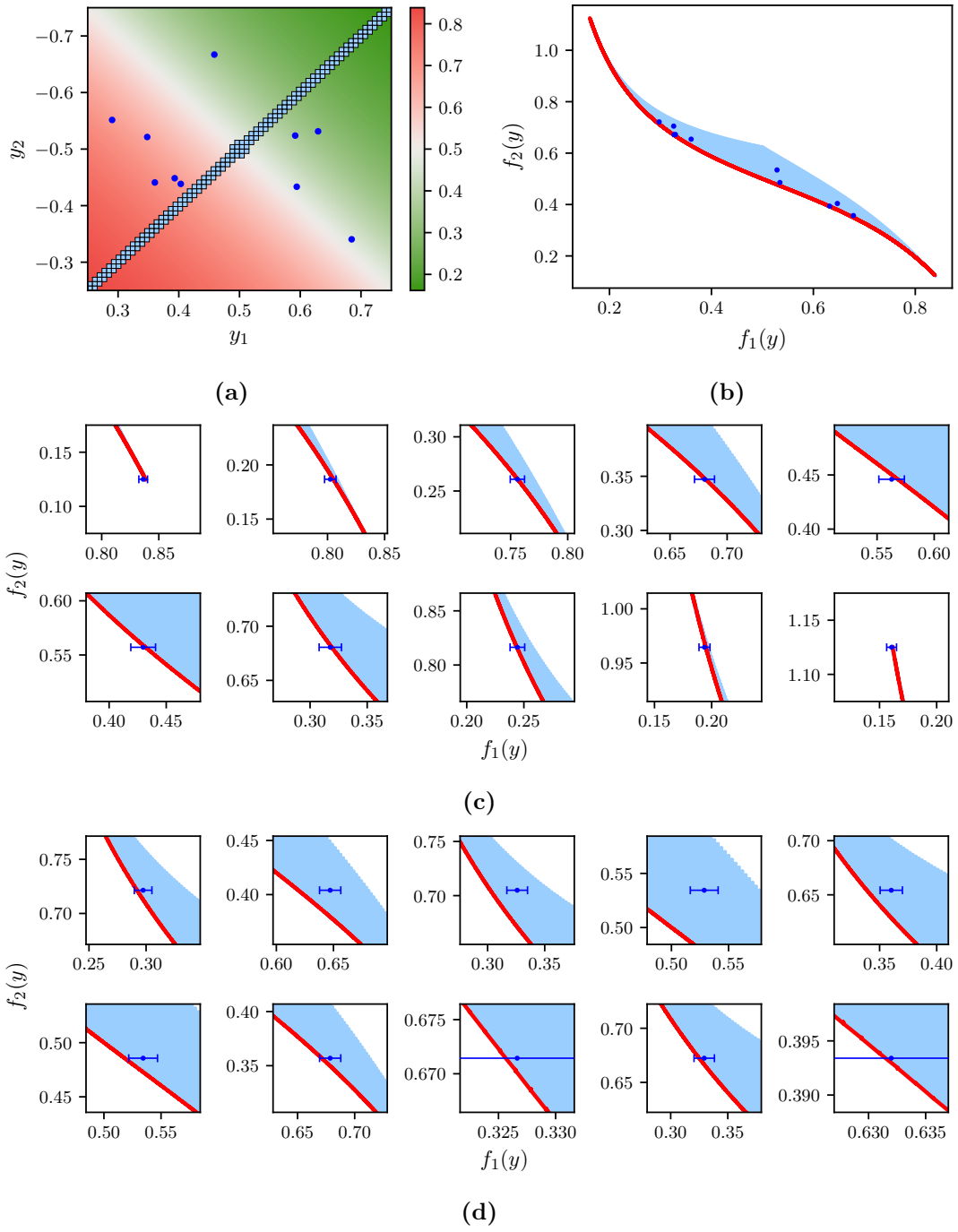


Figure 6.5: (a) Pareto set covering for refined feasible decision space $\mathcal{R}^\star \subset \mathcal{R}$ after 12 iterations using the objective function surrogate. (b) Pareto front (red/solid) and image of $\mathcal{R}^\star$ (light blue area) under the objective function (6.6). (c) and (d) close-ups of non-dominated and dominated test points (blue/dots), respectively, which have been computed via the full-complexity ABM. Error bars indicate a 99.9 % confidence level.

6.2.3 Civil Violence Model

Let us now consider the civil violence model as introduced in Section 2.2.5 using the parameters given in Table 2.2. Assume that we want to avoid underutilization of labor force while simultaneously keeping riots at a low level. Thus, we are interested in the long-term effect of the number of employed officers y on the the ABM dynamics for a monitoring period $[t_1, t_2]$, $t_2 \gg t_1 > t_0 = 0$. We assume that the employment costs for the officers is constant in time and that the number of officers does not change. In order to compute an outer approximation of the Pareto set, we set up a multi-objective optimization problem of the form (6.1) with objective function $F\colon \mathcal{R} \to \mathbb{R}$,

$$f_1(y) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} X_1(\tau, y)/N \,\mathrm{d}\tau, \quad t_2 > t_1 > t_0 \tag{6.7a}$$

$$f_2(y) = 500 - 0.5\,y^2 + 0.01\,y^3 \tag{6.7b}$$

and feasible decision space $\mathcal{R} \coloneqq [20, 100]$. Here, $X_1(t, y)$ denotes the number of active citizen at time t depending on the number of officers $y \in \mathcal{R}$. Again, it is not feasible to evaluate the objective (6.7a) directly using the ABM. Instead, we construct a surrogate model based on Koopman generators. As a ground-truth model is missing, we cannot check if the civil violence model actually fulfills the requirements of Theorem 6.2.1. However, numerical analysis shows that convex interpolation between only two Koopman generators leads to unsatisfactory results such that we learn nine Koopman generator approximations for $y \in \{20, 30, \ldots, 90\}$ as described in Section 5.2.4. Then, the objective function f_1 is given by

$$f_1(y) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} X_1(\tau, y)/N \,\mathrm{d}\tau \approx \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} C_1(\tau, y) \,\mathrm{d}\tau, \quad t_2 > t_1 > t_0$$

where $C_1(t, y)$ denotes the data-driven ODE model associated with the ABM at time t depending on decision variable $y \in \mathcal{R}$. We set $t_1 = 200$ and $t_2 = 250$. Again, objective (6.7b) is given by the modeler having its minimum at $y = 33\frac{1}{3}$. Figure 6.6 (a) shows both objective functions for all $y \in \mathcal{R}$. Again, we use Algorithm 6.1 to approximate the Pareto set of the ABM dynamics. The result after 8 iterations is displayed in Figure 6.6 (a) as blue/shaded area. We observe three potential Pareto set coverings $[33, 65]$, $[67, 79]$ and $[87, 100]$. The corresponding Pareto front (red/solid) is shown in Figure 6.6 (b). Additionally, we estimate for test points (blue with error bars) mean and standard deviation of the objective function (6.7a) from 1000 Monte Carlo simulations using the full-complexity ABM dynamics. We observe a good agreement of the coverings $[32, 65]$ and $[67, 79]$ with the Pareto front of the ABM dynamics. However, the covering $[87, 100]$ is not correct. Indeed, a refinement on $\mathcal{R}^\star = [75, 100]$ confirms that this covering does not approximate any part the Pareto set. Refining the feasible decision space on $\mathcal{R}^\star = [30, 80]$ closes the gap between the coverings $[32, 65]$ and $[67, 79]$ and results in a Pareto set approximation $[32, 77]$, see Figure 6.6 (a) hatched area. Note that other values for t_1 and t_2 might change the Pareto set and the approximation accuracy.

