

# **DATA-ASSISTED MODELING OF COMPLEX CHEMICAL AND BIOLOGICAL SYSTEMS**

by

**Georgios Psarellis**

**A dissertation submitted to Johns Hopkins University  
in conformity with the requirements for the degree of  
Doctor of Philosophy**

**Baltimore, Maryland**

**September, 2022**

**© 2022 Georgios Psarellis**

**All rights reserved**

# Abstract

Complex systems are abundant in chemistry and biology; they can be multiscale, possibly high-dimensional or stochastic, with nonlinear dynamics and interacting components. It is often nontrivial (and sometimes impossible), to determine and study the macroscopic quantities of interest and the equations they obey. One can only (judiciously or randomly) probe the system, gather observations and study trends. In this thesis, Machine Learning is used as a *complement* to traditional modeling and numerical methods to enable data-assisted (or data-driven) dynamical systems. As case studies, three complex systems are sourced from diverse fields: The first one is a high-dimensional computational neuroscience model of the Suprachiasmatic Nucleus of the human brain, where bifurcation analysis is performed by simply probing the system. Then, manifold learning is employed to discover a latent space of neuronal heterogeneity. Second, Machine Learning surrogate models are used to optimize dynamically operated catalytic reactors. An algorithmic pipeline is presented through which it is possible to program catalysts with active learning. Third, Machine Learning is employed to extract laws of Partial Differential Equations describing bacterial Chemotaxis. It is demonstrated how Machine Learning manages to capture the rules of bacterial motility in the macroscopic level, starting from diverse data sources (including real-world experimental data). More importantly, a framework is constructed through which already existing, partial

knowledge of the system can be exploited. These applications showcase how Machine Learning can be used *synergistically* with traditional simulations in different scenarios: (i) Equations are available but the overall system is so high-dimensional that efficiency and explainability suffer, (ii) Equations are available but lead to highly nonlinear black-box responses, (iii) Only data are available (of varying source and quality) and equations need to be discovered. For such data-assisted dynamical systems, we can perform fundamental tasks, such as integration, steady-state location, continuation and optimization. This work aims to unify traditional scientific computing and Machine Learning, in an efficient, data-economical, generalizable way, where *both the physical system and the algorithm matter*.

**Primary Reader and advisor:** Prof. Ioannis G. Kevrekidis

**Secondary Readers:** Prof. Michael Tsapatsis, Prof. Michael D. Shields

# Acknowledgments

Similarly to the subject of my research, my PhD journey has often been complex, nonlinear and with many “moving parts”. I would like to thank the people who helped me along that journey, who contributed in my (academic and personal) growth, and impacted me in various ways.

First and foremost, I want to thank my advisor, Prof. Ioannis Kevrekidis, who provided me with his guidance and intuition on the projects I’ve worked on. Even when the PhD work seemed sisyphian at times, he helped me stay motivated and focus on the bigger picture. Through his advisorship, I redefined what it means to be an engineer in the modern data-abundant world: having a fundamental understanding of algorithms and science that transcends specific fields while remaining grounded on real world applications and impact.

I am also grateful to the postdoctoral researchers for the helpful discussions, invaluable assistance and resources they provided me: Dr. Hassan Arbabi, Dr. Juan Bello-Rivas, Dr. Tom Bertalan, Dr. Felix Dietrich, Dr. Felix Kemeth, Dr. Mahdi Kooshkbaghi and Dr. Seungjoon Lee. I’m also indebted to Prof. Andreas Boudouvis, Dr. Eleni Koronaki, Prof. Mihalis Kavousanakis and Prof. Constantinos Siettos for their invaluable help and advice.

I also want to thank the members of my defense committee: Prof. Tsapatsis, Prof. Clancy, Prof. Bevan, Prof. Maggioni, Prof. Shields and Prof. Villar. I am also thankful for the financial support from the Onassis Foundation throughout my PhD years.

Finally, I want to thank my family, my parents Elli and Markos and my brother Antonis for their unconditional support, as well as all my friends (both new and old) who supported me, shared my concerns and celebrated my successes along my PhD journey.

“[One] cannot search for what he knows – since he knows it, there is no need to search – nor for what he does not know, for he does not know what to look for.”

Plato, Meno, 80 d-e (G.M.A. Grube translation)

# Table of Contents

<b>Abstract</b>	ii
<b>Acknowledgments</b>	iv
<b>Table of Contents</b>	vii
<b>List of Tables</b>	x
<b>List of Figures</b>	xi
<b>List of Algorithms</b>	xvii
<b>1 Introduction</b>	<b>1</b>
1.1 Periodically Forced Systems . . . . .	3
1.1.1 Systems with autonomous oscillations . . . . .	3
1.1.2 Systems without autonomous oscillations . . . . .	6
1.2 Multiscale Systems . . . . .	7
1.3 High-dimensional Systems and Networks . . . . .	9
References . . . . .	11
<b>2 Algorithms</b>	<b>17</b>
2.1 Numerical Methods . . . . .	18
2.1.1 Solvers for Periodic Steady States . . . . .	18
2.1.1.1 Variational approach . . . . .	18
2.1.1.2 Newton-Krylov GMRES . . . . .	19
2.1.2 Numerical Continuation . . . . .	20
2.2 Machine Learning . . . . .	23
2.2.1 Supervised Learning . . . . .	24
2.2.1.1 Gaussian Process Regression . . . . .	24
2.2.1.2 Artificial Neural Networks . . . . .	27
2.2.2 Unsupervised Learning . . . . .	27
2.2.2.1 Diffusion Maps . . . . .	28
2.2.3 Active Learning . . . . .	29

2.2.3.1	Bayesian Optimization . . . . .	29
2.2.3.2	Bayesian Continuation . . . . .	32
References	. . . . .	35
<b>3</b>	<b>Discovering Limits of Entrainment for Circadian Neuronal Networks</b>	<b>38</b>
3.1	Motivation . . . . .	38
3.2	Computational Model . . . . .	39
3.3	Single neuron bifurcation studies . . . . .	41
3.4	Heterogeneous network bifurcation studies . . . . .	48
3.5	Latent heterogeneity space . . . . .	53
3.6	Discussion and Conclusions . . . . .	54
References	. . . . .	56
<b>4</b>	<b>Optimizing Reactors under Dynamic Operation</b>	<b>60</b>
4.1	Motivation . . . . .	60
4.2	Case studies . . . . .	62
4.3	Discovering periodic steady states . . . . .	70
4.3.1	Solver efficiency . . . . .	70
4.3.2	Newton-Krylov GMRES . . . . .	72
4.3.3	Solving for conversion constraints . . . . .	73
4.4	Understanding the effect of the forcing parameters . . . . .	73
4.5	Continuation . . . . .	78
4.6	Step forcing function optimization . . . . .	81
4.7	Radial Basis Function optimization . . . . .	83
4.8	Discussion and Conclusions . . . . .	88
References	. . . . .	89
<b>5</b>	<b>Learning Chemotactic PDEs with Machine Learning</b>	<b>93</b>
5.1	PDE simulations . . . . .	98
5.1.1	PDE model . . . . .	98
5.1.2	Data driven PDEs . . . . .	99
5.1.3	Results . . . . .	100
5.1.4	Discussion and Conclusions . . . . .	115
5.2	Agent based modeling . . . . .	118
5.2.1	ABM model . . . . .	118
5.2.2	Data driven PDEs . . . . .	120
5.2.3	Results . . . . .	124
5.2.4	Discussion and Conclusions . . . . .	129
5.3	Experiments . . . . .	130
5.3.1	Experiment description . . . . .	130
5.3.2	Data driven PDE . . . . .	130



5.3.3	Results . . . . .	132
5.3.4	Discussion and Conclusions . . . . .	133
	References . . . . .	134
<b>6</b>	<b>Discussion and Conclusion</b>	<b>142</b>
	References . . . . .	144

# List of Tables

4.1	Parameters used to simulate the system of Eqs.4.1, 4.2, depicted in Fig. 4.1. Note, that “BEP” stands for Brønsted-Evans-Polyani principle (Akhade et al., 2018). . . . .	64
4.2	Parameters used to simulate the system of Eqs.4.3 - 4.8 depicted in Fig. 4.3. Note, that “BE” stands for Binding Energies, while “BEP” stands for Brønsted-Evans-Polyani principle (Akhade et al., 2018) . . . . .	68
4.3	Performance comparison between brute-force integration and Newton’s method (with the variational approach), for the cases shown in Fig. 4.5. . . . .	71
4.4	Static steady states and their eigenvalues. . . . .	75
5.1	Parameters used for the PDE simulation and a schematic representation of the domain. . . . .	101
5.2	Listing of the data-driven models explored. Notation: $f, h$ denote surrogate functions for the entire RHS of the $b$ - and $c$ -PDE respectively, while $g$ denotes a surrogate for the chemotactic term. Subscripts “GPR” and “NN” denote Gaussian Process Regression and Artificial Neural Network respectively. $\Delta t$ denotes the time delay used in all models with partial information. . . . .	102
5.3	Summary of all data-driven models for C-learning. . . . .	112
5.4	Selected groups of input features and the corresponding predicted quantity (output) for different data-driven PDE law correction approaches. “Reduced” represents models constructed using the (fewer) input features selected via ARD within the Gaussian process framework. We only reduce GP models through ARD; the corresponding NN reduction was not attempted. . . . .	124

# List of Figures

1.1	Bifurcation diagram of the Brusselator w.r.t the relative forcing angular frequency $\omega/\omega_0$ and forcing amplitude $a$ . Interested readers are referred to the original manuscript (Tomita and Kai, 1978) for explanation of numbered/indexed points on the diagram. . . . .	5
1.2	Hierarchy of points/loci of points for the periodically forced system of the Brusselator. Following a top-down approach: (left) 1:1 resonance horn. The blue region is where synchronization is possible, while the gray region denotes quasiperiodicity. Every point in the blue region (boundary inclusive) corresponds to (at least one) limit cycle. Two sample bifurcations at $a = 0.001$ are denoted with red stars. (middle) projection of the isola for $a = 0.001$ . The red stars in the left subfigure are now the bilateral limits of the isola where the stable (blue) and unstable (red) branches coalesce. Every point of the isola is a limit cycle (or, equivalently, a stroboscopic map fixed point). Two points are sampled for $\omega/\omega_0 = 1$ . (right) limit cycles (blue-stable, red-unstable) for the two points selected in the middle subfigure. . . . .	6
1.3	Hierarchy of multiple scales in fluid mechanics along with suitable laws for each scale (Weinan and Engquist, 2003). . . . .	7
1.4	Hierarchy of multiple scales in biology along with examples of data that can be collected at each scale ( <i>Multiscale Systems Biology and Modeling</i> , <i>howpublished = https://researcher.watson.ibm.com/researcher/view_group.php?id=5372</i> , <i>note = Accessed: 2022-07-09 n.d.</i> ). . . . .	8
2.1	Pseudo-arclength continuation iteration for the steady states of $dx/dt = f(x; \lambda) := \lambda + x^2$ : Given the direction vector (here, colored blue) the predictor for the next solution on the branch can be found (colored red). Solving the augmented system of equations (such as Eq. 2.9, 2.10) will restrict the next solution (colored green) on a plane perpendicular to the direction vector. . . . .	22
2.2	An example of Gaussian Process Regression (GPR), here learning from 10 randomly sampled points of the ground truth function $f(x) = x \cos(x)$ . The kernel used is Gaussian and the assumed additive noise variance $\sigma^2 = 10^{-3}$ (the effect of which can be better seen at the inset). GPR provides not only a prediction, but quantifies uncertainty around it (here, the 95% confidence interval is shown). In fact, GPR returns the best fitted posterior <i>distribution</i> of output values in the input space. As expected, the uncertainty grows where observations are sparse, especially at the boundaries of the input space. . . . .	26

2.3	En example of several Bayesian Optimization iterations borrowed from (Brochu, Cora, and Freitas, 2010). Consecutive iterates are indexed by $t$ . At the top row, the ground truth objective function is shown (dashed red line), along with the GP predictive mean (blue line) and its uncertainty (represented by the shaded region). Current sample is denoted by red points, and the <i>last</i> point discovered as a white point. At the bottom row, the UCB acquisition function is shown (the maximizing counterpart of LCB). At each iterate, the next suggested point, will be the one maximizing the acquisition function (maxima marked by red triangles). . . . .	31
2.4	En example of a Bayesian Continuation instance for one-parameter continuation of roots of $f(x; r) = r + x^2$ . Here, a Gaussian Process fit is performed <i>locally</i> on the last few points on the solution curves. This fit provides predictors (in the form of GP predictive means) of where the next solution point might lie. As can be seen in the inset, the predictor is close enough to the actual solution (corrector). This method can be adaptive, exploiting uncertainty estimates of the local fit (like the 95% confidence interval shown here). Automatic Relevance Determination (see Subchapter 2.2.1.1) can also be exploited to overcome turning points. . . . .	34
3.1	(left) GABA adjacency matrix, (middle) VIP adjacency matrix, (right) histograms sampled from different $v_{sP_0}$ considered here. . . . .	41
3.2	Timeseries of steady state oscillations of the (left) unforced system with natural period $T_0 = 22.04h$ , and (right) forced system with $T_f = 24h$ . . . . .	42
3.3	Bifurcation diagram for periodic steady states under photic stimulus for a single circadian neuron. Here, the <i>MP</i> ( <i>Per</i> mRNA) projection is shown. For reference $\omega_f = 0.262rad/h$ for $T_f = 24h$ . . . . .	42
3.4	Unforced <i>MP</i> oscillations at the periodic steady state of $k_1 = 0.2$ and $k_1 = 0.8$ . . . . .	44
3.5	Bifurcation diagram for periodic steady states under photic stimulus for varying angular frequency $\omega_f$ and Longdaysin effect ( $k_1$ value) in the case of a single circadian neuron. At the bottom subfigure, a collection of isolas is shown for discrete values of $k_1$ . At the top figure, the <i>resonance horn</i> is approximated in the $\omega_f - k_1$ space. . . . .	45
3.6	Loss of entrainment for $k_1 = 0.05, \omega_f = 0.16746rad/h$ . Near the <i>tip</i> of the resonance horn (low $k_1$ ), quasiperiodicity is observed outside, but close to the entrainment limits. This can be confirmed by plotting iterates of the stroboscopic map (here, shown in the <i>MC - MP</i> projection), where an invariant circle is observed (left). In the phase portrait representation (right), trajectories are attracted to a torus (here, the <i>MP - PC<sub>C</sub> - PC</i> projection is shown, along with stroboscopic map iterates). Note that <i>MP</i> : <i>Per</i> mRNA, <i>MB</i> : <i>Bmal1</i> mRNA, <i>MC</i> : <i>Cry</i> mRNA, <i>PC</i> : nonphosphorylated <i>Cry</i> protein in the cytosol, <i>PC<sub>C</sub></i> : nonphosphorylated <i>Per-Cry</i> protein complex in the cytosol. . . . .	46
3.7	Loss of entrainment for $k_1 = 0.49 : \omega_f = 0.2446rad/h$ . (left panel) $\omega_f = 0.3639rad/h$ (right panel). Stroboscopic map iterates reveal frequency-locking and chaos respectively (here in the <i>MC - MP</i> projection) . . . . .	46
3.8	Bifurcation diagram for periodic steady states w.r.t the duty cycle $\phi$ under photic stimulus of fixed angular frequency $\omega_f = 0.268rad/h$ and without drug intervention ( $k_1 = 0.49$ ) in the case of a single circadian neuron. Notice the additional bifurcations (leading to complex dynamics) towards lower day/night ratios. . . . .	47
3.9	Phase portraits at low $\phi$ values where bistability as well as chaos is observed. . . . .	48