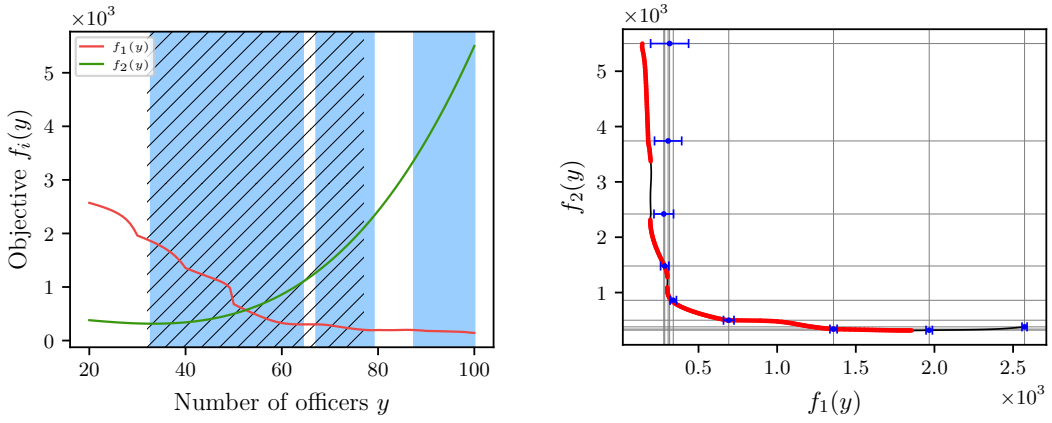


Figure 6.6: (a) Objective functions f_1 and f_2 as defined in (6.7) for the parameters given in Table 2.2 depending on the number of officers $y \in \mathcal{R} = [20, 100]$. The approximated Pareto set is indicated in blue/shaded. (b) Pareto front (red/solid) corresponding to the Pareto set in (a) and image of $\mathcal{R}$ under F (black/solid). Test points (blue with error bars) indicating mean and standard deviation of the objective function (6.7a) estimated from 1000 Monte Carlo simulations using the full-complexity ABM. While the calculated Pareto set coverings $[32, 65]$ and $[67, 79]$ actually approximate the Pareto front of the ABM dynamics, this is not true for the covering $[87, 100]$. Restricting $\mathcal{R}$ on $\mathcal{R}^\star = [30, 80]$ closes the gap between the left and middle coverings and results in the hatched area, i.e., $[32, 77]$. Restriction on $\mathcal{R}^\star = [75, 100]$ confirms that the third covering on the far right does not any part of the Pareto set.

Remark 6.2.2. Similar as for the voter model, we can also construct a surrogate for the objective function (6.7a), e.g., based on interpolation of points for which the objective is evaluated via the full-complexity ABM. Although both methods compute essentially the same outer approximation of the Pareto set, the objective function surrogates are significantly faster to evaluate than the respective surrogate models. However, they are rigid and do not have the same flexibility as the surrogate models (e.g., with respect to time t). Table 6.1 summarizes and compares the computation times of both surrogate types.

Remark 6.2.3. For the sake of illustration, we considered the case where only one objective functions is defined by the agent dynamics. However, it is possible to find the Pareto set with respect to several parameters of the ABM. Additionally, the proposed approach can also be used to calibrate agent-based systems against target metrics. In this case, the goal might be to minimize the errors with respect to the first and second order moments or to calibrate the model such that the dynamics agree with experimental observations.

Table 6.1: Computation times of Algorithm 6.1 to solve the multi-objective optimization problems analyzed in Sections 6.2.2 and 6.2.3 comparing both surrogates methods. The computation was performed using an Intel i7-8569U CPU and 16 GB of RAM.

		Surrogate type	
Agent-based model	Iterations	Model	Objective function
Voter model	4	50.1673 sec	0.0225 sec
	8	23.1 min	0.2861 sec
	12	13 h	9.2361 sec
	16	-	4.8 min
Civil violence model	4	47.0111 min	0.0972 sec
	8	4.87 d	5.6997 sec
	12	-	21.7789 min

Remark 6.2.4. In the preceding two sections, the focus is not on (efficiently) solving multi-objective optimization problems, but in demonstrating that surrogate models based on the Koopman generator are sufficiently well suited for this purpose. Especially in the case of the voter model on complete networks, it is more efficient to use the well-known ODE or SDE limit models as surrogates (cf. Remark 6.2.2 and Table 6.1). However, such limit models are not necessarily known or do not even exist at all, such as in the case of the civil violence model. Thus, building surrogates using the Koopman generator provides a way to solve multi-objective optimization problems with complex agent-based systems.

7 Conclusion and Outlook

Given the wealth of data available in agent-based modeling, data-driven methods are the next logical step in gaining new insights into the emergent collective time-evolving phenomena arising from the interactions of agents. To answer tomorrow's questions, understanding the dynamical evolution of aggregate quantities is of great interest and importance, for example, to initiate the transition to environmentally friendly technologies or to fight pandemics. Agent-based systems modeling social interactions provide a high level of detail into artificial societies but at often high computational costs. Therefore, the question addressed in this thesis is how to learn reduced models from data for large-scale agent-based systems that allow for analysis and prediction of the emergent collective behavior.

Contributions. Data-driven methods based on transfer operators help with a plethora of tasks such as metastability analysis, system identification or model reduction. In Chapter 3, we presented an extension of the classical EDMD algorithm to compute a finite-dimensional matrix approximation of the infinitesimal generator of the Koopman or Perron–Frobenius operator from data. The presented framework gEDMD applies to both deterministic and non-deterministic dynamical systems, and allows a decomposition into eigenfunctions, eigenvalues and modes, which then can be used for system identification, model reduction or the discovery of conservation laws. Furthermore, we showed that gEDMD is related to other methods such as SINDy, which results as a special case in the deterministic setting.

An important aspect is the behavior of agent-based systems for large numbers of agents or long time scales of interest or both. In Chapter 4, we therefore studied the metastable behavior of ABMs given as continuous-time Markov jump processes and their pathwise approximations by SDEs for large population sizes. We demonstrated in Section 4.1 that the Perron–Frobenius operator can reveal metastable structures and long time scales associated with rare events for both the ABM and the SDE process, and that for large numbers of agents these structures are close to each other. Due to the pathwise closeness, we then characterized the long-term behavior of the ABM via the corresponding SDE using Ulam's method, i.e., we generated many short trajectories of finite length from the SDE. Since the computational cost of the SDE, unlike the ABM, is essentially not dependent on the number of agents, this is of high relevance. However, this approach reaches its limits when we consider rare events that occur on exponential time scales. We illustrated in Section 4.2 that in this case large deviation theory can characterize the asymptotic rate functions for both processes. Using the voter model, we demonstrated that they are close under certain conditions such that

the metastable behavior of the ABM can be described via the corresponding SDE for sufficiently large agent numbers. However, in general a characterization via the SDE is not enough, cf. Section 4.2.4. When we consider more complex ABMs, large deviation theory might also not be helpful to describe the rare events on long time scales, since it might be unclear, for example, whether the SDE can replace the ABM, or how the associated Hamilton–Jacobi equations can be solved efficiently, if this is the case. In this context, the transfer operator approach provides a practical tool to identify metastable structures, quasi-stationary distributions and to approximate mean first exit times or transition rates. In addition, it allows us to use the finite-time, pathwise approximation results to characterize the behavior on long, but not necessarily exponentially long, time scales. In this way, the transfer operator approach connects the pathwise approximation by differential equations for large population sizes on finite time scales with methods for describing the long-term behavior on exponentially long time scales.

In Chapter 5, we used the insights of the previous two chapters and showed how the Koopman generator can be used to obtain reduced models represented as ODEs or SDEs from aggregate state variables of agent-based systems, however, this time directly from noisy data. We considered four problems of different complexity to benchmark the procedure.

In Sections 5.2.1 and 5.2.2, we considered the voter model with complete and clustered interaction networks of homogeneous agents, where each agent might interact with any other adjacent agent. In the first case, the temporal evolution of the ABM was given by a continuous-time Markov jump process, while we simulated the ABM on clustered networks in discrete time. We showed that the data-driven reduced models are predictive for a sufficiently large number of agents and consistent with the respective pathwise SDE limit models for both complete and clustered networks. The results in Section 5.2.2 also revealed that the aggregation of state variables led to an approximation error in the population state model due to the incomplete, clustered network, which then shortened the interval in which the data-driven model and its SDE approximation agree. The results also show that a sufficiently large number of agents or, alternatively, a high enough connectivity between clusters is required for the data-driven reduced model and the SDE model to agree. Additionally, for arbitrary networks the prediction horizon might be even shorter, cf. Remark 5.2.2. This implies that if the state of an ABM strongly depends on the spatial structure, e.g., clustering, coexistence or spatial heterogeneity, this must be taken into consideration. With the third benchmark problem in Section 5.2.3, we showed that the proposed approach can also identify reduced models for agent-based systems that are not bound to interactions defined on a network and where agents move freely in space. The reduced model captures the qualitative behavior of the aggregate state of the predator-prey system.

For both the predator-prey model and the previous two benchmark problems, the choice of basis functions was straightforward since either a theoretical limit model was known or a model similarity (here to the Lotka–Volterra differential equations) could be exploited. In Section 5.2.4, however, this choice was not obvious any longer. Physical insights about the civil violence model allowed us to find an extended set of basis

functions. Including the arrest probability function, the set of basis functions is rich enough to reproduce qualitatively the typical dynamical behavior of the aggregate state variables. The results show once more that the choice of the arrest probability function is largely responsible for dynamic behavior of the model.