3.11	2D projections of the 8925–dimensional limit cycle for an even higher value of heterogeneity variance studied here ( $\sigma^2 = 0.04$ ), for comparison purposes. This limit cycle was calculated for $k_1 = 0.49, \phi = 0.5$ with forcing period $T_f = 24h$ . This figure should be qualitatively compared to Figs. 3.10e, 3.10f. . . . .	49
3.10	2D projections of the 8925–dimensional limit cycles, for $k_1$ in $\{0.2, 0.35, 0.49$ (nominal value), $0.65, 0.8\}$ , for heterogeneity variance $10^{-3}$ or $10^{-4}$ , with forcing angular frequency equal to the intrinsic one (for each $k_1$ ) and for $\phi = 0.5$ . . . . .	51
3.12	Superimposed resonance horns for the case of a single circadian neuron, and a neuronal networks with varying extents of heterogeneity. . . . .	52
3.13	Loss of entrainment due to the emergence of a single “rogue” oscillator. Three panels show oscillation “snapshots” along a long trajectory. Initially (left panel), all neurons seem to be oscillating in synchrony. After some time (middle panel), the trajectory of one neuron slowly diverges from the rest, and for longer times (right panel) it starts oscillating erratically. Variable $MP_i$ is reported for every neuron, $i$ while each neuron is colored by its heterogeneity $v_{sP_0}$ value; the rogue oscillator has the highest. Forcing parameters and heterogeneity variance value are mentioned in the text. . . . .	52
3.14	Continuation branch of periodic steady states solutions for the neuronal network w.r.t. the duty cycle $\phi$ for heterogeneity variance of value $\sigma^2 = 10^{-3}$ and $k_1 = 0.49, T_f = 24h$ . Here the variable $MP_i$ is reported for every neuron $i$ at the initial phase of the forcing. This should be compared to the lower branch in Fig. 3.8. . . . .	53
3.15	(left) Data from a single phase of the limit cycle plotted on the space defined by the first two nontrivial diffusion maps eigenvectors. All data points are colored by the <i>a priori known</i> heterogeneity parameter $v_{sP_0}$ . Data are chosen from the case of $\phi = 0.5, k_1 = 0.49, \sigma^2 = 10^{-3}$ for forcing angular velocity equal to the intrinsic one. (right) The first nontrivial diffusion maps eigenvector plotted against the heterogeneity parameter. . . . .	54
4.1	Schematic with reaction steps and corresponding rate constants. $A, B$ are the gas species and $A^*, B^*$ the adsorbed species. . . . .	63
4.2	Step forcing function . . . . .	65
4.3	Schematic with reaction steps and corresponding rate constants. $A, B, C$ are the gas species and $A^*, B^*, C^*$ the adsorbed species. . . . .	66
4.4	Energy diagram of the two states of the step forcing function. The state colored red $U_R$ favors the surface reaction step, while the blue colored state $U_L$ favors desorption. . . . .	69
4.5	Locating periodic steady states with brute-force integration vs. with Newton’s method (variational formulation), for three values of forcing frequencies (colored differently). Integration converging to the same periodic steady state as Newton’s method (denoted as stars). For the integration only the endpoints of each phase are plotted, after the first phase. Note that the rest of the forcing parameters are fixed at $\phi = 0.5, U_L = -0.1eV, \Delta U = 0.6eV$ . . . . .	71
4.6	Newton convergence using Newton’s method with the variational approach vs. Newton Krylov-GMRES $f \approx 6667\text{Hz}, \phi = 0.50, U_L = -1.46eV, \Delta U = 0.6eV$ . . . . .	72
4.7	Energy diagram of the two states of the step forcing function. The state colored red $U_R$ favors the surface reaction step, while the blue colored state $U_L$ favors desorption. . . . .	73

4.8	Oscillation of the binding energies between phases $R$ and $L$ lead to oscillations of all chemical species (and functions thereof): (a) Four oscillations of two of the system variables: $\theta_A$ , the surface coverage of A, and $\theta^*$ , the surface coverage of empty sites. Blue and red colors denote the oscillation phase (see 4.7), (b) The limit cycle point of view of the oscillations on the left. The points of phase change are denoted on the limit cycle. Note that these results are obtained for $f = 10^3 \text{Hz}$ , $\phi = 0.5$ , $U_L = -0.1 \text{eV}$ , $\Delta U = 0.6 \text{eV}$ .	74
4.9	Oscillations of the chemical species during Dynamic Catalysis: (a) Static steady states and their slow and fast manifolds are shown on the limit cycle figure, (b) The net desorption rate of B of a number of points along the limit cycle. These points are separated by a fixed time step.	75
4.10	3D view of the limit cycle ( $f = 10^2 \text{Hz}$ , $\phi = 0.5$ , $U_L = -0.1 \text{eV}$ , $\Delta U = 0.6 \text{eV}$ ) where the Net Desorption rate of B is now the z-axis. Steady states are denoted as stars (red for phase $R$ and blue for $L$ ). Slow manifolds are colored green and fast ones, orange. Direction arrows on different parts of the limit cycle are colored analogously. The projection of the limit cycle is seen on the floor of the 3D plot.	76
4.11	Investigating the effect of forcing frequency: (left) $f = 10 \text{Hz}$ , (middle) $f = 10^3 \text{Hz}$ , (right) $f = 10^8 \text{Hz}$ all for $\phi = 0.5$ , $U_L = -0.1 \text{eV}$ , $\Delta U = 0.6 \text{eV}$ .	77
4.12	Investigating the effect of the duty cycle frequency: (left) $\phi = 0.25$ , (middle) $\phi = 0.5$ , (right) $\phi = 0.75$ all for $f = 10^3 \text{Hz}$ , $U_L = -0.1 \text{eV}$ , $\Delta U = 0.6 \text{eV}$ .	78
4.13	Continuation of periodic steady states w.r.t the forcing frequency using the software AUTO 07p (Doedel et al., 1999). Inset figures show representative projections of the limit cycles. All other forcing parameters were kept fixed at $\phi = 0.50$ , $U_L = -0.5 \text{eV}$ , $\Delta U = 1 \text{eV}$ .	79
4.14	Two parameter solution surface approximated by a grid of points where periodic steady states are found using Newton's method (here, reporting the turnover frequency of A for each such state). Red line on left panel corresponds to the zoomed region on the right. All calculations were performed for fixed $U_L = -1.46 \text{eV}$ , $\Delta U = 0.6 \text{eV}$ .	80
4.15	(left) Optimization result on top of the one-parameter continuation branch (for comparison, see right panel, Fig. 4.14). BO optimum location in agreement with what was expected from continuation, (right) optimal forcing function shape.	81
4.16	The limit cycle corresponding to the optimal forcing function shape found with Bayesian Optimization (Fig.4.15). Also plotted, are the two static steady states and all relevant eigendirections/ stable manifolds. The black arrows on the limit cycle denote the direction of time.	82
4.17	(left) Eigenvalues for the two static steady states, on top forcing frequency. The eigenvalues are colored according to their corresponding relevant eigendirections shown in Fig.4.16 (if any). (right) The turnover frequency over one period for the optimum found with BO and for less or greater $\phi$ values.	83
4.18	(a) A single Radial Basis Function (RBF) parametrized by four parameters: $w_0, w_{amp}, w_{center}, w_{spread}$ , (b) Random sample of RBFs.	84
4.19	Out of all RBFs that can be added to the $L$ phase of the already optimal step forcing function, BO locates the one that maximizes $TO_{fA}$ .	85
4.20	Limit cycle projections for the two optimization solutions found: one for the optimal step forcing function (blue) and one for the optimal additional RBF (red) with the optimal step forcing function. The left inset shows these two optimal forcing functions. The right inset shows a zoomed region where the trajectories of the two limit cycles diverge, along with the relevant eigendirections. Black arrows denote the time direction. Note that the steady states (and respective eigenvalues/ eigenvectors) will be slightly different for the two cases shown here, but we include only one for simplicity.	86

4.21	Two of the converged solutions of optimization with two RBFs. The converged solutions approximate (left) the step forcing function (right) the step forcing function with the additional RBF, both previously examined. . . . .	87
5.1	Overview of proposed algorithmic pipeline: from individual bacterium motility patterns to data-driven, possibly partially physics-informed, surrogate models for chemotactic Partial Differential Equations (PDEs). . . . .	96
5.2	Parameters used for the PDE simulation of the Extended Keller-Segel Model and a schematic representation of the computational domain. . . . .	101
5.3	PDE simulations representing the ground truth data: (left) $b(r, t)$ field and (right) $c(r, t)$ field. For clarity, arrows are added to denote the direction of time. . . . .	102
5.4	Black-box learning of both PDEs with an Artificial Neural Network: (left) Integration results for the <b>first</b> data-driven PDE (for $b(r, t)$ ) and (right) relative error (%). . . . .	103
5.5	Black-box learning of both PDEs with an Artificial Neural Network: (left) Integration results for the <b>second</b> data-driven PDE (for $c(r, t)$ ) and (right) relative error (%). . . . .	103
5.6	Black-box learning of both PDEs with Gaussian Process Regression: (left) Integration results for the <b>first</b> data-driven PDE (for $b(r, t)$ ) and (right) relative error (%). Note that the white horizontal line separates the training data set from the rest of the validation data set. . . . .	104
5.7	Black-box learning of both PDEs with Gaussian Process Regression: (left) Integration results for the <b>second</b> data-driven PDE (for $c(r, t)$ ) and (right) relative error (%). . . . .	104
5.8	Black-box learning with a Neural Network: (left) Integration results for the data-driven PDE and (right) % relative error. . . . .	106
5.9	Black-box learning with Gaussian Process Regression: (left) Integration results for the data-driven PDE and (right) % relative error. . . . .	106
5.10	Black-box partial-information learning with a Neural Network: (left) Integration results for the data-driven PDE and (right) % relative error. . . . .	108
5.11	Black-box partial-information learning with Gaussian Process Regression: (left) Integration results for the data-driven PDE and (right) % relative error. . . . .	109
5.12	Gray-box learning with a Neural Network: (left) Integration results for the data-driven PDE and (right) % relative error. . . . .	110
5.13	Gray-box learning with Gaussian Process Regression: (left) Integration results for the data-driven PDE and (right) % relative error. . . . .	110
5.14	Gray-box partial-information learning with a Neural Network: (left) Integration results for the data-driven PDE and (right) % relative error. . . . .	111
5.15	Gray-box partial-information learning with Gaussian Process Regression: (left) Integration results for the data-driven PDE and (right) % relative error. . . . .	112
5.16	Transformation on the inputs: (left) representative profiles of both fields $b(r, t), c(r, t)$ , (middle) visualization of the singularity, (right) transformed variable. . . . .	114
5.17	Learning the $c$ -field with a Neural Network: (left) Field prediction and (right) % relative error. . . . .	114
5.18	Learning the $c$ -field with Gaussian Process Regression: (left) Field prediction and (right) % relative error. . . . .	115

5.19	A schematic of the proposed data-driven numerical methodology: given the bacterial density profile in space and time, as well as the chemoattractant profile, (and, for the closure correction approach, the analytically available approximate chemotactic term $CH_g$ ) from Eq.(5.14) and their spatial derivatives, we extract possible candidates as input features for learning the PDE right-hand-side (RHS) operator. Then, we train our Machine Learning models (ANN or GP) to approximate $\frac{\partial b}{\partial t}$ or $CH$ (the “ground truth” chemotactic term). . . . .	122
5.20	(a) <b>Training/reconstruction</b> case. Ground Truth (GT) evolution for $\mu = 7, \sigma = 1.25$ : smoothed profiles of bacterial density derived from post-processing agent-based simulations. (b) Quantitative performance of representative data-driven models (DD) trained by Gaussian Process Regression (GP), reduced Gaussian Process Regression with ARD (GP+ARD) or a Neural Network (NN): relative error (%) based on maximum density for black-box model, gray-box model, and correction model (first, second and third rows respectively). (c) Qualitative comparison of profiles of bacterial density at $t = 1000$ s: PDEs learned through (left) Gaussian process; (middle) Reduced Gaussian processes (with ARD); and (right) Neural Networks. Profiles are colored as follows: blue – ground truth, orange – black box, green – gray box, purple – PDE Eq.[3]. The bottom row is a blowup of the profile’s peak. Note that for the Monte-Carlo simulations the initial state ( $t = 0$ s) is at $x = 5.5$ cm for all agents, while all PDE simulations (DD models, PDE in 5.14) begin at $t = 20$ s. No flux boundary conditions were used, consistent with (Erban and Othmer, 2004). . . . .	127
5.21	(a) <b>Testing</b> case. Ground Truth (GT) evolution for $\mu = 6.5, \sigma = 1.35$ : smoothed profiles of bacterial density derived from post-processing agent-based simulations. (b) Quantitative performance of representative data-driven models (DD) trained by Gaussian Process Regression (GP), reduced Gaussian Process Regression with ARD (GP+ARD) or a Neural Network (NN): relative error (%) based on maximum density for black-box model, gray-box model, and correction model (first, second and third rows respectively). (c) Qualitative comparison of profiles of bacterial density at $t = 1000$ s: PDEs learned through (left) Gaussian process; (middle) Reduced Gaussian processes (with ARD); and (right) Neural Networks. Profiles are colored as follows: blue – ground truth, orange – black box, green – gray box, purple – PDE Eq.[3]. The bottom row is a blowup of the profile’s peak. Note that for the Monte-Carlo simulations the initial state ( $t = 0$ s) is at $x = 5.5$ cm for all agents, while all PDE simulations (DD models, PDE in 5.14) begin at $t = 20$ s. No flux boundary conditions were used, consistent with (Erban and Othmer, 2004). . . . .	128
5.22	Pre-processing of experimental measurements: (left) Smoothing in space (right) Interpolation in time. . . . .	131
5.23	Segmentation of the pre-processed data into: boundary corridors/ discarded data, training subset (the complement), subset chosen for reproduction (red rectangle). . . . .	132
5.24	Reproduction of experimental observations using a Data-driven Neural Network for the traveling wave regime: (left) Ground Truth, (middle) Neural-Network PDE integration results, (right) % Relative Error. . . . .	133



# List of Algorithms

1	Pseudo-arclength continuation . . . . .	22
2	Bayesian Optimization - LCB . . . . .	31
3	Bayesian Continuation for Bayesian Optimization . . . . .	33

# Chapter 1

## Introduction

There are various definitions of complexity across different academic fields. In the context of dynamical systems, complex can be a system that exhibits the following three characteristics (Richardson and Chemero, 2014; Gallagher and Appenzeller, 1999; Ottino, 2003):

- i. Consists of (homogeneous or heterogeneous) interacting components/agents.
- ii. The collective behavior of these components/agents exhibits coherent emergent patterns.
- iii. This collective behavior is self-organized, in the sense that is not externally directed.

Equivalently, we can think of them as systems with a large range of temporal and spatial scales, intrinsic high dimensionality, and nonlinear dynamics (Fernex, Noack, and Semaan, 2021)

As our understanding of the natural world deepens, along with breakthroughs in experimental capabilities and computational efficiency, it is now possible (and preferred) to approach complex systems *holistically*. This encourages scientists and engineers to move from reductionism (the Newtonian paradigm) to more realistic representations of complex systems (Heylighen, Cilliers, and Gershenson Garcia, 2007). In other words, the (post-) modern scientist does not consider the “whole” to be equal to the sum of its parts (Aristotle, 350BCE).

This trend was accelerated with the popularization of Machine Learning algorithms and democratization of powerful computational resources. Such algorithms are naturally suited to complex systems, as:

- They can deal with large quantities of data, which are usually required to study high-dimensional,

highly nonlinear, multiscale phenomena (Qi and Majda, 2020; Lee et al., 2020; Chen et al., 2019; Gkeka et al., 2020).

- They can efficiently discover patterns in large data sets, which is essential for studying emergent behaviors in multiscale systems (Bengio, Courville, and Vincent, 2013; Jia et al., 2022).
- They can universally approximate nonlinear surfaces, where the responses (or laws) of complex systems naturally live. Often, such approximations are smooth and tractable enabling surrogate modeling and digital twins (Hornik, Stinchcombe, and White, 1990; Park and Sandberg, 1991; Cybenko, 1989).

In this thesis, Machine Learning is used *synergistically* with traditional modeling to perform tasks such as steady state location, numerical continuation, optimization, and system identification of complex dynamical systems in a variety of settings. An arsenal of Machine Learning algorithms is used for specific purposes, suited for each dynamical system:

**Multiple scales:** Macroscale system identification can be achieved with Machine Learning methods, starting from microscale simulations or experiments. The amount and identity of the proper macroscale variables can be assumed to be known *a priori* (e.g. through already existing approximate models) or can be data-driven (e.g. Automatic Relevance Determination, Autoencoders, Diffusion Maps).

**Nonlinear dynamics:** Nonlinearly activated Neural Networks and Gaussian Process Regression schemes are able to approximate the ground truth nonlinear dynamics given sufficient data.

**High dimensionality:** Unsupervised learning (e.g. Diffusion Maps, Autoencoders) can be used for dimensionality-reduction of high-dimensional data sets. Proper initialization for numerical methods in high-dimensional problems can be achieved with Bayesian methods.

For the purposes of this thesis, focus will be given to three categories of complex systems: **Periodically forced systems (Chapters 3, 4)**, **Multiscale systems (Chapter 5)**, **high-dimensional systems/networks (Chapter 3, 4, 5)**. These categories are by no means mutually exclusive; however, there is great value in examining each one separately.

## 1.1 Periodically Forced Systems

Periodically forced systems are abundant in nature, owing mainly to the periodicity of the day-night cycle (circadian rhythm (Edery, 2000), social dynamics (Li, Schwartz, and Indic, 2020), heat transfer models (Sojoudi et al., 2016), meteorology (Liu and Moncrieff, 1998)) and seasonal cycle (circadian rhythm (Edery, 2000), population dynamics (Gao, Shi, and Li, 2009), epidemiology (Chitnis, Hardy, and Smith, 2012), meteorology (Luther and O'Brien, 1985)), but also to more complex cycles (cell cycle (Mosheiff et al., 2018), autocatalytic reactions (Lin et al., 2004), neuronal electrophysiology (Pakdaman, 2001)). Periodically forced systems have long attracted the attention of both practitioners and theorists. The former because by periodically forcing a dynamical system, one can significantly increase its degrees of freedom and lead to a rich variety of responses, many of which are possibly of practical interest (Starosvetsky and Gendelman, 2007; Wang and Song, 2016; Qiao et al., 2008). The latter because periodically forced dynamical systems exhibit a wealth of dynamics, which can be elegantly connected to fundamental concepts of dynamical systems and mathematics in general (McGehee and Peckham, 1996; Jensen, Bak, and Bohr, 1983; Ding, 1987).

Such systems can be complex, due to the added degrees of freedom, and diversity of responses, resulting from the interaction of a system's intrinsic dynamics and forcing dynamics. Periodically forced dynamical systems can be usefully divided into two categories: systems that exhibit periodic behavior even in the absence of external stimulus (naturally oscillating/ exhibiting **autonomous** limit cycles) and systems that do not (naturally non-oscillating, only **nonautonomous** oscillations possible).