To successfully identify a reduced model from data, the presented approach requires accurate, *pointwise* estimates of the drift and diffusion terms. To avoid nonsparse solutions of the generator approximation in the case of inaccuracies or insufficiency, iterative hard thresholding or denoising techniques might be applied to improve the results. Additionally, we assumed in this work that all types of agents are present in sufficient numbers, since in this case the discrete dynamics can be meaningfully represented by continuous differential equations. If the number of agents is small, the discreteness of the system can induce metastability such that this approach might fail, cf. Remark 4.1.2.

In Chapter 6, we demonstrated how multi-objective optimization problems associated with agent-based systems can be solved with the aid of the data-driven reduced models computed in Chapter 5. We showed in Section 6.2.1 that under certain conditions linear interpolation between Koopman generators associated with non-deterministic dynamical systems is possible, which allows for the construction of surrogates with varying parameters. The proposed approach can be used for both support of decision-making processes and model calibration. We demonstrated the procedure for the voter model on complete networks and the civil violence model. In Section 6.2.2, the goal was to achieve a change in the long-term majority ratio of the voter model by influencing two of the four rate constants. We tested the surrogate model based on Koopman generators against an analytical derived surrogate for the fixed points of the corresponding ODE limit model and found good agreement of both Pareto set coverings. In Section 6.2.3, we showed that this approach also leads to good results for the civil violence model, where we sought the Pareto set for the tradeoffs between labor force utilization and security in society. With the ever-increasing spread of ABMs, the construction of data-driven surrogates is of great importance especially in the case when limit models are unknown or non-existent. Koopman generator based surrogates make it possible to solve multi-objective optimization problems, where objectives are defined by the dynamical behavior of the agents and thus would otherwise be computationally impossible due to the very expensive objective functions.

Open Questions and Perspectives. Data-driven model reduction of large-scale agent-based systems does not only reduce the computational effort but also opens up new possibilities for further analysis and applications. Nevertheless, there are still unanswered questions and future perspectives that need to be addressed:

- Approximating the Koopman generator associated with dynamical systems has proven to be beneficial for many applications. Open problems concern, e.g., the convergence of gEDMD to the Koopman generator as the number of basis functions goes to infinity, the question which part of the spectrum is approximated

in the case of continuous spectra or non-compactness, or the extension to non-autonomous dynamical systems. Another important issue is the optimal choice of basis functions in order to avoid over- and underfitting.

- Open questions related to agent-based systems concern the optimal choice of aggregate variables. From the application point of view, the population state space provides important and obvious aggregate variables. However, there might exist other non-obvious aggregate states. For instance, instead of using the number of insurgents and detained citizens as aggregate state variables to learn a reduced model for the civil violence model, we might also consider the perceived arrest probability of the total population as aggregate variable. The question is how non-obvious aggregate can be found. Alternatively, we might extend the population state space to include intermediate states of agitation given by the arrest probability function. Here, the question is whether this increase in dimension actually helps to find good reduced models. In this context kernel-based [83] or tensor-based [106] variants of gEDMD can help to mitigate the curse of dimensionality that makes many data-driven methods unacceptably expensive to use in the case of high-dimensional systems.

- The acquisition of data is another problem. Especially for large-scale agent-based systems like the Mobility Transition Model, high computational costs require a proper method for data acquisition. The question is how inside knowledge of the model can be exploited to mitigate the enormous computational effort.

- In Chapter 4, the analysis using transfer operators was restricted to agent-based models with complete interaction networks and homogeneous agents. In Chapter 5, we went beyond this and learned reduced models for more complex ABMs with both incomplete interaction networks and spatially dependent interactions. To some extent a macro-scale representation by ODEs or SDEs is useful. However, with increasing spatial complexity and influence, the use of this representation leads also to a higher loss of information. It might fail completely if spatial interaction or interaction with the space itself significantly influence the behavior of the agents and therefore the outcome of the model. Meso-scale models, e.g., systems of compartment chemical master equations [173], are intermediate between micro- and macro-scale models and might be helpful in this case. It would be interesting to investigate how data-driven approaches can help to find meso-scale models.

- In addition to metastability analysis or optimization, also control of agent-based systems is an important and interesting research direction. Here, the question is how the reduced models can be used to find (optimal) control schemes that steer the system into a desired state. For example, how can we use them to find time-dependent harvesting schedules for systems like the predator-prey model, or emergency response strategies to avoid outbreaks in the civil violence model without permanently increasing the number of officers? Another research direc-

tion concerns the concept of derivatives for agent-based systems. Recent advances use, e.g., adjoint concepts [6, 54] for efficient gradient computation.

Future research should provide answers to these questions.

Data Availability. All the presented methods were implemented using Python and MATLAB. The code for gEDMD and some of the examples discussed in Chapter 3 are available at `https://github.com/sklus/d3s`. The ABM codes used for the simulations in this work and the tools of Chapter 5 can be found at `https://github.com/Henningston/ABMs` with corresponding data deposited at `https://doi.org/10.5281/zenodo.4522119`. Supplementary data can be found at `https://doi.org/10.5281/zenodo.5561165`.

A Appendix

A.1 Quadruple-Well Problem

We compute a representation of the Koopman generator for the quadruple-well problem. Choosing a dictionary consisting of monomials up to order 4, we generated 8000 random training points in $\mathbb{X} = [-2, 2] \times [-2, 2]$ and use the exact values for $b(x)$ and $\sigma(x)$. Then, the first six columns of the generator approximation are given by

$$
\begin{array}{c}
\\
1 \\
x_1 \\
x_2 \\
x_1^2 \\
x_1\,x_2 \\
x_2^2 \\
x_1^3 \\
x_1^2\,x_2 \\
x_1\,x_2^2 \\
x_2^3 \\
x_1^4 \\
x_1^3\,x_2 \\
x_1^2\,x_2^2 \\
x_1\,x_2^3 \\
x_2^4
\end{array}
\begin{array}{c}
\begin{array}{cccccc} 1 & x_1 & x_2 & x_1^2 & x_1\,x_2 & x_2^2 \end{array} \\
\left[\begin{array}{cccccc}
0 & 0 & 0 & 1 & 0 & 1 \\
0 & 4 & 0 & 0 & 0 & 0 \\
0 & 0 & 4 & 0 & 0 & 0 \\
0 & 0 & 0 & 8 & 0 & 0 \\
0 & 0 & 0 & 0 & 9 & 0 \\
0 & 0 & 0 & 0 & 0 & 8 \\
0 & -4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -4 & 0 & 0 & 0 \\
0 & 0 & 0 & -8 & 0 & 0 \\
0 & 0 & 0 & 0 & -4 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & -4 & 0 \\
0 & 0 & 0 & 0 & 0 & -8
\end{array}\right]
\end{array}.
$$

Using columns two and three, we correctly recover the drift term b. Following (3.3), the entries of matrix a are given by

$$
\begin{aligned}
a_{11}(x) &= (\mathcal{L}\psi_4)(x) - 2\,b_1(x)\,x_1 &&= 1 + x_1^2\,x_2^2, \\
a_{12}(x) &= (\mathcal{L}\psi_5)(x) - b_1(x)\,x_2 - b_2(x)\,x_1 &&= x_1\,x_2, \\
a_{22}(x) &= (\mathcal{L}\psi_6)(x) - 2\,b_2(x)\,x_2 &&= 1,
\end{aligned}
$$

which is in fact $\sigma\,\sigma^\top$. Note that monomials of order up to 3 are not sufficient to recover the drift a.

A.2 Fixed Points of ODE Limit Model for Two Types of Agents

Consider the ODE limit model (2.15) with two types S_1 and S_2 of agents and transition rates γ_{12}, γ_{21}, γ'_{12} and γ'_{21}. Then, exploiting $c_1 + c_2 = 1$, the ODE process can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}c_1 = c_1(1-c_1)(\gamma_{21}-\gamma_{12}) - c_1(\gamma'_{12}+\gamma'_{21}) + \gamma'_{21}.$$