### 1.1.1 Systems with autonomous oscillations

As mentioned, such systems exhibit periodic behavior (limit cycles) even when not forced. For example, consider a generic autonomous dynamical system:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; \mathbf{p}), \quad (1.1)$$

where  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{p} \in \mathbb{R}^m$ ,  $\mathbf{f} : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^n$ . Solutions of this system are maps of the form  $t \rightarrow \mathbf{x}(t)$ ,  $\mathbf{x}(t) : \mathbb{R} \rightarrow \mathbb{R}^n$  (Wiggins, 1990). In certain regimes of parameter values, limit cycles arise, characterized by an intrinsic period  $T_0$ , or, equivalently, by an intrinsic angular frequency  $\omega_0 = 2\pi f_0$ , where  $f_0 = 1/T_0$  is the intrinsic frequency (note that in general  $\omega_0 = \omega_0(\mathbf{p})$ ). The intrinsic period can be found by solving the following set of equations:

$$\mathbf{x} - \mathbf{P}(\mathbf{x}) = 0, \quad (1.2)$$

$$\int_0^{T_0} \langle \mathbf{x}(t) - \mathbf{x}_0, \mathbf{f}(\mathbf{x}_0; \mathbf{p}) \rangle dt = 0, \quad (1.3)$$

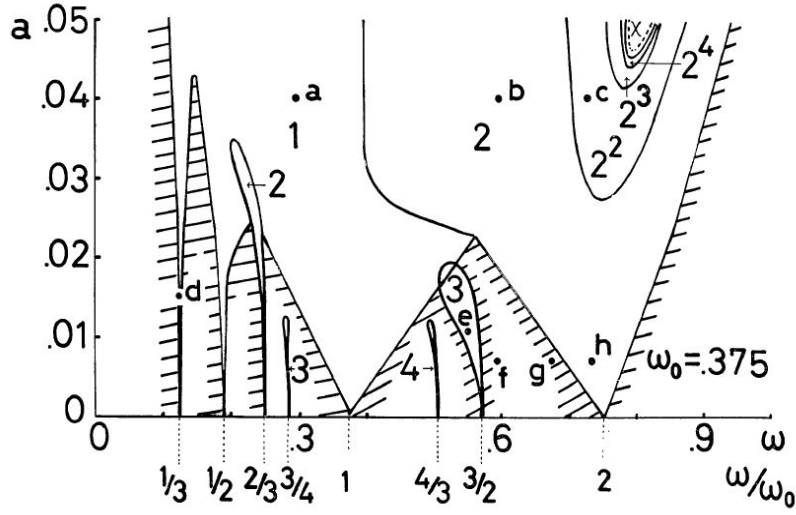
where  $P_i(\mathbf{y}) := \int_0^{T_0} f_i(\mathbf{x}; \mathbf{p}) dt$ ,  $\mathbf{x}(0) = \mathbf{y}$ ,  $i = 1, \dots, n$  and  $\mathbf{x}_0$  is some state.  $\mathbf{P}(\mathbf{x})$  is called the Poincaré map and Eq. 1.2 defines its fixed points. Eq. 1.3 is an *anchor* equation, necessary to *also* solve for  $T_0$ . This anchor equation requires orthogonality to some reference state  $\mathbf{x}_0$ , but more options are possible, such as restricting the solution to a  $(n - 1)$ -dimensional surface.

This autonomous dynamical system becomes nonautonomous, when forced with a time-dependent term  $\mathbf{g} : \mathbb{R}^{k+2} \rightarrow \mathbb{R}^n$ , for example:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; \mathbf{p}) + \mathbf{g}(t; \omega, \mathbf{p}_f), \quad (1.4)$$

where  $\omega > 0$ ,  $\mathbf{p}_f \in \mathbb{R}^k$  are the forcing angular frequency and a vector with other forcing parameters (for example parametrizing the *shape*, with the most fundamental being the forcing amplitude  $a$ ), respectively. Note that the forcing is usually introduced to just one (or a few ODEs) and not all; however, this suffices to affect the periodic response of the entire  $n$ -dimensional system, given that all ODEs are coupled. Also note that the forcing introduces extra  $k + 1$  degrees of freedom, creating a  $(k + 1)$ -family of dynamical systems (Peckham, 1988).

A useful example is the Brusselator, a 2D theoretical model of an autocatalytic reaction, which exhibits limit cycles for certain parameter values. Forcing is introduced to one of the ODEs, and has the form  $g(t, a, \omega) = a \cos(\omega t)$ . The effect of the two forcing parameters  $(a, \omega)$  can be explored by constructing a bifurcation diagram. Here, reporting the bifurcation diagram constructed by Tomita and Kai, 1978:



**Figure 1.1:** Bifurcation diagram of the Brusselator w.r.t the relative forcing angular frequency  $\omega/\omega_0$  and forcing amplitude  $a$ . Interested readers are referred to the original manuscript (Tomita and Kai, 1978) for explanation of numbered/indexed points on the diagram.

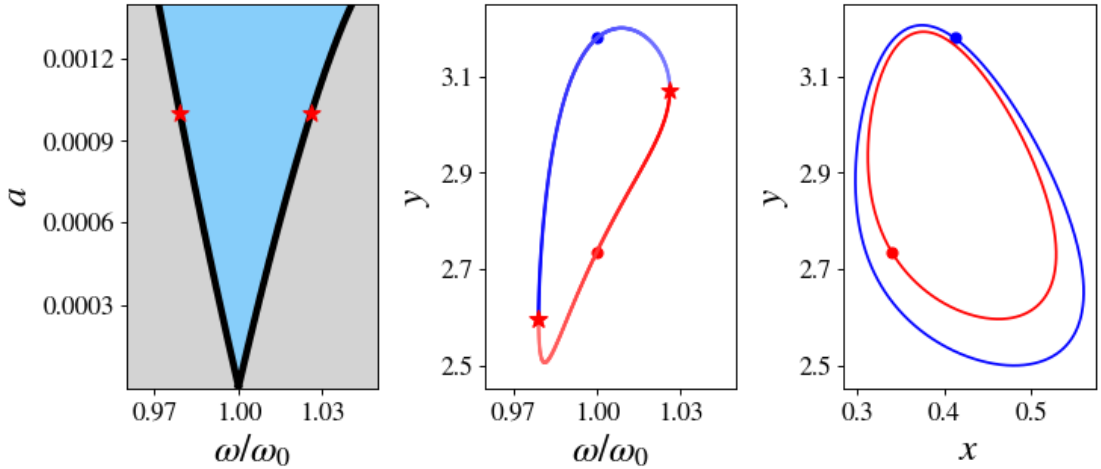
As shown in the Fig.1.1 what really matters is how the forcing angular frequency changes w.r.t. the intrinsic one (i.e. ratio  $\omega/\omega_0$ ). At certain values  $\omega/\omega_0 = k/m$ , (where  $k/m$  is rational)  $k : m$ -resonancies are observed, i.e. the dynamical system performs  $m$  oscillations during the time needed for the external forcing to complete  $k$  oscillations. As expected, as  $a \rightarrow 0$ , the limit cycles of the forced system are more and more akin to the system's autonomous limit cycles. Shaded regions denote quasiperiodicity, while, for higher values of the forcing amplitude, chaos emerges. Of most practical interest is the 1:1 resonance region, (called the 1:1 *resonance horn*) where the dynamical system is synchronized with the external stimulus. For low amplitudes, at the bilateral boundaries of the 1:1 resonance horn, synchrony is lost via a saddle-node bifurcation of limit cycles; at that point, a stable and an unstable limit cycle coalesce and disappear. The closed curve formed by that stable and unstable branch (for a fixed forcing amplitude) is called an *isola*.

To construct such a diagram, finding periodic steady states conveniently reduces to solving a boundary value problem. Following the notation of Eq.1.4 and denoting  $\mathbf{F} := \mathbf{f} + \mathbf{g}$  the following can be defined:

$$\mathbf{y}_{j+1} = \mathbf{S}(\mathbf{y}_j), \quad S_i(\mathbf{y}) := \int_0^T F_i(\mathbf{x}, t; \mathbf{p}, \omega, \mathbf{p}_f) dt, \quad i = 1, \dots, n, \quad \mathbf{x}(0) = \mathbf{y} \quad (1.5)$$

$$\mathbf{R}(\mathbf{x}) := \mathbf{x} - \mathbf{S}(\mathbf{x}), \quad (1.6)$$

where  $T = \omega/2\pi$  is the forcing period.  $\mathbf{S} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is called the *stroboscopic map* and  $\mathbf{R} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  the stroboscopic map's residual function. One can think of the stroboscopic map as a "isophase" version of a Poincaré map, where the period is a priori known and no anchor equation is required (Kevrekidis, Schmidt, and Aris, 1986). Fixed points of 1.5 (or, equivalently, roots of 1.6) correspond to limit cycles of the forced dynamical system. Continuation of such fixed points w.r.t. the forcing angular frequency  $\omega$  results in isolas, and continuation of the codimension-two bifurcations on the isola corresponding to saddle-node bifurcations of limit cycles w.r.t.  $\omega, a$  results in resonance horns. That leads to the hierarchy of important points/loci of points better shown in Fig. 1.2.



**Figure 1.2:** Hierarchy of points/loci of points for the periodically forced system of the Brusselator. Following a top-down approach: (left) 1:1 resonance horn. The blue region is where synchronization is possible, while the gray region denotes quasiperiodicity. Every point in the blue region (boundary inclusive) corresponds to (at least one) limit cycle. Two sample bifurcations at  $a = 0.001$  are denoted with red stars. (middle) projection of the isola for  $a = 0.001$ . The red stars in the left subfigure are now the bilateral limits of the isola where the stable (blue) and unstable (red) branches coalesce. Every point of the isola is a limit cycle (or, equivalently, a stroboscopic map fixed point). Two points are sampled for  $\omega/\omega_0 = 1$ . (right) limit cycles (blue-stable, red-unstable) for the two points selected in the middle subfigure.

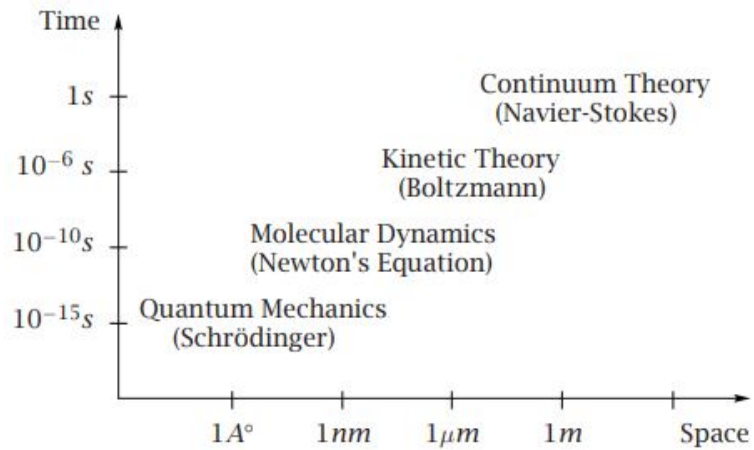
## 1.1.2 Systems without autonomous oscillations

In this case, the system's autonomous dynamics do not exhibit limit cycles, therefore an intrinsic angular frequency  $\omega_0$  can not be defined, neither a variety or resonances are possible. However, it is expected that the dynamics of the forced system will be periodic and in fact always synchronized with the external forcing. Forcing parameter  $\omega$  but also, any additional  $k$  degrees of freedom in  $\mathbf{p}_f$  (as in 1.4), create a  $(k + 1)$ -family of periodic responses. This is again a complex system, as nonlinearities emerge

from the interplay of intrinsic dynamics and the periodic function.

## 1.2 Multiscale Systems

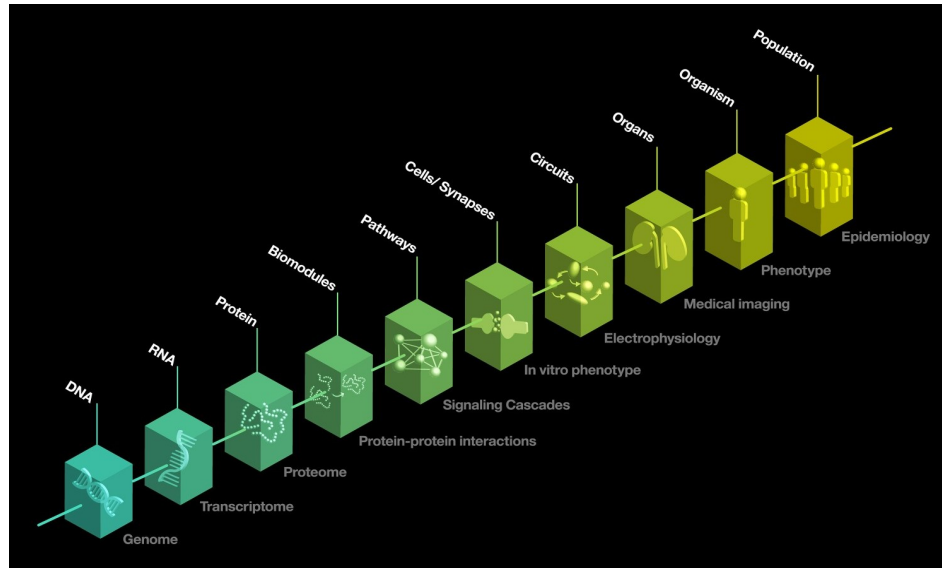
Multiscale systems are those which can be studied at multiple spatial or temporal scales. In fact, any system can be viewed as a multiscale system, depending on the desired resolution and the experimental/computational/theoretical resources available (Weinan and Engquist, 2003; Walpole, Papin, and Peirce, 2013).



**Figure 1.3:** Hierarchy of multiple scales in fluid mechanics along with suitable laws for each scale (Weinan and Engquist, 2003).

At different scales, equations of different character are needed: For the macroscale, continuum models are constructed (e.g. Algebraic Equations - AEs, Ordinary Differential Equations - ODEs, Partial Differential Equations - PDEs), while for the microscale, usually stochastic (possibly discrete space) models are preferred (e.g. Monte Carlo - MC, Master Equation - ME, Stochastic Differential Equations - SDEs) (Vlachos, 2005; Lu and Kaxiras, 2004). The Fokker-Planck equation is mentioned here as the quintessential example: The stochastic motion of particles under a potential (continuous-time, uncountable-space Markov process) can be described using an SDE. In the macroscale, an analytic, deterministic PDE can be derived describing the particle density (Givon, Kupferman, and Stuart, 2004). Another example relevant to fluid mechanics is presented in 1.3.





**Figure 1.4:** Hierarchy of multiple scales in biology along with examples of data that can be collected at each scale (*Multiscale Systems Biology and Modeling*, howpublished = [https://researcher.watson.ibm.com/researcher/view\\_group.php?id=5372](https://researcher.watson.ibm.com/researcher/view_group.php?id=5372), note = Accessed: 2022-07-09 n.d.).

In most settings, *only* the macroscale (equivalently, in the literature: emergent, latent, hidden, slow, low-dimensional, reduced-order, effective, collective) dynamics are of practical interest, especially when:

- **Individual agents' dynamics are not informative.** The choice to study the emergent dynamics reflects the relevance of the corresponding time/space scale to real-world problems. For example, we are not interested in the velocities of the individual particles of a gas, but at the variance of their velocity distribution, or equivalently the gas' temperature (Mandl, 1988).
- **It is not straightforward how to couple phenomena at different scales:** Multiscale modeling is numerically complex and computationally demanding due to separation in scales (Ricardez-Sandoval, 2011; Lu and Kaxiras, 2004). Furthermore it is nontrivial to couple computations of different fidelity, each performed under different assumptions. In fact, multiscale closures might be only approximate or even intractable (Kevrekidis et al., 2003; Chatterjee and Vlachos, 2006).
- **The microscale dynamics are unavailable.** In this case, all one can do is *probe* the system, judiciously gather data, and uncover the effective macroscopic variables and/or the laws of their evolution in a data-driven way (Meier-Schellersheim, Fraser, and Klauschen, 2009; Dsilva et al., 2018). This is especially useful and important in biology/biomedicine where scales are spatial, temporal and functional (see Fig. 1.4) (Alber et al., 2019).

After examining *why* the focus is concentrated in macroscopic dynamics, it is worth examining some

examples of *how* this can be done. Here, a non-exhaustive collection of methods is presented:

- **Derived:** Approximate invariant manifolds (Titi, 1990; Jolly, Kevrekidis, and Titi, 1990), averaging and homogenization methods (G.A. Pavliotis, 2008; Ye, Yang, and Maggioni, 2021), white noise approximation (Givon, Kupferman, and Stuart, 2004), moment closures (Xue and Othmer, 2009), Mori-Zwanzig formalism (Mori, 1965; Zwanzig, 2001).
- **Approximated / Data-driven:** Equation-free framework (Kevrekidis et al., 2003), Proper Orthogonal Decomposition (also known as PCA/SVD) (Deane et al., 1991; Liao and Maggioni, 2019), Dynamic Mode Decomposition (Schmid, 2010), Koopman operator (Williams, Kevrekidis, and Rowley, 2015), Diffusion Maps (Singer et al., 2009; Kemeth et al., 2022), Deep Neural Networks (Zhang et al., 2018), Convolutional Neural Networks (Arbabi et al., 2020), Gaussian Process Regression (Lee et al., 2020), Recurrent Neural Networks (Vlachas et al., 2022) and Autoencoders (Chen, Tan, and Ferguson, 2018).