We calculate the fixed points. For the general case, we assume $\gamma_{12} \neq \gamma_{21}$ and $\gamma'_{12} \neq \gamma'_{21}$. We obtain

$$\begin{aligned} 0 = \frac{\mathrm{d}}{\mathrm{d}t}c_1 &= c_1(1-c_1)(\gamma_{21}-\gamma_{12}) - c_1(\gamma'_{12}+\gamma'_{21}) + \gamma'_{21} \\ &= c_1^2 + c_1\left(\frac{\gamma'_{12}+\gamma'_{21}}{\gamma_{21}-\gamma_{12}} - 1\right) - \frac{\gamma'_{21}}{\gamma_{21}-\gamma_{12}} \end{aligned}$$

and using p-q-formula

$$\begin{aligned} c_1^* &= -\frac{\gamma'_{12}+\gamma'_{21}}{2(\gamma_{21}-\gamma_{12})} + \frac{1}{2} \pm \frac{\sqrt{(\gamma'_{12}+\gamma'_{21}-(\gamma_{21}-\gamma_{12}))^2 + 4\gamma'_{21}(\gamma_{21}-\gamma_{12})}}{2(\gamma_{21}-\gamma_{12})} \\ &= \frac{(\gamma_{21}-\gamma_{12}) - (\gamma'_{12}+\gamma'_{21}) \pm \sqrt{(\gamma'_{12}+\gamma'_{21}-(\gamma_{21}-\gamma_{12}))^2 + 4\gamma'_{21}(\gamma_{21}-\gamma_{12})}}{2(\gamma_{21}-\gamma_{12})} \\ &= \frac{\gamma - (\gamma'_{12}+\gamma'_{21}) \pm \sqrt{(\gamma'_{12}+\gamma'_{21}-\gamma)^2 + 4\gamma'_{21}\gamma}}{2\gamma}, \end{aligned}$$

where $\gamma \coloneqq \gamma_{21} - \gamma_{12}$. In order to guarantee $0 \leq c_1^* \leq 1$, we have to choose $+$ in the above formula due to the following observation. For the case $\gamma > 0$, we have

$$\gamma - (\gamma'_{12}+\gamma'_{21}) < \sqrt{(\gamma - (\gamma'_{12}+\gamma'_{21}))^2 + 4\gamma'_{21}\gamma},$$

and therefore subtracting the square root leads to a negative nominator and $c_1^* < 0$. If $\gamma < 0$, on the other hand, it holds

$$\left|\gamma - (\gamma'_{12}+\gamma'_{21})\right| > \sqrt{(\gamma - (\gamma'_{12}+\gamma'_{21}))^2 + 4\gamma'_{21}\gamma}$$

and with this the $-$ solution leads to

$$c_1^* = \frac{\gamma - (\gamma'_{12}+\gamma'_{21}) - \sqrt{(\gamma'_{12}+\gamma'_{21}-\gamma)^2 + 4\gamma'_{21}\gamma}}{2\gamma} > \frac{2(\gamma - (\gamma'_{12}+\gamma'_{21}))}{2\gamma} > 1.$$

Therefore, for $\gamma = \gamma_{21} - \gamma_{12} \neq 0$ it holds that

$$c_1^* = \frac{\gamma - (\gamma'_{12}+\gamma'_{21}) + \sqrt{(\gamma'_{12}+\gamma'_{21}-\gamma)^2 + 4\gamma'_{21}\gamma}}{2\gamma}. \tag{A.1}$$

For the case where $\gamma_{12} = \gamma_{21}$, $\gamma'_{12} \neq \gamma'_{21}$, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} c_1 = -c_1(\gamma'_{12} + \gamma'_{21}) + \gamma'_{21}$$

and thus

$$c_1^* = \frac{\gamma'_{21}}{\gamma'_{12} + \gamma'_{21}}.$$

For the case where $\gamma_{12} \neq \gamma_{21}$, $\gamma' \coloneqq \gamma'_{12} = \gamma'_{21}$, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} c_1 = c_1(1 - c_1)(\gamma_{21} - \gamma_{12}) + \gamma'(1 - 2c_1)$$

and, using Equation (A.1),

$$c_1^* = \frac{\gamma - 2\gamma' + \sqrt{4\gamma'^2 + \gamma^2}}{2\gamma}.$$

For the case where $\gamma_{12} = \gamma_{21}$, $\gamma' \coloneqq \gamma'_{12} = \gamma'_{21}$, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} c_1 = \gamma'(1 - 2c_1)$$

such that $c_1^* = \frac{1}{2}$. In all cases, it holds that $c_2^* = 1 - c_1^*$.

A.3 Reduced Two-Dimensional System

We consider the voter model with $d = 3$ opinions as defined in Section 2.2.3 and assume a fully connected interaction network. As the number of agents is assumed to be constant, this is essentially a two-dimensional system. Since it holds that $c_3(t) = 1 - c_1(t) - c_2(t)$, we obtain the drift $b\colon \mathbb{X} \to \mathbb{R}^2$ given by

$$\begin{aligned} b_1(c) &= (\gamma_{13} - \gamma_{31})\, c_1^2 + (\gamma_{21} - \gamma_{12} + \gamma_{13} - \gamma_{31})\, c_1 c_2 \\ &\quad + (\gamma_{31} - \gamma_{13} - \gamma'_{12} - \gamma'_{13} - \gamma'_{31})\, c_1 + (\gamma'_{21} - \gamma'_{31})\, c_2 + \gamma'_{31}, \end{aligned} \tag{A.2a}$$

$$\begin{aligned} b_2(c) &= (\gamma_{23} - \gamma_{32})\, c_2^2 + (\gamma_{12} - \gamma_{21} + \gamma_{23} - \gamma_{32})\, c_1 c_2 \\ &\quad + (\gamma_{32} - \gamma_{23} - \gamma'_{21} - \gamma'_{23} - \gamma'_{32})\, c_2 + (\gamma'_{12} - \gamma'_{32})\, c_1 + \gamma'_{32}. \end{aligned} \tag{A.2b}$$

The diffusion term $a\colon \mathbb{X} \to \mathbb{R}^{2\times 2}$ with $a(c) = a(c)^\top = (a_{ij}(c))$ is given by

$$\begin{aligned} a_{11}(c) &= \frac{1}{N}\Big((-\gamma_{13} - \gamma_{31})\, c_1^2 + (\gamma_{12} + \gamma_{21} - \gamma_{13} - \gamma_{31})\, c_1 c_2 + (\gamma_{13} + \gamma_{31} + \gamma'_{12} \\ &\quad + \gamma'_{13} - \gamma'_{31})\, c_1 + (\gamma'_{21} - \gamma'_{31})\, c_2 + \gamma'_{31}\Big), \end{aligned} \tag{A.3a}$$

$$a_{12}(c) = -\frac{1}{N}\Big((\gamma_{12} + \gamma_{21})\, c_1 c_2 + \gamma'_{12}\, c_1 + \gamma'_{21}\, c_2\Big), \tag{A.3b}$$

$$\begin{aligned} a_{22}(c) &= \frac{1}{N}\Big((-\gamma_{23} - \gamma_{32})\, c_2^2 + (\gamma_{12} + \gamma_{21} - \gamma_{23} - \gamma_{32})\, c_1 c_2 \\ &\quad + (\gamma_{23} + \gamma_{32} + \gamma'_{21} + \gamma'_{23} - \gamma'_{32})\, c_2 + (\gamma'_{12} - \gamma'_{32})\, c_1 + \gamma'_{32}\Big). \end{aligned} \tag{A.3c}$$

The remaining entries are given by

$$\begin{aligned} a_{13} &= a_{11} - a_{12}, \\ a_{23} &= a_{22} - a_{12}, \\ a_{33} &= a_{11} + a_{22} + 2\,a_{12}. \end{aligned}$$

The entries of matrix L_N can be reconstructed using the coefficients appearing in Equation (A.2) and Equation (A.3) via Formula (3.3). For instance, we obtain

$$l_{22} = \gamma_{31} - \gamma_{13} - \gamma'_{12} - \gamma'_{13} - \gamma'_{31}.$$

Note that the indices ij depend on the ordering of the basis elements.

Zusammenfassung

Die Modellierung sozialer Systeme und die Untersuchung ihres dynamischen Verhaltens spielt in vielen Forschungsbereichen eine wichtige Rolle. Agentenbasierte Modellierung ermöglicht einen hohen Detaillierungsgrad künstlicher Gesellschaften, indem das Modell aus der Perspektive der Agenten beschrieben wird. Die Interaktionen der Agenten, die oft durch einfache Regeln vorgegeben sind, führen zu komplexen, sich zeitlich entwickelnden Mustern, deren Verständnis von großer Bedeutung zum Beispiel für die Vorhersage und Beeinflussung von Epidemien ist. Analyse und Simulation werden jedoch oft unverhältnismäßig zeitaufwendig, wenn die Anzahl der Agenten oder die betrachtete Zeitskala groß ist. Diese Arbeit widmet sich daher dem Erlernen deutlich reduzierter Modelle großer agentenbasierter Systeme aus Simulationsdaten. Es wird gezeigt, wie auf Transferoperatoren basierende, datengetriebene Methoden verwendet werden können, um reduzierte, durch gewöhnliche oder stochastische Differentialgleichungen darstellbare Modelle zu finden, die das dynamische Verhalten größerer Gruppen oder ganzer Populationen beschreiben und somit die Analyse und Vorhersage agentenbasierter Systeme ermöglichen. Dazu wird zunächst eine Erweiterung von EDMD vorgestellt, um den Koopman Generator anhand von Daten zu approximieren. Die gEDMD genannte Methode kann zur Berechnung von Eigenfunktionen, Eigenwerten und Moden des Generators sowie zur Systemidentifikation und Modellreduktion sowohl deterministischer als auch nichtdeterministischer dynamischer Systeme verwendet werden. Im nachfolgenden Kapitel wird das Langzeitverhalten bestimmter agentenbasierter Modelle und deren pfadweisen Approximationen durch stochastische Differentialgleichungen für große Agentenanzahlen mittels Transferoperatoren analysiert. Es wird gezeigt, dass der Transferoperatoransatz unter bestimmten Bedingungen die pfadweisen Approximationen auf endlichen Zeitskalen mit Methoden zur Beschreibung des Verhaltens auf möglicherweise exponentiell langen Zeitskalen verbindet. Dies bedeutet, dass die pfadweisen Näherungen auf endlicher Zeitskala genutzt werden können, um das metastabile Verhalten auf langen Zeitskalen mit Hilfe von Transferoperatoren zu charakterisieren. Dies kann die Rechenkosten erheblich reduzieren. Der dritte Teil befasst sich mit der datengesteuerten Modellreduktion, da in vielen Fällen keine analytischen Grenzmodelle bekannt oder vorhanden sind. Es wird demonstriert, wie die Koopman-Operatortheorie verwendet werden kann, um die maßgeblichen Gleichungen agentenbasierter Systeme direkt aus Simulationsdaten herzuleiten. Mittels Testprobleme wird gezeigt, dass die datengetriebenen Modelle für ausreichend große Populationen gut mit analytischen Grenzwertgleichungen übereinstimmen und dass die reduzierten Modelle sogar in Fällen, die weit vom Grenzwert entfernt sind oder wenn keine Grenzwertgleichungen bekannt sind, Vorhersagen ermöglichen. Zum Schluss wird ein Ansatz zur Mehrzieloptimierung agentenbasierter Systeme mit Hilfe von datengetriebenen, auf dem Koopman Generator basierenden Ersatzmodellen präsentiert und damit das Potenzial der vorgestellten Methode aufgezeigt. Insbesondere bei unbekannten oder inexistenten Grenzmodellen macht dieser Ansatz Mehrzieloptimierungsprobleme lösbar, die andernfalls aufgrund sehr teurer Zielfunktionen rechnerisch undurchführbar wären.

Bibliography

[1] S. Abar, G. K. Theodoropoulos, P. Lemarinier, and G. M. P. O'Hare. Agent Based Modelling and Simulation tools: A review of the state-of-art software. *Computer Science Review*, 24:13 – 33, 2017.

[2] B. Afsar, D. Podkopaev, and K. Miettinen. Data-driven Interactive Multiobjective Optimization: Challenges and a Generic Multi-agent Architecture. *Procedia Computer Science*, 176:281–290, 2020. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020.

[3] D. Aldous. Interacting particle systems as stochastic social dynamics. *Bernoulli*, 19(4):11221149, Sep 2013.

[4] G. An, B. G. Fitzpatrick, S. Christley, P. Federico, A. Kanarek, R. Miller Neilan, M. Oremland, R. Salinas, R. Laubenbacher, and S. Lenhart. Optimization and Control of Agent-Based Models in Biology: A Perspective. *Bulletin of Mathematical Biology*, 79(1):63–87, 2017.

[5] G. An, Q. Mi, J. Dutta-Moscato, and Y. Vodovotz. Agent-based models in translational systems biology. *Wiley interdisciplinary reviews. Systems biology and medicine*, 1(20835989):159–171, 2009.

[6] P. Andelfinger. Differentiable Agent-Based Simulation for Gradient-Guided Simulation-Based Optimization. pages 27–38, 05 2021.

[7] H. Arbabi, M. Korda, and I. Mezić. A data-driven Koopman model predictive control framework for nonlinear flows. *2018 IEEE Conference on Decision and Control (CDC)*, pages 6409–6414, 2018.

[8] M. Assaf and B. Meerson. WKB theory of large deviations in stochastic populations. *Journal of Physics A: Mathematical and Theoretical*, 50(26):263001, 2017.

[9] S. Banisch, R. Lima, and T. Araújo. Agent based models and opinion dynamics as Markov chains. *Social Networks*, 34(4):549–561, 2012.

[10] B. Bartkowski, M. Beckmann, M. Drechsler, A. Kaim, V. Liebelt, B. Müller, F. Witing, and M. Strauch. Aligning Agent-Based Modeling With Multi-Objective Land-Use Allocation: Identification of Policy Gaps and Feasible Path-

ways to Biophysically Optimal Landscapes. *Frontiers in Environmental Science*, 8(July):1–15, 2020.

[11] J. R. Baxter and J. S. Rosenthal. Rates of convergence for everywhere-positive markov chains. *Statistics & probability letters*, 22(4):333–338, 1995.

[12] M. B. Berkemeier and S. Peitz. Derivative-Free Multiobjective Trust Region Descent Method Using Radial Basis Function Surrogate Models. *Mathematical and Computational Applications*, 26(2), 2021.

[13] T. Biancalani, L. Dyson, and A. J. McKane. Noise-induced bistable states and their mean switching time in foraging colonies. *Physical Review Letters*, 112(3):1–5, 2014.

[14] M. J. Blondin and M. Hale. An Algorithm for Multi-Objective Multi-Agent Optimization. In *2020 American Control Conference (ACC)*. IEEE, jul 2020.

[15] P. Bolzern, P. Colaneri, and G. De Nicolao. Opinion influence and evolution in social networks: A Markovian agents model. *Automatica*, 100:219–230, 2019.

[16] E. Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl 3):7280–7287, 2002.

[17] L. Boninsegna, F. Nüske, and C. Clementi. Sparse learning of stochastic dynamical equations. *The Journal of Chemical Physics*, 148(24):241723, 2018.

[18] F. Bouchet, K. Gawedzki, and C. Nardini. Perturbative calculation of quasi-potential in non-equilibrium diffusions: A mean-field example. *Journal of Statistical Physics*, 163(5):1157–1210, Jun 2016.

[19] G. R. Bowman, V. S. Pande, and F. Noé, editors. *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*, volume 797 of *Advances in Experimental Medicine and Biology*. Springer, 2014.

[20] S. Brunton, J. Proctor, and J. Kutz. Discovering governing equations from data: Sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113:39323937, 09 2015.

[21] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLOS ONE*, 11(2), 2016.

[22] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz. Modern Koopman Theory for Dynamical Systems. 2021.

[23] M. Budišić, R. Mohr, and I. Mezić. Applied Koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4), 2012.

[24] B. Calvez and G. Hutzler. Automatic Tuning of Agent-Based Models Using Genetic Algorithms. pages 41–57, 03 2006.

[25] C. Castellano, S. Fortunato, and V. Loreto. Statistical physics of social dynamics. *Reviews of Modern Physics*, 81:591–646, May 2009.

[26] C. Chen, A. Surana, A. Bloch, and I. Rajapakse. Multilinear Time Invariant System Theory. In *2019 Proceedings of the Conference on Control and its Applications*, pages 118–125. Society for Industrial and Applied Mathematics, jan 2019.

[27] K. K. Chen, J. H. Tu, and C. W. Rowley. Variants of Dynamic Mode Decomposition: Boundary Condition, Koopman, and Fourier Analyses. *Journal of Nonlinear Science*, 22(6):887–915, 2012.

[28] S. Christley, R. Miller Neilan, M. Oremland, R. Salinas, and S. Lenhart. Optimal control of sugarscape agent-based model via a PDE approximation model. *Optimal Control Applications and Methods*, 38(4):473–497, 2017.

[29] P. Clifford and A. Sudbury. A model for spatial conflict. *Biometrika*, 60(3):581–588, 12 1973.

[30] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation Series. Springer, Boston, MA, second edition, 2007.

[31] K. Comer. *Who Goes First? An Examination of the Impact of Activation on Outcome Behavior in Agent-based Models*. PhD thesis, George Mason University, Fairfax, VA, May 2014.

[32] K. Comer and A. Loerch. The Impact of Agent Activation on Population Behavior in an Agent-based Model of Civil Revolt. *Procedia Computer Science*, 20:183–188, 12 2013.

[33] J. Costello. On the number of points in regular discrete simplex (corresp.). *IEEE Transactions on Information Theory*, 17(2):211–212, 1971.

[34] N. Črnjarić-Žic, S. Maćešić, and I. Mezić. Koopman Operator Spectrum for Random Dynamical Systems. *Journal of Nonlinear Science*, 30:2007–2056, 2020.

[35] M. D'Auria, E. O. Scott, R. S. Lather, J. Hilty, and S. Luke. Assisted Parameter and Behavior Calibration in Agent-Based Models with Distributed Optimization. In Y. Demazeau, T. Holvoet, J. M. Corchado, and S. Costantini, editors, *Advances in Practical Applications of Agents, Multi-Agent Systems, and Trustworthiness. The PAAMS Collection*, pages 93–105, Cham, 2020. Springer International Publishing.