### 1.3 High-dimensional Systems and Networks

High dimensionality has been a significant hurdle in the study of complex systems, especially in performing tasks such as sampling, function approximation or nearest neighbor location (Rabitz and Alis, 1999; Scott and Thompson, 1983). For example, sampling a unit hypercube in  $\mathbb{R}^d$  equidistantly in each dimension (distance  $D$ ), would require  $(1/D)^d$  points - the well-known curse of dimensionality (Bellman, 1957). Even geometrical intuition often fails as the dimensionality increases, e.g.: the ratio of the volume of a sphere and its circumscribed cube approximates zero; the volume of a sphere is mostly contained in its outer shell. These are in agreement with unusual concentration phenomena in high dimensions in probability theory (Lee and Verleysen, 2007).

In general, the problems associated with high-dimensional systems are the great computational cost, lack of intuition and difficult visualization. However, not all dimensions are always useful. Some of them might be irrelevant for a specific task. Others might be relevant but not *independent* w.r.t. the rest. The former case can be resolved via supervised relevance determination methods (Aha and Bankert, 1996) and the latter with unsupervised dimensionality reduction techniques (Lee and Verleysen, 2007).

High-dimensional dynamical systems are often represented as networks (or graphs). In this case, apart from the node dynamics, the way the nodes are connected (network *structure*) matters and can be decisive to the emergent dynamics (Ottino, 2003). Some popular examples are compartmental models

in epidemiology (Brauer, 2008), social networks (Zhao et al., 2011) and coupled oscillators (Thiem et al., 2020).

# References

- Bellman, Richard (1957). *Dynamic Programming*. Dover Publications. ISBN: 9780486428093.
- Mori, Hazime (1965). "Transport, Collective Motion, and Brownian Motion\*"). In: *Progress of Theoretical Physics* 33.3, pp. 423–455. ISSN: 0033-068X. DOI: 10.1143/PTP.33.423. eprint: <https://academic.oup.com/ptp/article-pdf/33/3/423/5428510/33-3-423.pdf>. URL: <https://doi.org/10.1143/PTP.33.423>.
- Tomita, Kazuhisa and Tohru Kai (1978). "Stroboscopic phase portrait and strange attractors". In: *Physics Letters A* 66.2, pp. 91–93. ISSN: 0375-9601. DOI: [https://doi.org/10.1016/0375-9601\(78\)90004-X](https://doi.org/10.1016/0375-9601(78)90004-X). URL: <https://www.sciencedirect.com/science/article/pii/037596017890004X>.
- Jensen, M. Høgh, Per Bak, and Tomas Bohr (1983). "Complete Devil's Staircase, Fractal Dimension, and Universality of Mode-Locking Structure in the Circle Map". In: *Phys. Rev. Lett.* 50 (21), pp. 1637–1639. DOI: 10.1103/PhysRevLett.50.1637. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.50.1637>.
- Scott, David and James Thompson (1983). "Probability density estimation in higher dimension". In: *Computer Science and Statistics: Proceedings of the Fifteenth Symposium on the Interface*.
- Luther, Mark E. and James J. O'Brien (1985). "A model of the seasonal circulation in the Arabian Sea forced by observed winds". In: *Progress in Oceanography* 14, pp. 353–385. ISSN: 0079-6611. DOI: [https://doi.org/10.1016/0079-6611\(85\)90017-5](https://doi.org/10.1016/0079-6611(85)90017-5). URL: <https://www.sciencedirect.com/science/article/pii/0079661185900175>.
- Kevrekidis, I.G., L.D. Schmidt, and R. Aris (1986). "Some common features of periodically forced reacting systems". In: *Chemical Engineering Science* 41.5, pp. 1263–1276. ISSN: 0009-2509. DOI: [https://doi.org/10.1016/0009-2509\(86\)87099-3](https://doi.org/10.1016/0009-2509(86)87099-3). URL: <https://www.sciencedirect.com/science/article/pii/0009250986870993>.
- Ding, E. J. (1987). "Analytic treatment of a driven oscillator with a limit cycle". In: *Phys. Rev. A* 35 (6), pp. 2669–2683. DOI: 10.1103/PhysRevA.35.2669. URL: <https://link.aps.org/doi/10.1103/PhysRevA.35.2669>.
- Mandl, F. (1988). *Statistical physics*. English. Chichester [West Sussex]; New York: Wiley.
- Peckham, Bruce Bartlett (1988). "The closing of resonance horns for periodically forced oscillators". PhD thesis. University of Minnesota.
- Cybenko, G. (1989). "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals and Systems* 2.4, pp. 303–314. ISSN: 1435-568X. DOI: 10.1007/BF02551274. URL: <https://doi.org/10.1007/BF02551274>.
- Hornik, Kurt, Maxwell Stinchcombe, and Halbert White (1990). "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks". In: *Neural Networks* 3.5, pp. 551–560. ISSN: 0893-6080. DOI: [https://doi.org/10.1016/0893-6080\(90\)90005-6](https://doi.org/10.1016/0893-6080(90)90005-6). URL: <https://www.sciencedirect.com/science/article/pii/0893608090900056>.

- Jolly, M.S., I.G. Kevrekidis, and E.S. Titi (1990). "Approximate inertial manifolds for the Kuramoto-Sivashinsky equation: Analysis and computations". In: *Physica D: Nonlinear Phenomena* 44.1, pp. 38–60. ISSN: 0167-2789. DOI: [https://doi.org/10.1016/0167-2789\(90\)90046-R](https://doi.org/10.1016/0167-2789(90)90046-R). URL: <https://www.sciencedirect.com/science/article/pii/016727899090046R>.
- Titi, Edriss S (1990). "On approximate Inertial Manifolds to the Navier-Stokes equations". In: *Journal of Mathematical Analysis and Applications* 149.2, pp. 540–557. ISSN: 0022-247X. DOI: [https://doi.org/10.1016/0022-247X\(90\)90061-J](https://doi.org/10.1016/0022-247X(90)90061-J). URL: <https://www.sciencedirect.com/science/article/pii/0022247X9090061J>.
- Wiggins, S. (1990). *Introduction to applied nonlinear dynamical Systems and Chaos*. New York: Springer.
- Deane, A. E., I. G. Kevrekidis, G. E. Karniadakis, and S. A. Orszag (1991). "Low-dimensional models for complex geometry flows: Application to grooved channels and circular cylinders". In: *Physics of Fluids A: Fluid Dynamics* 3.10, pp. 2337–2354. DOI: 10.1063/1.857881. eprint: <https://doi.org/10.1063/1.857881>. URL: <https://doi.org/10.1063/1.857881>.
- Park, J. and I. W. Sandberg (1991). "Universal Approximation Using Radial-Basis-Function Networks". In: *Neural Computation* 3.2, pp. 246–257. DOI: 10.1162/neco.1991.3.2.246.
- Aha, David W. and Richard L. Bankert (1996). "A Comparative Evaluation of Sequential Feature Selection Algorithms". In: *Learning from Data: Artificial Intelligence and Statistics V*. Ed. by Doug Fisher and Hans-J. Lenz. New York, NY: Springer New York, pp. 199–206. ISBN: 978-1-4612-2404-4. DOI: 10.1007/978-1-4612-2404-4\_19. URL: [https://doi.org/10.1007/978-1-4612-2404-4\\_19](https://doi.org/10.1007/978-1-4612-2404-4_19).
- McGehee, Richard P. and Bruce B. Peckham (1996). "Arnold Flames And Resonance Surface Folds". In: *International Journal of Bifurcation and Chaos* 06.02, pp. 315–336. DOI: 10.1142/S0218127496000072. eprint: <https://doi.org/10.1142/S0218127496000072>. URL: <https://doi.org/10.1142/S0218127496000072>.
- Liu, Changhai and Mitchell W. Moncrieff (1998). "A Numerical Study of the Diurnal Cycle of Tropical Oceanic Convection". In: *Journal of the Atmospheric Sciences* 55.13, pp. 2329–2344. DOI: 10.1175/1520-0469(1998)055<2329:ANSOTD>2.0.CO;2. URL: [https://journals.ametsoc.org/view/journals/atsc/55/13/1520-0469\\_1998\\_055\\_2329\\_ansotd\\_2.0.co\\_2.xml](https://journals.ametsoc.org/view/journals/atsc/55/13/1520-0469_1998_055_2329_ansotd_2.0.co_2.xml).
- Gallagher, Richard and Tim Appenzeller (1999). "Beyond Reductionism". In: *Science* 284.5411, pp. 79–79. DOI: 10.1126/science.284.5411.79. eprint: <https://www.science.org/doi/pdf/10.1126/science.284.5411.79>. URL: <https://www.science.org/doi/abs/10.1126/science.284.5411.79>.
- Rabitz, Herschel and Ömer F. Aliş (1999). "General foundations of high-dimensional model representations". In: *Journal of Mathematical Chemistry* 25.2, pp. 197–233. ISSN: 1572-8897. DOI: 10.1023/A:1019188517934. URL: <https://doi.org/10.1023/A:1019188517934>.
- Ederly, Isaac (2000). "Circadian rhythms in a nutshell". In: *Physiological Genomics* 3.2, pp. 59–74. DOI: 10.1152/physiolgenomics.2000.3.2.59. eprint: <https://doi.org/10.1152/physiolgenomics.2000.3.2.59>. URL: <https://doi.org/10.1152/physiolgenomics.2000.3.2.59>.
- Pakdaman, K. (2001). "Periodically forced leaky integrate-and-fire model". In: *Phys. Rev. E* 63 (4), p. 041907. DOI: 10.1103/PhysRevE.63.041907. URL: <https://link.aps.org/doi/10.1103/PhysRevE.63.041907>.
- Zwanzig, Robert (2001). *Nonequilibrium statistical mechanics*. English. URL: <http://www.books24x7.com/marc.asp?bookid=34085>.
- Kevrekidis, Ioannis G., C. William Gear, James M. Hyman, Panagiotis G Kevrekidid, Olof Runborg, and Constantinos Theodoropoulos (2003). "Equation-Free, Coarse-Grained

- Multiscale Computation: Enabling Mocosopic Simulators to Perform System-Level Analysis". en. In: *Commun. Math. Sci.* 1.1, pp. 715–762. URL: <http://dml.mathdoc.fr/item/1119655353>.
- Ottino, J. M. (2003). "Complex systems". In: *AIChE Journal* 49.2, pp. 292–299. DOI: <https://doi.org/10.1002/aic.690490202>. eprint: <https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/aic.690490202>. URL: <https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690490202>.
- Weinan, E and Bjorn Engquist (2003). "Multiscale Modeling and Computation". In: *NOTICES OF THE AMS* 50.9.
- Givon, Dror, Raz Kupferman, and Andrew M. Stuart (2004). "Extracting macroscopic dynamics: model problems and algorithms". In: *Nonlinearity* 17.
- Lin, Anna L., Aric Hagberg, Ehud Meron, and Harry L. Swinney (2004). "Resonance tongues and patterns in periodically forced reaction-diffusion systems". In: *Phys. Rev. E* 69 (6), p. 066217. DOI: 10.1103/PhysRevE.69.066217. URL: <https://link.aps.org/doi/10.1103/PhysRevE.69.066217>.
- Lu, Gang and Efthimios Kaxiras (2004). "An Overview of Multiscale Simulations of Materials". In: *Handbook of Theoretical and Computational Nanotechnology*, American Scientific Publishers 10.
- Vlachos, Dionisios G. (2005). "A Review of Multiscale Analysis: Examples from Systems Biology, Materials Engineering, and Other Fluid–Surface Interacting Systems". In: *Advances in Chemical Engineering*. Ed. by Guy B. Marin. Vol. 30. Advances in Chemical Engineering. Academic Press, pp. 1–61. DOI: [https://doi.org/10.1016/S0065-2377\(05\)30001-9](https://doi.org/10.1016/S0065-2377(05)30001-9). URL: <https://www.sciencedirect.com/science/article/pii/S0065237705300019>.
- Chatterjee, Abhijit and Dionisios G. Vlachos (2006). "Multiscale spatial Monte Carlo simulations: Multigridding, computational singular perturbation, and hierarchical stochastic closures". In: *The Journal of Chemical Physics* 124.6, p. 064110. DOI: 10.1063/1.2166380. eprint: <https://doi.org/10.1063/1.2166380>. URL: <https://doi.org/10.1063/1.2166380>.
- Heylighen, Francis, Paul Cilliers, and Carlos Gershenson Garcia (2007). "Complexity and Philosophy". English. In: *Complexity, Science and Society*. Ed. by J. Bogg and R. Geyer. Radcliffe Publishing, pp. 117–134. ISBN: 978-1-84619-203-6.
- Lee, John and Michel Verleysen (2007). *Nonlinear Dimensionality Reduction*. Vol. 8226. ISBN: 978-0-387-39350-6. DOI: 10.1007/978-0-387-39351-3.
- Starosvetsky, Y. and O. V. Gendelman (2007). "Attractors of harmonically forced linear oscillator with attached nonlinear energy sink. II: Optimization of a nonlinear vibration absorber". In: *Nonlinear Dynamics* 51.1, p. 47. ISSN: 1573-269X. DOI: 10.1007/s11071-006-9168-z. URL: <https://doi.org/10.1007/s11071-006-9168-z>.
- Brauer, Fred (2008). "Compartmental Models in Epidemiology". In: *Mathematical Epidemiology*. Ed. by Fred Brauer, Pauline van den Driessche, and Jianhong Wu. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 19–79. ISBN: 978-3-540-78911-6. DOI: 10.1007/978-3-540-78911-6\_2. URL: [https://doi.org/10.1007/978-3-540-78911-6\\_2](https://doi.org/10.1007/978-3-540-78911-6_2).
- G.A. Pavliotis, A.M. Stuart (2008). *Multiscale Methods*. Springer New York. DOI: 10.1007/978-0-387-73829-1. URL: <https://doi.org/10.1007%2F978-0-387-73829-1>.
- Qiao, L., X. Li, I. G. Kevrekidis, C. Punckt, and H. H. Rotermund (2008). "Enhancement of surface activity in CO oxidation on Pt(110) through spatiotemporal laser actuation". In: *Phys. Rev. E* 77 (3), p. 036214. DOI: 10.1103/PhysRevE.77.036214. URL: <https://link.aps.org/doi/10.1103/PhysRevE.77.036214>.
- Gao, Meng, Honghua Shi, and Zizhen Li (2009). "Chaos in a seasonally and periodically forced phytoplankton–zooplankton system". In: *Nonlinear Analysis: Real World Applications* 10.3,

- pp. 1643–1650. ISSN: 1468-1218. DOI: <https://doi.org/10.1016/j.nonrwa.2008.02.005>. URL: <https://www.sciencedirect.com/science/article/pii/S1468121808000473>.
- Meier-Schellersheim, Martin, Iain D. C. Fraser, and Frederick Klauschen (2009). “Multiscale modeling for biologists”. In: *WIREs Systems Biology and Medicine* 1.1, pp. 4–14. DOI: <https://doi.org/10.1002/wsbm.33>. eprint: <https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wsbm.33>. URL: <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wsbm.33>.
- Singer, Amit, Radek Erban, Ioannis G. Kevrekidis, and Ronald R. Coifman (2009). “Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps”. In: *Proceedings of the National Academy of Sciences* 106.38, pp. 16090–16095. DOI: [10.1073/pnas.0905547106](https://doi.org/10.1073/pnas.0905547106). eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.0905547106>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.0905547106>.
- Xue, Chuan and Hans G. Othmer (2009). “Multiscale Models of Taxis-Driven Patterning in Bacterial Populations”. In: *SIAM Journal on Applied Mathematics* 70.1, pp. 133–167. DOI: [10.1137/070711505](https://doi.org/10.1137/070711505). eprint: <https://doi.org/10.1137/070711505>. URL: <https://doi.org/10.1137/070711505>.
- Schmid, Peter J. (2010). “Dynamic mode decomposition of numerical and experimental data”. In: *Journal of Fluid Mechanics* 656, 5–28. DOI: [10.1017/S0022112010001217](https://doi.org/10.1017/S0022112010001217).
- Ricardez-Sandoval, Luis A. (2011). “Current challenges in the design and control of multiscale systems”. In: *The Canadian Journal of Chemical Engineering* 89.6, pp. 1324–1341. DOI: <https://doi.org/10.1002/cjce.20607>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cjce.20607>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cjce.20607>.
- Zhao, Kun, Juliette Stehlé, Ginestra Bianconi, and Alain Barrat (2011). “Social network dynamics of face-to-face interactions”. In: *Phys. Rev. E* 83 (5), p. 056109. DOI: [10.1103/PhysRevE.83.056109](https://doi.org/10.1103/PhysRevE.83.056109). URL: <https://link.aps.org/doi/10.1103/PhysRevE.83.056109>.
- Chitnis, Nakul, Diggory Hardy, and Thomas Smith (2012). “A Periodically-Forced Mathematical Model for the Seasonal Dynamics of Malaria in Mosquitoes”. In: *Bulletin of Mathematical Biology* 74.5, pp. 1098–1124. ISSN: 1522-9602. DOI: [10.1007/s11538-011-9710-0](https://doi.org/10.1007/s11538-011-9710-0). URL: <https://doi.org/10.1007/s11538-011-9710-0>.
- Bengio, Yoshua, Aaron Courville, and Pascal Vincent (2013). “Representation Learning: A Review and New Perspectives”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8, pp. 1798–1828. DOI: [10.1109/TPAMI.2013.50](https://doi.org/10.1109/TPAMI.2013.50).
- Walpole, Joseph A., Jason A. Papin, and Shayn M. Peirce (2013). “Multiscale Computational Models of Complex Biological Systems”. In: *Annual Review of Biomedical Engineering* 15.1, pp. 137–154. DOI: [10.1146/annurev-bioeng-071811-150104](https://doi.org/10.1146/annurev-bioeng-071811-150104). eprint: <https://doi.org/10.1146/annurev-bioeng-071811-150104>. URL: <https://doi.org/10.1146/annurev-bioeng-071811-150104>.
- Richardson, Michael J. and Anthony Chemero (2014). “Complex dynamical systems and embodiment.” In: *Routledge handbooks in philosophy*. New York, NY, US: Routledge/Taylor & Francis Group, pp. 39–50. ISBN: 978-0-415-62361-2 (Hardcover); 978-1-315-77584-5 (PDF).
- Williams, Matthew O., Ioannis G. Kevrekidis, and Clarence W. Rowley (2015). “A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition”. In: *Journal of Nonlinear Science* 25.6, pp. 1307–1346. ISSN: 1432-1467. DOI: [10.1007/s00332-015-9258-5](https://doi.org/10.1007/s00332-015-9258-5). URL: <https://doi.org/10.1007/s00332-015-9258-5>.
- Sojoudi, Atta, Suvash C. Saha, Ali M. Sefidan, and Y.T. Gu (2016). “Natural convection subject to sinusoidal thermal forcing on inclined walls and heat source located on bottom wall of an attic-shaped space”. In: *Energy and Buildings* 128, pp. 845–866. ISSN: 0378-7788. DOI: [10.1016/j.enbuild.2016.08.011](https://doi.org/10.1016/j.enbuild.2016.08.011).