[36] M. Dellnitz, G. Froyland, and O. Junge. The Algorithms Behind GAIO — Set Oriented Numerical Methods for Dynamical Systems. In B. Fiedler, editor, *Ergodic Theory, Analysis, and Efficient Simulation of Dynamical Systems*, pages 145–174, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[37] M. Dellnitz, O. Schütze, and T. Hestermeyer. Covering Pareto sets by multilevel subdivision techniques. *Journal of Optimization Theory and Applications*, 124(1):113–136, 2005.

[38] N. Djurdjevac, M. Sarich, and C. Schütte. Estimating the eigenvalue error of Markov state models. *SIAM J. Multiscale Modeling and Simulation*, 10 (1):61–81, 2012.

[39] N. Djurdjevac Conrad, L. Helfmann, J. Zonker, S. Winkelmann, and C. Schütte. Human mobility and innovation spreading in ancient times: a stochastic agent-based simulation approach. *EPJ Data Science*, 7(1):24, 2018.

[40] N. Djurdjevac Conrad, M. Weber, and C. Schütte. Finding dominant structures of nonreversible Markov processes. *Multiscale Modeling and Simulation*, 14(4):1319 – 1340, 2016.

[41] O. Dolezal and H. Tomaskova. An Agent-Based Simulation to Minimize Losses during a Terrorist Attack. *Applied Sciences*, 10(9), 2020.

[42] A. Duncan, S. Liao, T. Vejchodskỳ, R. Erban, and R. Grima. Noise-induced multistability in chemical systems: Discrete versus continuum modeling. *Physical Review E*, 91(4):042111, 2015.

[43] V. Elgart and A. Kamenev. Rare event statistics in reaction-diffusion systems. *Physical Review E*, 70(4):041106, 2004.

[44] R. S. Ellis. Large deviations for a general class of random vectors. *Annals of Probability*, 12.1:1–12, 1984.

[45] J. M. Epstein. Modeling civil violence: An agent-based computational approach. *Generative Social Science: Studies in Agent-Based Computational Modeling*, 99(2):247–270, 2002.

[46] J. M. Epstein and R. Axtell. *Growing Artificial Societies: Social Science from the Bottom Up*. The Brookings Institution, Washington, DC, USA, 1996.

[47] S. N. Ethier and T. G. Kurtz. *Markov processes: characterization and convergence*, volume 282. John Wiley & Sons, 2009.

[48] S. Eubank, H. Guclu, V. S. A. Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang. Modelling disease outbreaks in realistic urban social networks. *Nature*, 429(6988):180–184, 2004.

[49] E. Faou and T. Lelièvre. Conservative stochastic differential equations: Mathematical and numerical analysis. *Mathematics of Computation*, 78(268):2047–2074, 2009.

[50] J. Feng and T. G. Kurtz. *Large Deviations for Stochastic Processes*. AMS, 2006.

[51] M. Fonoberova, V. A. Fonoberov, I. Mezić, J. Mezić, and P. J. Brantingham. Nonlinear dynamics of crime and violence in Urban settings. *JASSS*, 2012.

[52] M. Fonoberova, I. Mezić, J. Mezić, and R. Mohr. An agent-based model of urban insurgence: Effect of gathering sites and Koopman mode analysis. *PLOS ONE*, 13(10), 2018.

[53] M. I. Freidlin, , and A. D. Wentzell. *Random Perturbations of Dynamical Systems*. Grundlehren der mathematischen Wissenschaften. Springer, 1986.

[54] C. P. Fries. Stochastic algorithmic differentiation of (expectations of) discontinuous functions (indicator functions). *International Journal of Computer Mathematics*, 0(0):1–23, 2021.

[55] G. Froyland, G. Gottwald, and A. Hammerlindl. A computational method to extract macroscopic variables and their dynamics in multiscale systems. *SIAM Journal on Applied Dynamical Systems*, 13(4):1816–1846, 2014.

[56] G. Froyland, O. Junge, and P. Koltai. Estimating long term behavior of flows without trajectory integration: The infinitesimal generator approach. *SIAM Journal on Numerical Analysis*, 51(1):223–247, 2013.

[57] D. García-Álvarez. A comparison of a few numerical schemes for the integration of stochastic differential equations in the Stratonovich interpretation. *arXiv e-prints*, 2011.

[58] B. Gaveau, M. Moreau, and J. Toth. Decay of the metastable state in a chemical system: Different predictions between discrete and continuous models. *Letters in Mathematical Physics*, 37(3):285–292, 1996.

[59] P. Gelß, S. Klus, J. Eisert, and C. Schütte. Multidimensional approximation of nonlinear dynamical systems. *Journal of Computational and Nonlinear Dynamics*, 14:061006, 2019.

[60] D. Giannakis. Data-driven spectral decomposition and forecasting of ergodic dynamical systems. *Applied and Computational Harmonic Analysis*, 47(2):338–396, 2019.

[61] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of computational physics*, 22(4):403–434, 1976.

[62] D. T. Gillespie. A rigorous derivation of the chemical master equation. *Physica A: Statistical Mechanics and its Applications*, 188(1-3):404–425, 1992.

[63] D. T. Gillespie. The chemical langevin equation. *The Journal of Chemical Physics*, 113(1):297–306, 2000.

[64] R. Grima, P. Thomas, and A. V. Straube. How accurate are the nonlinear chemical fokker-planck and chemical langevin equations? *The Journal of chemical physics*, 135(8):084103, 2011.

[65] V. Grimm, S. F. Railsback, C. E. Vincenot, U. Berger, C. Gallagher, D. L. DeAngelis, B. Edmonds, J. Ge, J. Giske, J. Groeneveld, A. S. A. Johnston, A. Milles, J. Nabe-Nielsen, J. G. Polhill, V. Radchuk, M.-S. Rohwäder, R. A. Stillman, J. C. Thiele, and D. Ayllón. The ODD Protocol for Describing Agent-Based and Other Simulation Models: A Second Update to Improve Clarity, Replication, and Structural Realism. *Journal of Artificial Societies and Social Simulation*, 23(2), 2020.

[66] M. Hallier. *Formalization and Metastability Analysis of Agent-Based Evolutionary Models*. PhD thesis, Freie Universität Berlin, 2014.

[67] P. Hanggi, H. Grabert, P. Talkner, and H. Thomas. Bistable systems: Master equation versus fokker-planck modeling. *Physical Review A*, 29(1):371, 1984.

[68] L. Helfmann, N. Djurdjevac Conrad, A. Djurdjevac, S. Winkelmann, and C. Schütte. From interacting agents to density-based modeling with stochastic PDEs. *Communications in Applied Mathematics and Computational Science*, 16(1):1 – 32, 2021.

[69] F. Herreriás-Azcué and T. Galla. Consensus and diversity in multistate noisy voter models. *Physical Review E*, 100(2), 2019.

[70] J. Hogg, M. Fonoberova, I. Mezić, and R. Mohr. Koopman mode analysis of agent-based models of logistics processes. *PLOS ONE*, 14(9), 2019.

[71] R. A. Holley and T. M. Liggett. Ergodic theorems for weakly interacting infinite systems and the voter model. *The Annals of Probability*, 3(4):643–663, 1975.

[72] B. J. Hollingsworth. *Stochastic differential equations: A dynamical systems approach*. PhD thesis, Auburn University, 2008.

[73] W. Huisinga, S. Meyn, and C. Schütte. Phase transitions and metastability in Markovian and molecular systems. *Ann. Appl. Probab.*, 14(1):419–458, 02 2004.

[74] A. Jędrzejewski and K. Sznajd-Weron. Statistical physics of opinion formation: is it a spoof? *Comptes Rendus Physique*, 20(4):244–261, 2019.

[75] E. Kaiser, J. N. Kutz, and S. L. Brunton. Discovering conservation laws from data for control. In *Proceedings of the IEEE Conference on Decision and Control*, 2019.

[76] E. Kaiser, J. N. Kutz, and S. L. Brunton. Data-driven discovery of Koopman eigenfunctions for control. *Machine Learning: Science and Technology*, 2021.

[77] J. Kazil, D. Masad, and A. Crooks. Utilizing Python for Agent-Based Modeling: The Mesa Framework. In R. Thomson, H. Bisgin, C. Dancy, A. Hyder, and M. Hussain, editors, *Social, Cultural, and Behavioral Modeling*, pages 308–317, Cham, 2020. Springer International Publishing.

[78] I. G. Kevrekidis, C. W. Gear, and G. Hummer. Equation-free: The computer-aided analysis of complex multiscale systems. *AIChE J.*, 2004.

[79] I. G. Kevrekidis, C. W. Gear, J. Hyman, P. G. Kevrekidid, O. Runborg, and C. Theodoropoulos. Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis. *Communications in Mathematical Sciences*, 1:715–762, 01 2003.

[80] E. Kiesling, M. Günther, C. Stummer, and L. M. Wakolbinger. Agent-based simulation of innovation diffusion: A review. *Central European Journal of Operations Research*, 20(2):183–230, 2012.

[81] S. Klus, P. Gelß, S. Peitz, and C. Schütte. Tensor-based dynamic mode decomposition. *Nonlinearity*, 31(7), 2018.

[82] S. Klus, P. Koltai, and C. Schütte. On the numerical approximation of the Perron–Frobenius and Koopman operator. *Journal of Computational Dynamics*, 3(1):51–79, 2016.

[83] S. Klus, F. Nüske, and B. Hamzi. Kernel-Based Approximation of the Koopman Generator and Schrödinger Operator. *Entropy*, 22(7), 2020.

[84] S. Klus, F. Nüske, P. Koltai, H. Wu, I. Kevrekidis, C. Schütte, and F. Noé. Data-driven model reduction and transfer operator approximation. *Journal of Nonlinear Science*, 28(3):985 – 1010, 2018.

[85] S. Klus, F. Nüske, S. Peitz, J.-H. Niemann, C. Clementi, and C. Schütte. Data-driven approximation of the Koopman generator: Model reduction, system identification, and control. *Physica D: Nonlinear Phenomena*, 406:132416, 2020.

[86] S. Klus, I. Schuster, and K. Muandet. Eigendecompositions of transfer operators in reproducing kernel Hilbert spaces. *Journal of Nonlinear Science*, 2019.

[87] B. Koopman. Hamiltonian systems and transformations in Hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315, 1931.

[88] M. Korda and I. Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149 – 160, 2018.

[89] M. Korda and I. Mezić. Optimal Construction of Koopman Eigenfunctions for Prediction and Control. *IEEE Transactions on Automatic Control*, 65(12):5114–5129, 2020.

[90] R. Korkmaz Tan and Ş. Bora. Adaptive parameter tuning for agent-based modeling and simulation. *SIMULATION*, 95(9):771–796, 2019.

[91] T. G. Kurtz. Solutions of ordinary differential equations as limits of pure jump Markov processes. *Journal of Applied Probability*, 7(1):49–58, 1970.

[92] T. G. Kurtz. The relationship between stochastic and deterministic models for chemical reactions. *The Journal of Chemical Physics*, 57(7):2976–2978, 1972.

[93] T. G. Kurtz. Limit theorems and diffusion approximations for density dependent Markov chains. In *Stochastic Systems: Modeling, Identification and Optimization, I*, pages 67–78. Springer, 1976.

[94] T. G. Kurtz. Strong approximation theorems for density dependent Markov chains. *Stochastic Processes and their Applications*, 6(3):223–240, 1978.

[95] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM, Philadelphia, 2016.

[96] J. Källström and F. Heintz. Tunable Dynamics in Agent-Based Simulation using Multi-Objective Reinforcement Learning. In *Adaptive and Learning Agents Workshop (ALA-19) at AAMAS*, Montreal, Canada, May 2019.

[97] F. Lamperti, A. Roventini, and A. Sani. Agent-based model calibration using machine learning surrogates. *Journal of Economic Dynamics and Control*, 90:366–389, 2018.

[98] A. Lasota and M. C. Mackey. *Chaos, fractals, and noise: Stochastic aspects of dynamics*, volume 97 of *Applied Mathematical Sciences*. Springer, New York, 2 edition, 1994.

[99] T. Lekvam, B. Gambäck, and L. Bungum. Agent-based modeling of language evolution. In *Proceedings of the 5th Workshop on Cognitive Aspects of Computational Language Learning (CogACLL)*, pages 49–54, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

[100] C. Lemos, R. J. Lopes, and H. Coelho. Analysis of the Decision Rule in Epstein's Agent-Based Model of Civil Violence. In *2015 Third World Conference on Complex Systems (WCCS)*, pages 1–6, 2015.

[101] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(10):103111, 2017.

[102] P. Liu, H. R. Safford, I. D. Couzin, and I. G. Kevrekidis. Coarse-grained variables for particle-based models: diffusion maps and animal swarming simulations. *Computational Particle Mechanics*, 1(4):425–440, 2014.

[103] P. Liu, C. I. Siettos, C. W. Gear, and I. G. Kevrekidis. Equation-free model reduction in agent-based computations: Coarse-grained bifurcation and variable-free rare event analysis. *Mathematical Modelling of Natural Phenomena*, 10(3):71–90, 2015.

[104] Z. Liu, D. Rexachs, F. Epelde, and E. Luque. A simulation and optimization based method for calibrating agent-based emergency department models under data scarcity. *Computers & Industrial Engineering*, 103:300–309, 2017.

[105] F. Lu, M. Zhong, S. Tang, and M. Maggioni. Nonparametric inference of interaction laws in systems of agents from trajectory data. *Proceedings of the National Academy of Sciences of the United States of America*, 116(29):14424–14433, 2019.

[106] M. Lücke and F. Nüske. tgEDMD: Approximation of the Kolmogorov Operator in Tensor Train Format. 2021.

[107] B. Lusch, J. N. Kutz, and S. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9, 2017.

[108] A. Mardt, L. Pasquali, H. Wu, and F. Noé. VAMPnets for deep learning of molecular kinetics. *Nature Communications*, 9, 2018.

[109] A. Mauroy and J. Goncalves. Linear identification of nonlinear systems: A lifting technique based on the Koopman operator. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 6500–6505, 2016.

[110] A. Mauroy and J. Goncalves. Koopman-Based Lifting Techniques for Nonlinear Systems Identification. *IEEE Transactions on Automatic Control*, 2019.

[111] A. J. McKane and T. J. Newman. Stochastic models in population biology and their deterministic analogs. *Phys. Rev. E*, 70:041902, Oct 2004.

[112] A. J. McKane and T. J. Newman. Predator-prey cycles from resonant amplification of demographic stochasticity. *Physical review letters*, 94(21):218102, 2005.

[113] D. A. Messenger and D. M. Bortz. Weak SINDy for partial differential equations. *Journal of Computational Physics*, 443:110525, 2021.

[114] P. Metzner. *Transition path theory for Markov processes: Application to molecular dynamics*. PhD thesis, Freie Universität Berlin, 2007.

[115] I. Mezić. Analysis of Fluid Flows via Spectral Properties of the Koopman Operator. *Annual Review of Fluid Mechanics*, 45(1):357–378, 2013.

[116] I. Mezić. Spectrum of the Koopman Operator, Spectral Expansions in Functional Spaces, and State-Space Geometry. *Journal of Nonlinear Science*, 30(5):2091–2145, 2020.

[117] J. Mielke and A. Geiges. Model-Stakeholder Interactions for a Sustainable Mobility Transition. *Global Climate Forum Berlin*, 2018.

[118] K. Miettinen. *Nonlinear Multiobjective Optimization*. International Series in Operations Research & Management Science. Springer, Boston, MA, 1998.

[119] T. Misawa. Conserved Quantities and Symmetries Related to Stochastic Dynamical Systems. *Annals of the Institute of Statistical Mathematics*, 51:779–802, 02 1999.

[120] A. K. Misra. A simple mathematical model for the spread of two political parties. *Nonlinear Analysis: Modelling and Control*, 17(3):343–354, 2012.

[121] A. Montefusco. *Dynamic Coarse-Graining via Large-Deviation Theory*. PhD thesis, ETH Zürich, 2019.

[122] I. Moya, M. Chica, and Ó. Cordón. A multicriteria integral framework for agent-based model calibration using evolutionary multiobjective optimization and network-based visualization. *Decision Support Systems*, 124:113111, 2019.

[123] V. Mysore, G. Narzisi, and B. Mishra. Agent modeling of a sarin attack in manhattan. *In Proceedings of the First International Workshop on Agent Technology for Disaster Management, ATDM*, pages 108–115, 2006.

[124] J. T. Nardini, R. E. Baker, M. J. Simpson, and K. B. Flores. Learning differential equation models from stochastic agent-based model simulations. *Journal of The Royal Society Interface*, 18(176), 2021.

[125] G. Narzisi, V. Mysore, and B. Mishra. Multi-objective evolutionary optimization of agent-based models: An application to emergency response planning. *Proceedings of the 2nd IASTED International Conference on Computational Intelligence, CI 2006*, pages 224–230, 2006.

[126] J.-H. Niemann, S. Klus, and C. Schütte. Data-driven model reduction of agent-based systems using the Koopman generator. *PLOS ONE*, 16(5):1–23, 05 2021.

[127] J.-H. Niemann, S. Winkelmann, S. Wolf, and C. Schütte. Agent-based modeling: Population limits and large timescales. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(3):17, 2021.