- <https://doi.org/10.1016/j.enbuild.2016.07.053>. URL: <https://www.sciencedirect.com/science/article/pii/S0378778816306636>.
- Wang, Nan and Aiguo Song (2016). “Enhanced Logical Stochastic Resonance in Synthetic Genetic Networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* 27.12, pp. 2736–2739. DOI: 10.1109/TNNLS.2015.2495155.
- Chen, Wei, Aik Rui Tan, and Andrew L. Ferguson (2018). “Collective variable discovery and enhanced sampling using autoencoders: Innovations in network architecture and error function design”. In: *The Journal of Chemical Physics* 149.7, p. 072312. DOI: 10.1063/1.5023804. eprint: <https://doi.org/10.1063/1.5023804>. URL: <https://doi.org/10.1063/1.5023804>.
- Dsilva, Carmeline J., Ronen Talmon, Ronald R. Coifman, and Ioannis G. Kevrekidis (2018). “Parsimonious representation of nonlinear dynamical systems through manifold learning: A chemotaxis case study”. In: *Applied and Computational Harmonic Analysis* 44.3, pp. 759–773. ISSN: 1063-5203. DOI: <https://doi.org/10.1016/j.acha.2015.06.008>. URL: <https://www.sciencedirect.com/science/article/pii/S1063520315000949>.
- Mosheiff, Noga, Bruno M. C. Martins, Sivan Pearl-Mizrahi, Alexander Grünberger, Stefan Helfrich, Irina Mihalcescu, Dietrich Kohlheyer, James C. W. Locke, Leon Glass, and Nathalie Q. Balaban (2018). “Inheritance of Cell-Cycle Duration in the Presence of Periodic Forcing”. In: *Phys. Rev. X* 8 (2), p. 021035. DOI: 10.1103/PhysRevX.8.021035. URL: <https://link.aps.org/doi/10.1103/PhysRevX.8.021035>.
- Zhang, Linfeng, Jiequn Han, Han Wang, Roberto Car, and Weinan E (2018). “DeePCG: Constructing coarse-grained models via deep neural networks”. In: *The Journal of Chemical Physics* 149.3, p. 034101. DOI: 10.1063/1.5027645. eprint: <https://doi.org/10.1063/1.5027645>. URL: <https://doi.org/10.1063/1.5027645>.
- Alber, Mark, Adrian Buganza Tepole, William R. Cannon, Suvranu De, Salvador Dura-Bernal, Krishna Garikipati, George Karniadakis, William W. Lytton, Paris Perdikaris, Linda Petzold, and Ellen Kuhl (2019). “Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences”. In: *npj Digital Medicine* 2.1, p. 115. ISSN: 2398-6352. DOI: 10.1038/s41746-019-0193-y. URL: <https://doi.org/10.1038/s41746-019-0193-y>.
- Chen, Duxin, Yuchen Wang, Ge Wu, Mingyu Kang, Yongzheng Sun, and Wenwu Yu (2019). “Inferring causal relationship in coordinated flight of pigeon flocks”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.11, p. 113118. DOI: 10.1063/1.5120787. eprint: <https://doi.org/10.1063/1.5120787>. URL: <https://doi.org/10.1063/1.5120787>.
- Liao, Wenjing and Mauro Maggioni (2019). “Adaptive Geometric Multiscale Approximations for Intrinsically Low-dimensional Data”. In: *Journal of Machine Learning Research* 20.98, pp. 1–63. URL: <http://jmlr.org/papers/v20/17-252.html>.
- Arbabi, Hassan, Judith E. Bunder, Giovanni Samaey, Anthony J. Roberts, and Ioannis G. Kevrekidis (2020). “Linking Machine Learning with Multiscale Numerics: Data-Driven Discovery of Homogenized Equations”. In: *JOM* 72.12, pp. 4444–4457. ISSN: 1543-1851. DOI: 10.1007/s11837-020-04399-8. URL: <https://doi.org/10.1007/s11837-020-04399-8>.
- Gkeka, Paraskevi, Gabriel Stoltz, Amir Barati Farimani, Zineb Belkacemi, Michele Ceriotti, John D. Chodera, Aaron R. Dinner, Andrew L. Ferguson, Jean-Bernard Maillet, Hervé Minoux, Christine Peter, Fabio Pietrucci, Ana Silveira, Alexandre Tkatchenko, Zofia Trstanova, Rafal Wiewiora, and Tony Lelièvre (2020). “Machine Learning Force Fields and Coarse-Grained Variables in Molecular Dynamics: Application to Materials and Biological Systems”. In: *Journal of Chemical Theory and Computation* 16.8, pp. 4757–4775. DOI: 10.1021/acs.jctc.0c00355. eprint: <https://doi.org/10.1021/acs.jctc.0c00355>. URL: <https://doi.org/10.1021/acs.jctc.0c00355>.



- Lee, Seungjoon, Mahdi Kooshkbaghi, Konstantinos Spiliotis, Constantinos I. Siettos, and Ioannis G. Kevrekidis (2020). "Coarse-scale PDEs from fine-scale observations via machine learning". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 30.1, p. 013141. DOI: 10.1063/1.5126869. eprint: <https://doi.org/10.1063/1.5126869>. URL: <https://doi.org/10.1063/1.5126869>.
- Li, Ying, William J. Schwartz, and Premananda Indic (2020). "Dynamics of periodically forced finite N-oscillators, with implications for the social synchronization of animal rest-activity rhythms". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 30.10, p. 103106. DOI: 10.1063/5.0019848. eprint: <https://doi.org/10.1063/5.0019848>. URL: <https://doi.org/10.1063/5.0019848>.
- Qi, Di and Andrew J. Majda (2020). "Using machine learning to predict extreme events in complex systems". In: *Proceedings of the National Academy of Sciences* 117.1, pp. 52–59. DOI: 10.1073/pnas.1917285117. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.1917285117>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.1917285117>.
- Thiem, Thomas N., Mahdi Kooshkbaghi, Tom Bertalan, Carlo R. Laing, and Ioannis G. Kevrekidis (2020). "Emergent Spaces for Coupled Oscillators". In: *Frontiers in Computational Neuroscience* 14. ISSN: 1662-5188. DOI: 10.3389/fncom.2020.00036. URL: <https://www.frontiersin.org/articles/10.3389/fncom.2020.00036>.
- Fernex, Daniel, Bernd R. Noack, and Richard Semaan (2021). "Cluster-based network modeling: From snapshots to complex dynamical systems". In: *Science Advances* 7.25, eabf5006. DOI: 10.1126/sciadv.abf5006. eprint: <https://www.science.org/doi/pdf/10.1126/sciadv.abf5006>. URL: <https://www.science.org/doi/abs/10.1126/sciadv.abf5006>.
- Ye, Felix X. F., Sichen Yang, and Mauro Maggioni (2021). *Nonlinear model reduction for slow-fast stochastic systems near manifolds*. DOI: 10.48550/ARXIV.2104.02120. URL: <https://arxiv.org/abs/2104.02120>.
- Jia, Weikuan, Meili Sun, Jian Lian, and Sujuan Hou (2022). "Feature dimensionality reduction: a review". In: *Complex & Intelligent Systems* 8.3, pp. 2663–2693. ISSN: 2198-6053. DOI: 10.1007/s40747-021-00637-x. URL: <https://doi.org/10.1007/s40747-021-00637-x>.
- Kemeth, Felix P., Tom Bertalan, Thomas Thiem, Felix Dietrich, Sung Joon Moon, Carlo R. Laing, and Ioannis G. Kevrekidis (2022). "Learning emergent partial differential equations in a learned emergent space". In: *Nature Communications* 13.1, p. 3318. ISSN: 2041-1723. DOI: 10.1038/s41467-022-30628-6. URL: <https://doi.org/10.1038/s41467-022-30628-6>.
- Vlachas, Pantelis R., Julija Zavadlav, Matej Praprotnik, and Petros Koumoutsakos (2022). "Accelerated Simulations of Molecular Systems through Learning of Effective Dynamics". In: *Journal of Chemical Theory and Computation* 18.1, pp. 538–549. DOI: 10.1021/acs.jctc.1c00809. eprint: <https://doi.org/10.1021/acs.jctc.1c00809>. URL: <https://doi.org/10.1021/acs.jctc.1c00809>.
- Aristotle (350BCE). *Metaphysics*. Ed. by Translated by W. D. Ross. The Internet Classics Archive. URL: <http://classics.mit.edu/Aristotle/metaphysics.html>.
- Multiscale Systems Biology and Modeling, howpublished* = [https://researcher.watson.ibm.com/researcher/view\\_group.php?id=5372](https://researcher.watson.ibm.com/researcher/view_group.php?id=5372), note = Accessed: 2022-07-09 (n.d.).

## Chapter 2

# Algorithms

The main algorithms used in the rest of this thesis are presented in this chapter. They are organized in two categories: Numerical methods (Subchapter 2.1) and Machine Learning (Subchapter 2.2). For each category, the algorithms are presented in an increasing order of complexity of the computational task. Specifically:

**Numerical Methods:** Starting in Subchapter 2.1.1 algorithms are presented that solve for periodic steady states (limit cycles). The two Subchapters (2.1.1.1, 2.1.1.2) explore two different methodologies for the same purpose. Following that, in Subchapter 2.1.2 numerical continuation will be discussed and specifically pseudo-arclength continuation. This algorithm explores solution branches of periodic steady states, using the solvers presented in 2.1.1.

**Machine Learning:** Following the established classification of Machine Learning Algorithms, the first Subchapter deals with Supervised learning 2.2.1 i.e. algorithms for labeled data, suitable for system identification. Here, these include feedforward Artificial Neural Networks (possibly Deep) (2.2.1.2) and Gaussian Process Regression (2.2.1.1). Naturally, Unsupervised learning follows (2.2.2), i.e. algorithms for unlabeled data. These algorithms are useful for revealing intrinsic dimensionalities in data sets and finding data-driven coordinates for low-dimensional embeddings. The one used in this thesis are Diffusion Maps (2.2.2.1). Lastly, Active Learning (2.2.3) is included as a separate category of Machine Learning algorithms, even though all data are labeled. The difference is that data are *sequentially*

sampled, and their sampling is conditioned on all previous data points. In this category we include Bayesian Continuation 2.2.3.2 and Bayesian Optimization 2.2.3.1.

## 2.1 Numerical Methods

### 2.1.1 Solvers for Periodic Steady States

As mentioned in Subchapter 1.1 periodic steady states (limit cycles) can be found by locating fixed points of the stroboscopic map (Eq.1.5) or, equivalently, roots of its residual function (Eq.1.6). For this task, Newton's method can be employed, provided access to the Jacobian of the stroboscopic map (or residual function):

$$\mathbf{J}_R(\mathbf{x}_n) \cdot (\mathbf{x}_{n+1} - \mathbf{x}_n) = -\mathbf{R}(\mathbf{x}_n), \quad (2.1)$$

where, as a reminder,  $\mathbf{S} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is the stroboscopic map,  $\mathbf{R} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is the residual function, and  $\mathbf{J}_S : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ ,  $\mathbf{J}_R : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$  are their Jacobians respectively. Note that because  $\mathbf{R}(\mathbf{x}) = \mathbf{x} - \mathbf{S}(\mathbf{x})$ , it follows that  $\mathbf{J}_R = \mathbf{I}_n - \mathbf{J}_S$ , where  $\mathbf{I}_n$  is the  $n \times n$  identity matrix.

In the following two Subchapters (2.1.1.1 and 2.1.1.2) algorithms are presented that either calculate that Jacobian, or circumvent the need for its calculation in general.

#### 2.1.1.1 Variational approach

As a reminder from Subchapter 1.1.1, we will consider a generic dynamical system:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; \mathbf{p}), \quad (2.2)$$

where  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{p} \in \mathbb{R}^m$ ,  $\mathbf{f} : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^n$ . Consider  $\mathbf{J}_f(\mathbf{x}; \mathbf{p}) := \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \end{bmatrix} \in \mathbb{R}^{n \times n}$  the Jacobian of this vectorfield and  $\mathbf{J}_{fp}(\mathbf{x}; \mathbf{p}) := \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \mathbf{p}} \end{bmatrix} \in \mathbb{R}^{n \times m}$  the parametric derivatives matrix. We can compute the Jacobian of the stroboscopic map and its parametric derivatives (sensitivities), by integrating the

additional ODEs (Kevrekidis, Schmidt, and Aris, 1986):

$$\frac{d\mathbf{V}}{dt} = \mathbf{J}_f \cdot \mathbf{V} \quad (2.3)$$

$$\frac{d\mathbf{P}}{dt} = \mathbf{J}_f \cdot \mathbf{P} + \mathbf{J}_{fp}, \quad (2.4)$$

where  $\mathbf{V}(t) := \left[ \frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(0)} \right] \in \mathbb{R}^{n \times n}$ ,  $\mathbf{V}(0) = \mathbf{I}_n$  and  $\mathbf{P}(t) := \left[ \frac{\partial \mathbf{x}(t)}{\partial \mathbf{p}} \right] \in \mathbb{R}^{n \times m}$ ,  $\mathbf{P}(0) = \mathbf{0}_{n \times m}$ . Note that this is a *matrix* system of ODEs, totaling  $n^2$  and  $nm$  coupled ODEs respectively. The variables  $\mathbf{V}(t), \mathbf{P}(t)$  can be interpreted as the sensitivities of a *trajectory* of Eq.2.2 beginning from  $\mathbf{x}(0) = \mathbf{x}_0$  w.r.t  $\mathbf{x}_0$  or  $\mathbf{p}$  respectively.

By integrating 2.2 along with 2.3, 2.4 we can find:

$$\mathbf{J}_S(\mathbf{x}_n) = \frac{\partial \mathbf{S}(\mathbf{x}_n)}{\partial \mathbf{x}_n} = \mathbf{V}(T), \quad \mathbf{J}_R(\mathbf{x}_n) = \mathbf{I}_n - \mathbf{V}(T) \quad (2.5)$$

$$\frac{\partial \mathbf{S}(\mathbf{x}_n)}{\partial \mathbf{p}} = \mathbf{P}(T), \quad (2.6)$$

where  $T = \omega/2\pi$  is the forcing period associated with the stroboscopic map (see Eq. 1.4). Here, we assume that the forcing period is independent of the parameters  $\mathbf{p}$  or the initial state  $\mathbf{x}(0)$ .

With the Jacobian of the stroboscopic map at hand (Eq. 2.5), we can perform Newton's method (Eq. 2.1) and locate limit cycles given a good enough initial condition. Furthermore, calculation of the parametric derivatives allows numerical continuation of the fixed points of the stroboscopic map (for more details see Subchapter 2.1.2). Note that integration of the system of ODEs consisting of Eqs. 2.2, 2.3, 2.4 can be done with implicit integrators, as the Jacobian of the entire system conveniently consists of  $1 + n + m$  diagonal blocks of  $\mathbf{J}_f$ .

### 2.1.1.2 Newton-Krylov GMRES

It becomes apparent that in high dimensions ( $n \gg 1$ ) the Variational approach (Subchapter 2.1.1.1) becomes impractical (both in terms of efficiency and computational memory). In fact, independently of the way it's calculated, the Jacobian used in Newton's method (of size  $\mathbb{R}^{n \times n}$ ) requires a lot of computational memory and prohibits efficient implementation in high-dimensional systems. This can be circumvented via *matrix-free* methods, and specifically Newton-Krylov GMRES (Kelley, Kevrekidis, and Qiao, 2004). This method has the following essential components:

- Estimating the matrix-vector product in the left-hand side of Eq.2.1 using directional derivatives:

$$\mathbf{J}_R(\mathbf{x}_n) \cdot \mathbf{y} \approx \frac{\mathbf{R}\left(x_n + \epsilon \frac{\mathbf{y}}{\|\mathbf{y}\|_2}\right) - \mathbf{R}(\mathbf{x}_n)}{\epsilon}, \quad (2.7)$$

where , in this case,  $\mathbf{y} = \mathbf{x}_{n+1} - \mathbf{x}_n$  and  $\epsilon$  is a user-defined sufficiently small scalar.

- Solving any linear system  $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$  by projecting the residual to the Krylov subspace of  $\mathbf{A}$ , defined by  $\mathbf{A}$ ,  $\mathbf{b}$  and subspace dimension  $d$ :  $K(\mathbf{A}, \mathbf{b}, d) = \text{span} \left\{ \mathbf{b}, \mathbf{A}\mathbf{b}, \dots, \mathbf{A}^{d-1}\mathbf{b} \right\}$  (Kelley, 2003):

$$\mathbf{x}^* - \mathbf{x}_0 = \sum_{j=0}^{d-1} \gamma_j \mathbf{A}^j \mathbf{r}_0, \quad (2.8)$$

where  $\mathbf{x}^*$ ,  $\mathbf{x}_0$  are the solution and the initial guess respectively,  $d$  is the Krylov dimension,  $\gamma_j$  are coefficients, and  $\mathbf{r}_0$  is the initial residual of the linear system ( $\mathbf{r}_0 = \mathbf{A} \cdot \mathbf{x} - \mathbf{b}$ ). Here, a new linear system arises for every Newton iteration (Eq. 2.1.)

Combining these two components, we can efficiently implement Eq. 2.1 and locate periodic steady states in high-dimensional systems by finding fixed points of the stroboscopic map.

## 2.1.2 Numerical Continuation

With an efficient solver at hand (as, for example, the ones described in Subchapter 2.1.1) we can explore algorithms that follow entire one-parameter solution branches of periodic steady states. For that task, a single degree of freedom is chosen (some parametrization) along with an additional equation to solve. Each continuation algorithm decides how to expand a solution branch and, equivalently, what the additional equation should be. The resulting algorithm is an iterative “predictor-corrector” scheme. Here we focus on pseudo-arclength continuation which assigns the arclength  $s$  as the extra degree of freedom, expressing both the state vector and the parameter as functions of  $s$  (Keller, 1978). Advancing along a solution branch in terms of the arclength tackles a fundamental issue relevant to the Implicit Function Theorem: solution curves can be continued past direction changes w.r.t the parameter, which, in the context of dynamical systems, correspond to folds (Doedel, Keller, and Kernevez, 1991a).

In the following, the pseudo-arclength continuation algorithm will be presented for a generic dynamical system  $\frac{dx}{dt} = \mathbf{f}(\mathbf{x}; \lambda)$  where  $\mathbf{f} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ ,  $\mathbf{x} \in \mathbb{R}^n$  is the state vector and  $\lambda \in \mathbb{R}$  is a parameter. To follow steady states of that system, the pseudo-arclength algorithm seeks points that satisfy the *augmented* system:

$$\mathbf{f}(\mathbf{x}(s); \lambda(s)) = \mathbf{0} \quad (2.9)$$

$$(\mathbf{x}(s) - \mathbf{x}_0)^T \dot{\mathbf{x}}_0 + (\lambda(s) - \lambda_0) \dot{\lambda}_0 = s - s_0, \quad (2.10)$$

where  $\dot{\mathbf{x}}_0, \dot{\lambda}_0$  are the direction vectors with respect to the arclength  $s$  (i.e.  $d\mathbf{x}/ds, d\lambda/ds$  respectively) at some point of the solution branch, where:  $\mathbf{x}_0 = \mathbf{x}(s_0), \lambda_0 = \lambda(s_0)$ . The direction vectors can be found by solving:

$$\mathbf{J}_f \dot{\mathbf{x}} + \left[ \frac{\partial f}{\partial \lambda} \right] \dot{\lambda} = \mathbf{0} \quad (2.11)$$

$$\|\dot{\mathbf{x}}\|^2 + \dot{\lambda}^2 = 1, \quad (2.12)$$

where  $\mathbf{J}_f \in \mathbb{R}^{n \times n}$  is the Jacobian of  $\mathbf{f}$  and  $\left[ \frac{\partial f}{\partial \lambda} \right] \in \mathbb{R}^{n \times 1}$  its parametric derivatives. Intuitively, one can think of Eq.2.11 as a chain rule and Eq.2.12 as normalizing the direction vector.

Consider a solution branch of points  $(\mathbf{x}_i, \lambda_i)$  where  $i = 1, 2, \dots$  is an index. A very convenient simplification for direction vectors at some point  $(\mathbf{x}_i, \lambda_i)$  is to approximate them by  $\dot{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{x}_{i-1}, \dot{\lambda}_i = \lambda_i - \lambda_{i-1}$ . Using this simplification, it is no longer necessary to solve Eqs.2.11, 2.12, or, consequently, to have the Jacobian and parametric derivatives at hand. When solving the system Eq.2.9, 2.10 (e.g. with Newton's method) convergence strongly depends on the initial guess. A good such guess (a good predictor) can be intuitively formulated as  $(\mathbf{x}_{i+1}^{pred}, \lambda_{i+1}^{pred}) = (\mathbf{x}_i + \Delta s \dot{\mathbf{x}}_i, \lambda_i + \Delta s \dot{\lambda}_i)$ , where  $\Delta s$  is a stepsize.

For a summary of the most simplified iterative scheme of pseudo-arclength continuation, see the following algorithm (Alg. 1). Some practical considerations: (i) the stepsize  $\Delta s$  is usually adaptive to the number of Newton iterations required for convergence, (ii) If the Eqs. (2.11, 2.12) are solved instead of the linear approximation to the direction vector, then the algorithm can be initialized with just one point (but the initial direction needs to be specified).

---

**Algorithm 1** Pseudo-arclength continuation

---

**Input:** Two consecutive points on the solution branch:  $(\mathbf{x}_1, \lambda_1), (\mathbf{x}_2, \lambda_2)$ , Number of iterations  $N_{it} > 2$ , Step size  $\Delta s$ .

**Output:** Continuation branch  $[\mathbf{x}_i, \lambda_i]_{i=1, \dots, N_{it}}$ .

**for**  $i = 2$  to  $N_{it} - 1$  **do**

$(\dot{\mathbf{x}}_i, \dot{\lambda}_i) \leftarrow (\mathbf{x}_i - \mathbf{x}_{i-1}, \lambda_i - \lambda_{i-1})$ , or solve 2.12

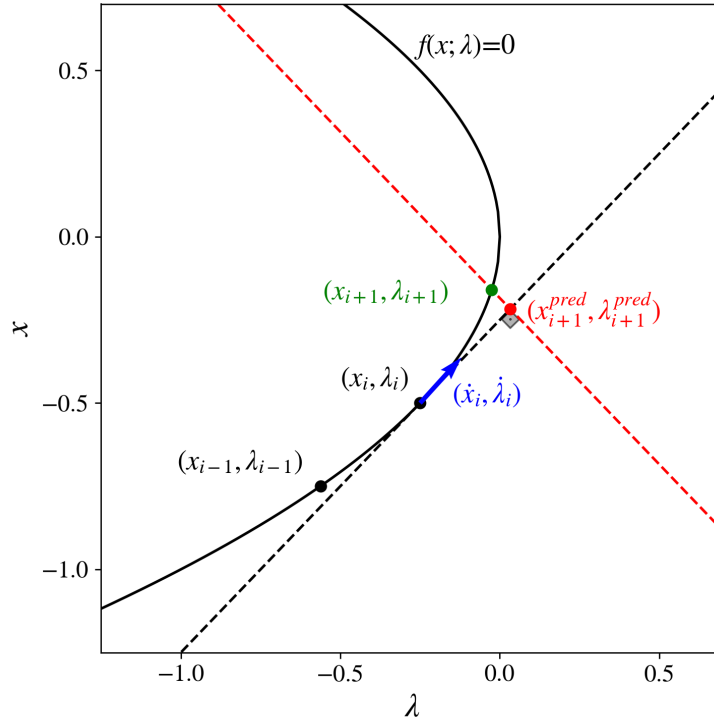
$(\mathbf{x}_{i+1}^{pred}, \lambda_{i+1}^{pred}) \leftarrow (\mathbf{x}_i + \Delta s \dot{\mathbf{x}}_i, \lambda_i + \Delta s \dot{\lambda}_i)$

$(\mathbf{x}_{i+1}^{corr}, \lambda_{i+1}^{corr}) \leftarrow (\mathbf{x}_{i+1}^{pred}, \lambda_{i+1}^{pred})$ , by solving the system, 2.9, 2.10 with Newton's method

**end for**

---

A schematic of an instance of pseudo-arclength procedure can be seen in Fig. 2.1.



**Figure 2.1:** Pseudo-arclength continuation iteration for the steady states of  $dx/dt = f(x; \lambda) := \lambda + x^2$ : Given the direction vector (here, colored blue) the predictor for the next solution on the branch can be found (colored red). Solving the augmented system of equations (such as Eq. 2.9, 2.10) will restrict the next solution (colored green) on a plane perpendicular to the direction vector.

In the context of periodically forced systems, continuation is especially useful in exploring the

bifurcation diagram of periodic responses w.r.t. the forcing parameters (for an example, see Fig. 1.2). In this case, the solution branches to be continued are those of the stroboscopic map residual function and the parameter is one of the forcing parameters. In the notation used in Eqs. 1.1-1.6, the pseudo-arclength augmented system will be:

$$\mathbf{R}(\mathbf{x}_i; \lambda_i) = \mathbf{0} \quad (2.13)$$

$$(\mathbf{x}_i - \mathbf{x}_{i-1})^T \dot{\mathbf{x}}_{i-1} + (\lambda_i - \lambda_{i-1}) \dot{\lambda}_{i-1} = \Delta s \quad (2.14)$$

where  $\lambda$  is some forcing parameter (i.e.  $\omega$  or a component of  $\mathbf{p}_f$ ). The Jacobian and parametric derivatives required in each continuation step (specifically, in Eqs. 2.11, 2.12) and in Newton’s method (Eq. 2.1) can be calculated through the variational approach (Eqs. 2.5, 2.6). Note that here, it is possible that the forcing period depends on the continuation parameter. This would affect the final expression for trajectory sensitivities (Eq. 2.6) which should be altered to include the sensitivity of the integration limits to the forcing parameter, according to Leibniz’s integral rule. For example, this is the *de facto* case when performing the same tasks for Poincaré maps (see Eqs. 1.2, 1.3). Again, the need for the Jacobian and parametric derivatives in the pseudo-arclength continuation algorithm can be circumvented by approximating the direction vectors from the previous points on the solution branch.

Despite the simplicity in its formulation, pseudo-arclength continuation can be augmented/modified to perform many useful tasks in practice (adaptive stepping, bifurcation detection, branch point detection, stability analysis, and more). Importantly, in the context of this work, it can perform 2–parameter continuation (for example, to construct resonance horns, as in Fig. 1.2). This requires solving an additional equation, constraining the solution to the loci of points of interest (e.g. some constraint on the eigenvalues when continuing folds). In practice, this has been demonstrated by various established software implementations, such as: AUTO (Doedel, Keller, and Kernevez, 1991a; Doedel, Keller, and Kernevez, 1991b; Doedel et al., 2007), MATCONT (Dhooge, Govaerts, and Kuznetsov, 2003), xppaut (Ermentrout, 2012) and COCO (Dankowicz and Schilder, 2013).

## 2.2 Machine Learning

In general, Machine Learning includes algorithms that are able to *learn* from data (Goodfellow, Bengio, and Courville, 2016). A general enough definition of *learning* has been given by Mitchell (1997):



“A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks  $T$ , as measured by  $P$ , improves with experience  $E$ ” (Mitchell, 1997).

Machine Learning algorithms can be organized based on the kind of task  $T$  (discrete or continuous) or experience  $E$  (labeled or unlabeled data). The Machine Learning algorithms relevant to this dissertation, deal with tasks in the form of continuous functions/mappings. We organize this section as learning from labeled data (**supervised learning**, Subchapter 2.2.1) or unlabeled data (**unsupervised learning**, Subchapter 2.2.2). To this organization, we also include **active learning** (Subchapter 2.2.3) where the Machine Learning algorithm interactively samples a system to gather and label new data (Settles, 2009). This is juxtaposed to supervised/unsupervised learning where the data set is fixed prior to learning.

## 2.2.1 Supervised Learning

As mentioned in the introduction of Subchapter 2.2 supervised learning deals with the learning of continuous functions from labeled data.

Specifically we will consider input data in the form of a design matrix  $\mathbf{X} \in \mathbb{R}^{n \times m}$ , where  $n$  is the number of samples and  $m$  the number of input features. The output data (or labels) are denoted as  $\mathbf{Y} \in \mathbb{R}^{n \times k}$  where  $k$  is the number of target features. The objective is to learn (fit) a target function  $\hat{\mathbf{f}} : \mathbb{R}^m \rightarrow \mathbb{R}^k$ , which is considered an approximation of a ground-truth, underlying function  $\mathbf{f}$ . The goodness of fit is usually measured by a loss function which quantifies the proximity of  $\mathbf{y}_i$  to  $\hat{\mathbf{y}}_i := \hat{\mathbf{f}}(\mathbf{x}_i)$ , where  $\mathbf{x}_i, \mathbf{y}_i$  are the  $i$ -th rows of  $\mathbf{X}, \mathbf{Y}$  respectively.

In the following, two supervised learning algorithms will be presented, used in this dissertation: Gaussian Process Regression and Artificial Neural Networks.

### 2.2.1.1 Gaussian Process Regression

An algorithm that learns a target function from labeled data is Gaussian Process Regression (GPR). GPR assumes that the (here, assumed one-dimensional) target function  $\hat{f}(\mathbf{x}), \hat{f} : \mathbb{R}^m \rightarrow \mathbb{R}$  is distributed according to a Gaussian process, which can be fully specified by its mean function  $m(\mathbf{x})$  and covariance function  $k(\mathbf{x}, \mathbf{x}')$  (Rasmussen and Williams, 2006):

$$\hat{f}(\mathbf{x}) \sim \mathcal{N}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

This can be understood as setting a Gaussian prior distribution over the space of functions. The mean is usually set to zero by centering the data during preprocessing. The covariance function  $k(\mathbf{x}, \mathbf{x}')$  is commonly formulated by a Euclidean-distance kernel function in the input space (Lee et al., 2020). A popular choice is the Matèrn32 kernel with a constant:

$$k(\mathbf{x}_i, \mathbf{x}_j) = c \left( 1 + \sqrt{3}d(\mathbf{x}_i, \mathbf{x}_j; \mathbf{l}) \right) e^{-(\sqrt{3}d(\mathbf{x}_i, \mathbf{x}_j; \mathbf{l}))}, \quad (2.15)$$

$$d(\mathbf{x}_i, \mathbf{x}_j; \mathbf{l}) = \sqrt{\sum_{k=1}^n \frac{(x_{ik} - x_{jk})^2}{l_k}}, \quad (2.16)$$

where  $\mathbf{x}_i, \mathbf{x}_j$  are any two input feature vectors (rows of a design matrix  $\mathbf{X}$ ),  $c$  is a scalar,  $\mathbf{l}$  is a vector with number of entries equal to the dimension of the input space.  $c$  and  $\mathbf{l}$  are the hyperparameters to be optimized (here, collectively denoted  $\boldsymbol{\theta}$ ).

The case of noisy observations is usually considered, assuming  $y = f(\mathbf{x}) + \epsilon$ , where  $\epsilon \sim \mathcal{N}(0, \sigma^2)$  is i.i.d. Gaussian additive noise with known variance. Given a data set  $(\mathbf{X}, \mathbf{Y})$  the optimal hyperparameter vector  $\boldsymbol{\theta}^*$  is the maximum likelihood estimator :

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \{-\log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta})\}$$

This estimator defines the posterior Gaussian Process given  $(X, Y)$ . To find the Gaussian distribution of the function values at test data points, we represent the multivariate Gaussian distribution with the covariance matrix whose entries are given by a function such as Eq. 2.15:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}^* \end{bmatrix} \sim N \left( \mathbf{0}, \begin{bmatrix} \mathbf{K} + \sigma^2 I & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{bmatrix} \right),$$

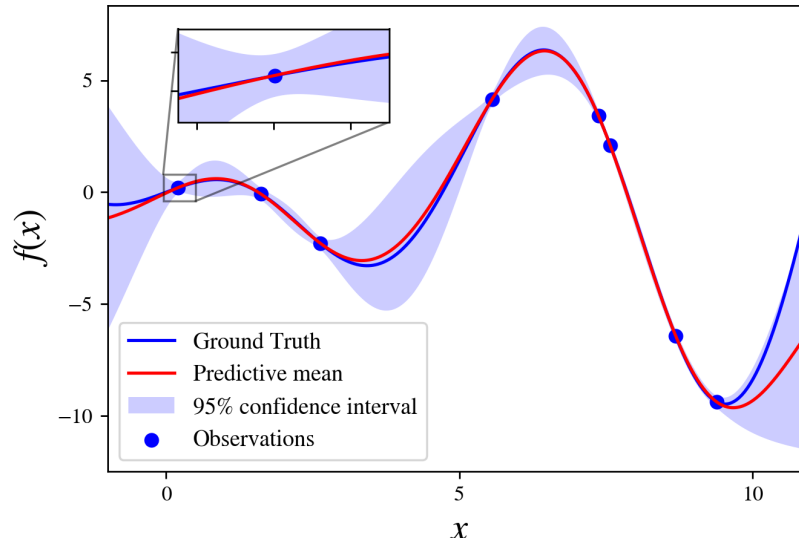
where  $\mathbf{y}^*$  is a predictive distribution for test data  $\mathbf{x}^*$ ,  $\mathbf{K}_*$  represents a covariance matrix between training and test data while  $\mathbf{K}_{**}$  represents a covariance matrix between test data. Finally, we represent a Gaussian distribution for the target function at the test points in terms of a predictive mean and its variance, by conditioning a multivariate Gaussian distribution:

$$\hat{\mathbf{y}}^* = \mathbf{K}_*(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y} \quad (2.17)$$

$$\mathbf{K}(\mathbf{y}^*) = \mathbf{K}_{**} - \mathbf{K}_*^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_*,$$

and we assign the predictive mean ( $\hat{\mathbf{y}}^*$ ) as the estimated target function for the corresponding data point. All components of Gaussian Process Regression on a simple example can be seen on Fig. 2.2.