[128] J. Ohkubo, N. Shnerb, and D. A. Kessler. Transition Phenomena Induced by Internal Noise and Quasi-Absorbing State. *Journal of the Physical Society of Japan*, 77(4):044002, 2008.

[129] G. H. Orcutt. A New Type of Socio-Economic System. *The Review of Economics and Statistics*, 39(2):116–123, May 1957.

[130] M. Oremland and R. Laubenbacher. Using difference equations to find optimal tax structures on the SugarScape. *Journal of Economic Interaction and Coordination*, 2014.

[131] G. A. Pavliotis. *Stochastic Processes and Applications: Diffusion Processes, the Fokker–Planck and Langevin Equations*, volume 60 of *Texts in Applied Mathematics*. Springer, New York, 2014.

[132] S. Peitz. *Exploiting structure in multiobjective optimization and optimal control*. PhD thesis, Universität Paderborn, 2017.

[133] S. Peitz and M. Dellnitz. A Survey of Recent Trends in Multiobjective Optimal Control – Surrogate Models, Feedback Control and Objective Reduction. *Mathematical and Computational Applications*, 23(2), 2018.

[134] S. Peitz and S. Klus. Koopman operator-based model reduction for switched-system control of PDEs. *Automatica*, 106:184 – 191, 2019.

[135] S. Peitz and S. Klus. *Feedback Control of Nonlinear PDEs Using Data-Efficient Reduced Order Models Based on the Koopman Operator*, pages 257–282. Springer International Publishing, Cham, 2020.

[136] S. Peitz, S. E. Otto, and C. W. Rowley. Data-driven model predictive control using interpolated Koopman generators. *SIAM Journal on Applied Dynamical Systems*, 19(3):2162–2193, 2020.

[137] D. Platt. A comparison of economic agent-based model calibration methods. *Journal of Economic Dynamics and Control*, 113:103859, 2020.

[138] J. Párraga-Álava, G. M. Garzón, and R. V. Valarezo. Multi-objective genetic algorithms: are they useful for tuning parameters in agent-based simulation? *Revista Ibérica de Sistemas e Tecnologias de Informação*, 19:172–184, 2019.

[139] M. N. Read, K. Alden, L. M. Rose, and J. Timmis. Automated multi-objective calibration of biological agent-based simulations. *Journal of The Royal Society Interface*, 13(122):20160543, 2016.

[140] S. Redner. Reality-inspired voter models: A mini-review. *Comptes Rendus Physique*, 20(4), 2019.

[141] A. N. Riseth and J. P. Taylor-King. Operator fitting for parameter estimation of stochastic differential equations. *ArXiv e-prints*, 2017.

[142] H. Risken and T. Frank. *The Fokker–Planck Equation: Methods of Solutions and Applications*. Springer Series in Synergetics. Springer-Verlag Berlin Heidelberg, 1996.

[143] S. Röblitz and M. Weber. Fuzzy spectral clustering by PCCA+: application to Markov state models and data classification. *Advances in Data Analysis and Classification*, 7(2):147 – 179, 2013.

[144] A. Rogers and P. Tessin. Multi-objective calibration for agent-based models. 01 2004.

[145] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115127, 2009.

[146] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3, April 2017.

[147] M. Sarich, F. Noé, and C. Schütte. On the approximation quality of Markov state models. *SIAM J. Multiscale Modeling and Simulation*, 8 (4):1154–1177, 2010.

[148] M. Sarich and C. Schütte. Approximating selected non-dominant timescales by Markov state models. *Comm. Math. Sci.*, 10(3):1001 – 1013, 2012.

[149] T. C. Schelling. Dynamic models of segregation. *The Journal of Mathematical Sociology*, 1(2):143–186, 1971.

[150] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.

[151] C. Schütte, W. Huisinga, and P. Deuflhard. Transfer Operator Approach to Conformational Dynamics in Biomolecular Systems. In B. Fiedler, editor, *Ergodic Theory, Analysis, and Efficient Simulation of Dynamical Systems*, pages 191–223, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[152] C. Schütte, F. Noé, J. Lu, M. Sarich, and E. Vanden-Eijnden. Markov state models based on milestoning. *Journal of Chemical Physics*, 134(20):204105, 2011.

[153] C. Schütte and M. Sarich. *Metastability and Markov State Models in Molecular Dynamics: Modeling, Analysis, Algorithmic Approaches*. Courant Lecture Notes No. 32. American Mathematical Society, 2014.

[154] C. Schütte and M. Sarich. A critical appraisal of Markov state models. *The European Physical Journal Special Topics*, 224(12):2445–2462, 2015.

[155] O. Schütze, A. Dell'Aere, and M. Dellnitz. On Continuation Methods for the Numerical Treatment of Multi-Objective Optimization Problems. *Dagstuhl Seminar Proceedings*, 2005.

[156] O. Schütze, K. Witting, S. Ober-Blöbaum, and M. Dellnitz. *Set Oriented Methods for the Numerical Treatment of Multiobjective Optimization Problems*, pages 187–219. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[157] C. R. Schwantes and V. S. Pande. Modeling molecular kinetics with tICA and the kernel trick. *Journal of Chemical Theory and Computation*, 11(2):600–608, 2015.

[158] A. Sîrbu, V. Loreto, V. D. P. Servedio, and F. Tria. *Opinion Dynamics: Models, Extensions and External Effects*, pages 363–401. Springer, 05 2017.

[159] L. Steels. Agent-based models for the emergence and evolution of grammar. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 371(1701):20150447, 2016.

[160] R. L. Stratonovich. A new representation for stochastic integrals and equations. *SIAM Journal on Control*, 4(2):362–371, 1966.

[161] P. Thomas, A. V. Straube, J. Timmer, C. Fleck, and R. Grima. Signatures of nonlinearity in single cell noise-induced oscillations. *Journal of theoretical biology*, 335:222–234, 2013.

[162] A. C. Tsoumanis, C. I. Siettos, I. G. Kevrekidis, and G. V. Bafas. Equation-Free Multiscale Computations in Social Networks: from Agent-based Modelling to Coarse-grained Stability and Bifurcation Analysis. *International Journal of Bifurcation and Chaos*, 20(11):3673–3688, 2010.

[163] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2), 2014.

[164] S. M. Ulam. *A Collection of Mathematical Problems*. Interscience Publisher NY, 1960.

[165] N. G. Van Kampen. *Stochastic Processes in Physics and Chemistry*. North-Holland Personal Library. Elsevier, Amsterdam, third edition, 2007.

[166] S. Venkatramanan, B. Lewis, J. Chen, D. Higdon, A. Vullikanti, and M. V. Marathe. Using data-driven agent-based models for forecasting emerging infectious diseases. *Epidemics*, 22:43 – 49, 2018. The RAPIDD Ebola Forecasting Challenge.

[167] J. von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, Princeton, USA, 1966.

[168] T. Warnke, O. Reinhardt, A. Klabunde, F. Willekens, and A. M. Uhrmacher. Modelling and simulating decision processes of linked lives: An approachbased on concurrent processes and stochastic race. *Population Studies*, 71:sup1:69–83, 2017.

[169] C. Weimer, J. O. Miller, R. Hill, and D. Hodson. Agent Scheduling in Opinion Dynamics: A Taxonomy and Comparison Using Generalized Models. *Journal of Artificial Societies and Social Simulation*, 22(4):5, 2019.

[170] U. Wilensky. Netlogo. Technical report, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999.

[171] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[172] S. Winkelmann and C. Schütte. Hybrid models for chemical reaction networks: Multiscale theory and application to gene regulatory systems. *The Journal of Chemical Physics*, 147(11):114115, 2017.

[173] S. Winkelmann and C. Schütte. *Stochastic Dynamics in Computational Biology*, volume 8. Springer International Publishing, 2020.

[174] H. Wulkow, T. O. F. Conrad, N. Djurdjevac Conrad, S. A. Müller, K. Nagel, and C. Schütte. Prediction of Covid-19 spreading and optimal coordination of countermeasures: From microscopic to macroscopic models to Pareto fronts. *PLOS ONE*, 16(4):1–29, 04 2021.

[175] W. Zhang, C. Hartmann, and C. Schütte. Effective dynamics along given reaction coordinates, and reaction rate theory. *Faraday Discussions*, 195:365–394, 2016.

[176] W. Zhou, L. Zhang, J. Hong, and S. Song. Projection methods for stochastic differential equations with conserved quantities. *BIT Numerical Mathematics*, 56(4):1497–1518, 2016.

[177] Y. Zou, V. A. Fonoberov, Maria F., I. Mezic, and I. G. Kevrekidis. Model reduction for agent-based social simulation: Coarse-graining a civil violence model. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 85(6), 2012.