As shown in Eq.(2.16), a large magnitude of the hyperparameter  $l_k$  would nullify the contribution along that direction to the distance metric (in the numerator), leading to the relative insignificance (low sensitivity) of that input feature towards the predictions. In the Gaussian Process literature (Neal, 1996), this is called “Automatic Relevance Determination”. One can a posteriori disregard inputs with too low relevance (large  $l_k$ ) in order to learn simpler models, with only the important input features.



**Figure 2.2:** An example of Gaussian Process Regression (GPR), here learning from 10 randomly sampled points of the ground truth function  $f(x) = x \cos(x)$ . The kernel used is Gaussian and the assumed additive noise variance  $\sigma^2 = 10^{-3}$  (the effect of which can be better seen at the inset). GPR provides not only a prediction, but quantifies uncertainty around it (here, the 95% confidence interval is shown). In fact, GPR returns the best fitted posterior *distribution* of output values in the input space. As expected, the uncertainty grows where observations are sparse, especially at the boundaries of the input space.

### 2.2.1.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are a family of functions constructed by composing many affine and nonlinear elementary functions (activation functions). In (feed-forward) neural networks, a universal approximation theorem (Cybenko, 1989; Chen and Chen, 1995) guarantees that for a single hidden layer with (sufficient) finite number of neurons, an approximation  $\hat{y} := \hat{f}(\mathbf{x})$  of the unknown target function,  $y := f(\mathbf{x})$  can be found for some  $\mathbf{x} \in \mathbb{R}^m$ . Here, approximation implies that the target and learned functions are sufficiently close in an appropriately chosen norm: for all  $\epsilon > 0$  there exists an ANN predicting  $\hat{f}(\mathbf{x})$ , where  $|y - \hat{f}(\mathbf{x})| < \epsilon$  for all  $\mathbf{x} \in A$  and some  $A \subseteq \mathbb{R}^m$ . The approximate form of the target function obtained through the feedforward neural net (here, with a single layer) can be written as:

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{N_n} \psi(\mathbf{w}_i^T \mathbf{x} + \mathbf{b}_i),$$

where  $\psi$  is a nonlinear activation function,  $\mathbf{w}_i, \mathbf{b}_i$  are tunable parameters (weights and biases) and  $N_n$  is the number of neurons, which is decided a priori. To find optimal weights and biases, an optimizer is used (employing a backpropagation scheme) to minimize the root-mean-square error cost function:

$$E_D = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(\mathbf{x}_i))^2,$$

which typically measures the goodness of the approximation. Intuitively, the success of Neural Networks in approximating highly nonlinear functions, can be understood geometrically: the addition of optimized nonlinear activation functions (e.g. sigmoids) creates a series of appropriately positioned “bumps” in the input space, which, in principle can approximate any surface (Lapedes and Farber, 1987). For multiple layers of neurons, the neuronal network is called “deep” (Goodfellow, Bengio, and Courville, 2016).

## 2.2.2 Unsupervised Learning

Unsupervised learning is used to discover useful structure in unlabeled data. Such structure can inform us of a reduced, latent (intrinsic) dimensionality (Goodfellow, Bengio, and Courville, 2016). It can also be used to construct mappings between the original (full, ambient) description and the reduced (latent) one.

As in the supervised case (Subchapter 2.2.1), the input data will be in the form of a design matrix

$\mathbf{X} \in \mathbb{R}^{n \times m}$ , where  $n$  is the number of samples and  $m$  the number of input features. Unsupervised learning describes the mapping  $\mathbf{g} : \mathbb{R}^m \rightarrow \mathbb{R}^d$ , where  $d \leq m$ . This can be thought as a change of coordinates or an encoding. Dimensionality reduction is achieved when  $d < m$ , yet other useful tasks can also be achieved, such as manifold learning, clustering or denoising (Lee and Verleysen, 2007).

### 2.2.2.1 Diffusion Maps

Diffusion maps (Coifman and Lafon, 2006; Nadler et al., 2006; Nadler et al., 2005) is a spectral nonlinear dimensionality reduction technique based on learning the geometry of an underlying manifold from sampled data. Fundamental to this method, is the construction of a random walk on the data. To that end, transition probabilities are calculated using an appropriate kernel and similarity measure. The Gaussian kernel along with the Euclidean distance are popular choices; the (symmetric) transition probability between data points indexed  $i, j$  will then be proportional to:

$$w_{ij} = \exp \left[ - \left( \frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\epsilon} \right)^2 \right],$$

where  $\mathbf{x}_i$  (and  $\mathbf{x}_j$ )  $\in \mathbb{R}^m$  for some  $i$  (and  $j$ ) =  $1, \dots, n$  and  $\epsilon$  a scale hyperparameter which quantifies the local similarity for each data point (kernel bandwidth). Calculating the entries  $w_{ij}$ , the symmetric weights matrix  $\mathbf{W} \in \mathbb{R}^{n \times n}$  can be constructed. To remove the effect of the sampling density, the normalization

$$\widetilde{\mathbf{W}} = \mathbf{R}^{-\alpha} \mathbf{W} \mathbf{R}^{-\alpha}$$

is performed, where  $\mathbf{R}$  is a diagonal matrix with entries  $R_{ii} = \sum_{j=1}^n W_{ij}$  and  $\alpha = 1$  to factor out the density effects. A second normalization applied on  $\widetilde{\mathbf{W}}$ ,

$$\mathbf{K} = \mathbf{D}^{-1} \widetilde{\mathbf{W}}$$

gives a Markov matrix  $\mathbf{K}$ ; where  $\mathbf{D}$  is a diagonal matrix, collecting the row sums of matrix  $\widetilde{\mathbf{W}}$ . It can be shown (Coifman and Lafon, 2006) that the matrix  $\frac{\mathbf{I} - \mathbf{K}}{\epsilon}$  (the infinitesimal generator of the Markov Chain) is a discrete approximation of the Laplacian operator  $\Delta$  on the underlying manifold.

The stochastic matrix  $\mathbf{K}$  has a set of real eigenvalues  $1 = \lambda_1 \geq \dots \geq \lambda_n$  with corresponding eigenvectors  $\phi_i, i = 1, \dots, n$ . The  $n$ -dimensional representation of a particular  $m$ -dimensional ambient vector

$\mathbf{x}_i$ , is given by:

$$\Psi_n(\mathbf{x}_i) = [\lambda_1^t \phi_{i,1}, \lambda_2^t \phi_{i,2}, \dots, \lambda_n^t \phi_{i,n}]$$

Note that not all of the above eigendirections are independent, (therefore, informative). In fact, the independent eigenvectors are not even necessarily the ones with the leading eigenvalues (as, for example, in Principal Component Analysis, the linear counterpart). The  $d$  **independent** eigenvectors can be found by local linear regression (Dsilva et al., 2018). The final  $d$ -dimensional representation can be thought as a change of coordinates from  $\mathbb{R}^m$  to  $\mathbb{R}^d$ .

For more details about the algorithm presented here, please refer to the manuscript (to be submitted) “From partial data to out-of-sample parameter and observation estimation with Diffusion Maps and Geometric Harmonics”.

### 2.2.3 Active Learning

As mentioned before (introduction of Subchapter 2.2), active learning deals with labeled data (like supervised learning), however, the data are collected *interactively*, i.e. the Machine Learning algorithm decides where to sample, as it learns (Settles, 2009). Under this general definition, we include Bayesian Optimization and Bayesian Continuation.

In the specific context of this dissertation, we will assume an  $m$ -dimensional state space  $X \subset \mathbb{R}^m$  and an  $k$ -dimensional parameter space  $P \subset \mathbb{R}^k$  and a scalar observation function on the state-space  $F : \mathbb{R}^m \rightarrow \mathbb{R}$  (here,  $F$  is deterministic, but that is not required in general). For specific parameter vectors  $\mathbf{p} \in P$  we assume that a state  $\mathbf{x} \in X$  exists for which a certain condition is satisfied (in the form of  $\mathbf{h}(\mathbf{x}; \mathbf{p}) = \mathbf{0}$ ). If such state is unique, the existence of a well-defined function  $\mathbf{x} = \mathbf{x}(\mathbf{p})$  is guaranteed, and the function  $F$  can be written as  $F(\mathbf{x}) = F(\mathbf{x}(\mathbf{p})) = F(\mathbf{p})$ . The specific goal of the algorithms of this Subchapter is to optimize  $F$ , although their usefulness is not restricted to that task.

#### 2.2.3.1 Bayesian Optimization

The optimization problem formulated in the introduction of Subchapter 2.2.3 is:

$$\mathbf{p}^* = \operatorname{argmax}_{\mathbf{p}} F(\mathbf{p}), \quad (2.18)$$

In the introduction of Subchapter 2.2.3 it is not given that  $F$  is analytical. In general, this is the case

of derivative-free optimization of an expensive, *black-box* objective function. Bayesian Optimization (BO) is a suitable algorithm for this category of problems (Kushner, 1964).

Bayesian Optimization's main ingredient is a probabilistic surrogate model supplied with a prior distribution that captures our beliefs about the behavior of the unknown objective function. This surrogate model is sequentially refined as more data are observed, via Bayesian posterior updating (Shahriari et al., 2016). Gaussian Processes (see Subchapter 2.2.1.1 for more details) have been very popular as a choice for surrogate models (Rasmussen and Williams, 2006).

To sample efficiently, BO minimizes an acquisition function to determine the best possible guess of where the optimum might lie, given all data previously sampled; this enables *active* learning. The acquisition function choice and the selection of its hyperparameters reflect the trade-off between exploration and exploitation (Brochu, Cora, and Freitas, 2010). For example, a popular choice for minimization problems is the Lower Confidence Bound acquisition function (Shahriari et al., 2016):

$$a(\mathbf{p}) = \mu(\mathbf{p}) - \beta\sigma(\mathbf{p}), \quad (2.19)$$

where  $\mu(\mathbf{p}), \sigma(\mathbf{p})$  is the mean and the standard deviation of the Gaussian Process surrogate model at  $\mathbf{p}$  and  $\beta$  is a hyperparameter. Larger values of  $\beta$  allows BO to search for optima at places with more uncertainty, while smaller values of  $\beta$  make BO more risk averse and the search is confined in the vicinity of already observed data. Convergence rates have been established for some cases of confidence bound acquisition functions and GP surrogate models (Srinivas et al., 2010). Other ubiquitous acquisition functions include Probability of Improvement and Expectation of Improvement (Brochu, Cora, and Freitas, 2010). A pseudoalgorithm for the overall iterative BO process is presented in Alg.2.

Owing to the Gaussian Process surrogate, Bayesian Optimization results in the following useful information:

- Optimum values and coordinates:  

$$F^* = \max_{\mathbf{p}} F(\mathbf{p}), \quad \mathbf{p}^* = \operatorname{argmax}_{\mathbf{p}} F(\mathbf{p}),$$
- Objective function approximation and uncertainty estimate (from the posterior output value distribution):  

$$F(\mathbf{p}) \sim GP(\mu(\mathbf{p}), k(\mathbf{p}, \mathbf{p}'))$$

In the context of complex dynamical systems (such as the one in Subchapter 4), one can think of the condition  $\mathbf{h}(\mathbf{x}; \mathbf{p}) = \mathbf{0}$  as a condition that  $\mathbf{x}$  is a steady state of some dynamical system for a parameter

vector  $\mathbf{p}$ , and the function  $F$  is a performance metric on the steady state, which we seek to optimize. Having to *solve* for the steady state (e.g. with Newton's method) prevents the derivation of an analytical function  $\mathbf{x}(\mathbf{p})$  and consequently  $F(\mathbf{p})$ , which makes the objective function a *black-box*.

---

**Algorithm 2** Bayesian Optimization - LCB

---

**Input:** Observations:  $S = [\mathbf{p}_i, F(\mathbf{p}_i)]_{i=1 \dots N_s}$ ,  $\beta$ , Termination criteria

**Output:**  $[\mathbf{p}^{opt}, F^{opt}]$

**while** Termination criteria not met **do**

Fit  $GP_F$  on  $[\mathbf{p}_i, F(\mathbf{p}_i)]_{i=1 \dots N_s}$

$\mathbf{p}^{new} \leftarrow \operatorname{argmin}_{\mathbf{p}} a(\mathbf{p})$ ,  $a(\mathbf{p}) = \mu(\mathbf{p}) - \beta\sigma(\mathbf{p})$

Append the sample  $S$  with  $[\mathbf{p}^{new}, F(\mathbf{p}^{new})]$

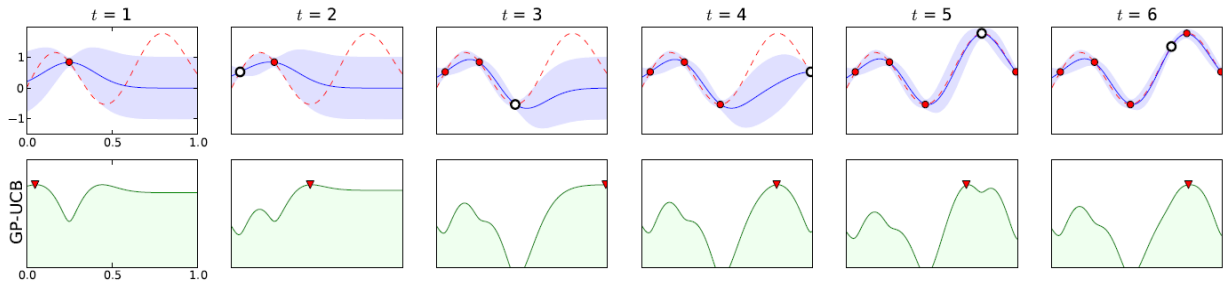
$N_s \leftarrow N_s + 1$

**end while**

$[\mathbf{p}^{opt}, F^{opt}] \leftarrow [\mathbf{p}^{new}, F(\mathbf{p}^{new})]$

---

Note, that terminating the Bayesian Optimization iterative process is an important decision. In problems with very expensive objective functions, a budget of objective function evaluations is specified. Otherwise, iterate convergence under some norm is preferred. An example of Bayesian Optimization iterates are shown in Fig.2.3.



**Figure 2.3:** An example of several Bayesian Optimization iterations borrowed from (Brochu, Cora, and Freitas, 2010). Consecutive iterates are indexed by  $t$ . At the top row, the ground truth objective function is shown (dashed red line), along with the GP predictive mean (blue line) and its uncertainty (represented by the shaded region). Current sample is denoted by red points, and the *last* point discovered as a white point. At the bottom row, the UCB acquisition function is shown (the maximizing counterpart of LCB). At each iterate, the next suggested point, will be the one maximizing the acquisition function (maxima marked by red triangles).



### 2.2.3.2 Bayesian Continuation

As mentioned in Subchapter 2.2.3.1 Bayesian Optimization constructs a surrogate model for the objective function:  $F(\mathbf{p}) \sim GP(\mu(\mathbf{p}), k(\mathbf{p}, \mathbf{p}'))$ . However, as noted in the introduction of 2.2.3 the objective function is actually calculated on the state space, where each state vector  $\mathbf{x}$  is a function of the parameter vector  $\mathbf{x}(\mathbf{p})$ . In this Subchapter we explore the benefits of constructing an *extra* surrogate model for the “solution” manifold (Renson et al., 2019), i.e. the state subspace where the condition  $\mathbf{h}(\mathbf{x}; \mathbf{p}) = \mathbf{0}$  is satisfied. State vectors in the solution manifold will be denoted  $\mathbf{x}^*(\mathbf{p}) = \{\mathbf{x} \in X : \mathbf{h}(\mathbf{x}; \mathbf{p}) = \mathbf{0}\}$ . Their (element-wise) surrogate models will have the form:

$$\mathbf{x}_i^*(\mathbf{p}) \sim GP_s(\mu(\mathbf{p}), k(\mathbf{p}, \mathbf{p}')), i = 1, \dots, m, \quad (2.20)$$

for each component  $i$  of the solution vector.

This extra surrogate model serves a very important role when solving for  $\mathbf{x}^*$  requires appropriate initialization in the state space (e.g. Newton’s method). Indeed, if Newton’s method (for example) is not appropriately initialized, convergence can be very slow, or Newton’s method might not converge at all. This hinders considerably the overall performance, as the user might have to manually try different initializations for just one calculation of the objective function. This problem is emphasized in the case of high-dimensional, nonlinear solution manifolds, where failure to converge from random initializations is the norm. However, when the solution manifold is also learned and updated iteratively through Eqs. 2.20, we can access (increasingly) good predictions of where  $\mathbf{x}^*$  might lie for a given  $\mathbf{p}$ . See the pseudo-algorithm Alg.3 for more details.

The name “**Bayesian Continuation**” (BC) is proposed for this approach, as it resembles prediction-correction schemes used in numerical continuation (Keller, 1978; Henderson, 2002), as explored in Subchapter 2.1.2.

---

**Algorithm 3** Bayesian Continuation for Bayesian Optimization

---

**Input:** Observations:  $S = [\mathbf{p}_i, \mathbf{x}_i^*(\mathbf{p}_i), F(\mathbf{p}_i)]_{i=1\dots N_s}$ , acquisition function  $a(\mathbf{p})$ , Termination criteria

**Output:**  $[\mathbf{p}^{opt}, \mathbf{x}^{opt}, F^{opt}]$

**while** Termination Criteria not met **do**

Fit  $GP_F$  on  $[\mathbf{p}_i, F(\mathbf{p}_i)]_{i=1\dots N_s}$

Fit  $GP_x$  on  $[\mathbf{p}_i, \mathbf{x}_i^*(\mathbf{p}_i)]_{i=1\dots N_s}$

$\mathbf{p}^{new} \leftarrow \operatorname{argmin}_{\mathbf{p}} a(\mathbf{p})$ , for  $GP_F$

$\mathbf{x}^{pred} \leftarrow GP_x(\mathbf{p}^{new})$

$\mathbf{x}^*(\mathbf{p}^{new}) \leftarrow \mathbf{x}^{pred}, \mathbf{p}^{new}$ , with Newton's method

$F(\mathbf{p}^{new}) \leftarrow \mathbf{x}^*(\mathbf{p}^{new})$

Append the sample  $S$  with  $[\mathbf{p}^{new}, \mathbf{x}^{new}, F(\mathbf{p}^{new})]$

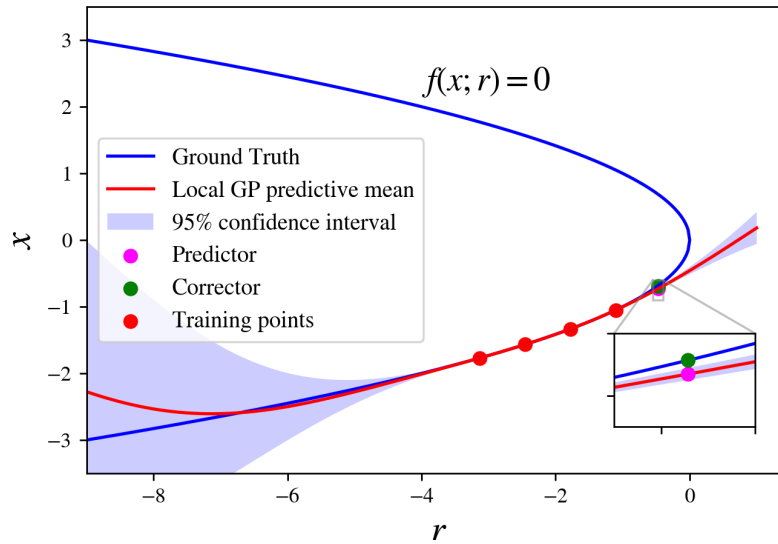
$N_s \leftarrow N_s + 1$

**end while**

$[\mathbf{p}^{opt}, \mathbf{x}^{opt}, F^{opt}] \leftarrow [\mathbf{p}^{new}, \mathbf{x}^{new}, F(\mathbf{p}^{new})]$

---

The merits of Bayesian Continuation are not restricted to accelerating Bayesian Optimization. This algorithm, when applied sequentially, can be utilized as an uncertainty-driven, Machine Learning-enabled continuation method (see Fig.2.4 for an example of an implementation instance). We propose that Bayesian Continuation can be useful for the continuation/manifold completion of *surrogate* functions from black-box simulations or experimental observations. More details about the algorithms presented here can be found in the manuscript (to be submitted): “Bayesian Continuation for Bayesian Optimization”.



**Figure 2.4:** An example of a Bayesian Continuation instance for one-parameter continuation of roots of  $f(x; r) = r + x^2$ . Here, a Gaussian Process fit is performed *locally* on the last few points on the solution curves. This fit provides predictors (in the form of GP predictive means) of where the next solution point might lie. As can be seen in the inset, the predictor is close enough to the actual solution (corrector). This method can be adaptive, exploiting uncertainty estimates of the local fit (like the 95% confidence interval shown here). Automatic Relevance Determination (see Subchapter 2.2.1.1) can also be exploited to overcome turning points.

# References

- Kushner, H. J. (1964). "A New Method of Locating the Maximum Point of an Arbitrary Multipipeak Curve in the Presence of Noise". In: *Journal of Basic Engineering* 86.1, pp. 97–106. ISSN: 0021-9223. DOI: 10.1115/1.3653121. eprint: <https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/86/1/97/5763745/97\1.pdf>. URL: <https://doi.org/10.1115/1.3653121>.
- Keller, Herbert B. (1978). "Global Homotopies and Newton Methods". In: *Recent Advances in Numerical Analysis*. Ed. by CARL DE BOOR and GENE H. GOLUB. Academic Press, pp. 73–94. ISBN: 978-0-12-208360-0. DOI: <https://doi.org/10.1016/B978-0-12-208360-0.50009-7>. URL: <https://www.sciencedirect.com/science/article/pii/B9780122083600500097>.
- Kevrekidis, I.G., L.D. Schmidt, and R. Aris (1986). "Some common features of periodically forced reacting systems". In: *Chemical Engineering Science* 41.5, pp. 1263–1276. ISSN: 0009-2509. DOI: [https://doi.org/10.1016/0009-2509\(86\)87099-3](https://doi.org/10.1016/0009-2509(86)87099-3). URL: <https://www.sciencedirect.com/science/article/pii/0009250986870993>.
- Lapedes, Alan and Robert Farber (1987). "How Neural Nets Work". In: *Neural Information Processing Systems*. Ed. by D. Anderson. Vol. 0. American Institute of Physics. URL: <https://proceedings.neurips.cc/paper/1987/file/093f65e080a295f8076b1c5722a46aa2-Paper.pdf>.
- Cybenko, G. (1989). "Approximation by Superpositions of a Sigmoidal Function". In: Doedel, Eusebius, Herbert B. Keller, and Jean Pierre Kernevez (1991a). "Numerical Analysis And Control Of Bifurcation Problems (I): Bifurcation In Finite Dimensions". In: *International Journal of Bifurcation and Chaos* 01.03, pp. 493–520. DOI: 10.1142/S0218127491000397. eprint: <https://doi.org/10.1142/S0218127491000397>. URL: <https://doi.org/10.1142/S0218127491000397>.
- Doedel, Eusebius, Herbert B. Keller, and Jean Pierre Kernevez (1991b). "Numerical Analysis And Control Of Bifurcation Problems (II): Bifurcation In Infinite Dimensions". In: *International Journal of Bifurcation and Chaos* 01.04, pp. 745–772. DOI: 10.1142/S0218127491000555. eprint: <https://doi.org/10.1142/S0218127491000555>. URL: <https://doi.org/10.1142/S0218127491000555>.
- Chen, Tianping and Hong Chen (1995). "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems". In: *IEEE Transactions on Neural Networks* 6.4, pp. 911–917.
- Neal, Radford M. (1996). *Bayesian Learning for Neural Networks*. Berlin, Heidelberg: Springer-Verlag. ISBN: 0387947248.
- Mitchell, Tom M. (1997). *Machine Learning*. New York: McGraw-Hill. ISBN: 978-0-07-042807-2.
- Henderson, Michael E. (2002). "Multiple Parameter Continuation: Computing Implicitly Defined k-Manifolds". In: *International Journal of Bifurcation and Chaos* 12.03, pp. 451–476. DOI:

- 10.1142/S0218127402004498. eprint: <https://doi.org/10.1142/S0218127402004498>. URL: <https://doi.org/10.1142/S0218127402004498>.
- Dhooge, A., W. Govaerts, and Yu. A. Kuznetsov (2003). "MATCONT: A MATLAB Package for Numerical Bifurcation Analysis of ODEs". In: *ACM Trans. Math. Softw.* 29.2, 141–164. ISSN: 0098-3500. DOI: 10.1145/779359.779362. URL: <https://doi.org/10.1145/779359.779362>.
- Kelley, C. T. (2003). *Newton—Krylov Methods*, pp. 57–83. DOI: 10.1137/1.9780898718898.ch3. eprint: <https://epubs.siam.org/doi/pdf/10.1137/1.9780898718898.ch3>. URL: <https://epubs.siam.org/doi/abs/10.1137/1.9780898718898.ch3>.
- Kelley, C. T., I. G. Kevrekidis, and L. Qiao (2004). *Newton-Krylov solvers for time-steppers*. DOI: 10.48550/ARXIV.MATH/0404374. URL: <https://arxiv.org/abs/math/0404374>.
- Nadler, Boaz, Stéphane Lafon, Ioannis Kevrekidis, and Ronald Coifman (2005). "Diffusion Maps, Spectral Clustering and Eigenfunctions of Fokker-Planck Operators". In: *Advances in Neural Information Processing Systems*. Ed. by Y. Weiss, B. Schölkopf, and J. Platt. Vol. 18. MIT Press. URL: <https://proceedings.neurips.cc/paper/2005/file/2a0f97f81755e2878b264adf39cba68e-Paper.pdf>.
- Coifman, Ronald R. and Stéphane Lafon (2006). "Diffusion maps". In: *Applied and Computational Harmonic Analysis* 21.1, pp. 5–30. ISSN: 1063-5203. DOI: <https://doi.org/10.1016/j.acha.2006.04.006>. URL: <https://www.sciencedirect.com/science/article/pii/S1063520306000546>.
- Nadler, Boaz, Stéphane Lafon, Ronald R. Coifman, and Ioannis G. Kevrekidis (2006). "Diffusion maps, spectral clustering and reaction coordinates of dynamical systems". In: *Applied and Computational Harmonic Analysis* 21.1, pp. 113–127. ISSN: 1063-5203. DOI: <https://doi.org/10.1016/j.acha.2005.07.004>. URL: <https://www.sciencedirect.com/science/article/pii/S1063520306000534>.
- Rasmussen, C. E. and C. K. I. Williams (2006). *Gaussian Processes for Machine Learning*. MIT Press.
- Doedel, Eusebius J., Thomas F. Fairgrieve, Björn Sandstede, Alan R. Champneys, Yuri A. Kuznetsov, and Xianjun Wang (2007). *AUTO-07P: Continuation and bifurcation software for ordinary differential equations*. Tech. rep.
- Lee, John and Michel Verleysen (2007). *Nonlinear Dimensionality Reduction*. Vol. 8226. ISBN: 978-0-387-39350-6. DOI: 10.1007/978-0-387-39351-3.
- Settles, Burr (2009). *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin–Madison.
- Brochu, Eric, Vlad M. Cora, and Nando de Freitas (2010). "A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning". In: *ArXiv abs/1012.2599*.
- Srinivas, Niranjan, Andreas Krause, Sham Kakade, and Matthias Seeger (2010). "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design". In: Omnipress. ISBN: 9781605589077.
- Ermentrout, Bard (2012). "XPPAUT". In: *Computational Systems Neurobiology*. Ed. by N. Le Novère. Dordrecht: Springer Netherlands, pp. 519–531. ISBN: 978-94-007-3858-4. DOI: 10.1007/978-94-007-3858-4\_17. URL: [https://doi.org/10.1007/978-94-007-3858-4\\_17](https://doi.org/10.1007/978-94-007-3858-4_17).
- Dankowicz, Harry and Frank Schilder (2013). *Recipes for Continuation*. USA: Society for Industrial and Applied Mathematics. ISBN: 1611972566.
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. MIT Press.

- Shahriari, Bobak, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas (2016). "Taking the Human Out of the Loop: A Review of Bayesian Optimization". In: *Proceedings of the IEEE* 104.1, pp. 148–175. DOI: [10.1109/JPROC.2015.2494218](https://doi.org/10.1109/JPROC.2015.2494218).
- Dsilva, Carmeline J., Ronen Talmon, Ronald R. Coifman, and Ioannis G. Kevrekidis (2018). "Parsimonious representation of nonlinear dynamical systems through manifold learning: A chemotaxis case study". In: *Applied and Computational Harmonic Analysis* 44.3, pp. 759–773. ISSN: 1063-5203. DOI: <https://doi.org/10.1016/j.acha.2015.06.008>. URL: <https://www.sciencedirect.com/science/article/pii/S1063520315000949>.
- Renson, L., J. Sieber, D. A. W. Barton, A. D. Shaw, and S. A. Neild (2019). "Numerical continuation in nonlinear experiments using local Gaussian process regression". In: *Nonlinear Dynamics* 98.4, pp. 2811–2826. DOI: [10.1007/s11071-019-05118-y](https://doi.org/10.1007/s11071-019-05118-y). URL: <https://doi.org/10.1007/s11071-019-05118-y>.
- Lee, Seungjoon, Mahdi Kooshkbaghi, Konstantinos Spiliotis, Constantinos I. Siettos, and Ioannis G. Kevrekidis (2020). "Coarse-scale PDEs from fine-scale observations via machine learning". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 30.1, p. 013141.

## Chapter 3

# Discovering Limits of Entrainment for Circadian Neuronal Networks

This work was completed in collaboration with Prof. Mihalis Kavousanakis and Prof. Ioannis Kevrekidis. Work presented here is included in the manuscript (under preparation) “Limits of Entrainment of Circadian Neuronal Networks”.

### 3.1 Motivation

The principal circadian pacemaker is located in the two Suprachiasmatic Nuclei of the anterior hypothalamus. The SCN directly receives input from photosensitive cells of the retina via the retino-hypothalamic tract (Ma and Morrison, 2022; Hastings, Maywood, and Brancaccio, 2018) while it is also connected to the pineal gland, inducing melatonin production during the night (Sack et al., 2007b; Ma and Morrison, 2022). The SCN also coordinates secondary cellular pacemakers across the body by controlling biochemical (e.g. neuroendocrine) signals that entrain them (Hastings, Maywood, and Brancaccio, 2018).

This orchestrating role, puts SCN neuronal networks at the center of numerous physiological processes and, consequently, its dysfunction gives rise to numerous disorders. Circadian desynchrony is linked to sleep disorders (jet lag disorder, shift work disorder, advanced sleep phase disorder, delayed

sleep phase disorder, free-running disorder, irregular sleep-wake rhythm) (Sack et al., 2007b; Sack et al., 2007a), cardiovascular disease, obesity, hypertension (Scheer et al., 2009), nephropathy (Nakano et al., 1991), cancer (Savvidis and Koutsilieris, 2012), depression and bipolar disorder (Germain and Kupfer, 2008; Vadnie and McClung, 2017).

Many pharmacological interventions have been suggested to correct misalignments of the circadian rhythm with the external day-night signal. Benzodiazepines and melatonin have been popular choices (Turek and Losee-Olson, 1986; Sack et al., 2000). As the understanding of the the exact biochemical processes involved deepens, more targeted therapies are developed. Small molecules like Longdaysin (CAS No. : 1353867-91-0) have been extensively studied (*in vitro* and *in silico*) as therapeutics in circadian rhythm disorders, as they directly intervene to the gene regulatory network giving rise to the oscillations (Huang et al., 2020; St. John et al., 2014; Hirota et al., 2012).

In this work, we use advanced algorithms from numerical analysis/scientific computation and nonlinear dynamics to investigate the ability of SCN neurons and SCN neuronal networks to synchronize with the external day-night signal (in dynamics terms, with the external *forcing*). Specifically, we explore entrained periodic solutions of *high-dimensional* dynamical systems arising from computational biology models, with respect to different:

1. Forcing angular frequencies  $\omega_f$ , where  $\omega_f = \frac{2\pi}{T_f}$ , for a forcing period  $T_f$  of the day-night (light-dark) cycle.
2. Forcing duty cycles  $\phi$ , where  $\phi = \frac{T_{light}}{T_f}$  for a day duration of  $T_{light}$ .
3. Simulated Longdaysin effects.
4. Network Heterogeneity extents.

For this purpose, several informative bifurcation diagrams are constructed and the extracted bifurcation points demarcate the limits of entrainment of circadian neurons or neuronal networks. Moreover, we construct a reduced, data-driven “emergent space” description of neuronal behavior heterogeneity using unsupervised learning.

## 3.2 Computational Model

The computational model used is adapted from the work of Vasalou *et al.* (Vasalou, Herzog, and Henson, 2011; Vasalou and Henson, 2011). They constructed a state-of-the-art computational model of circadian neuronal networks by coupling three components:



1. **Biomolecular clock:** The computational model of the biomolecular clock was developed by Leloup and Goldbeter (Leloup and Goldbeter, 2003; Leloup and Goldbeter, 2004) and describes the regulatory loops involving the *Per*, *Cry*, *Bmal1*, *Clock* and *Erv-ErbA* genes. The interaction of these loops gives rise to circadian oscillations.
2. **Electrophysiology dynamics:** This component describes the membrane dynamics of circadian neurons and the way they are coupled with neurotransmitter signaling i.e. with  $\gamma$ -aminobutyric acid (GABA) and vasoactive intestinal polypeptide (VIP) signaling (To et al., 2007; Vasalou and Henson, 2011). The firing frequency is incorporated in the model by associating membrane voltage and ion conductances with circadian gene expression (Vasalou and Henson, 2010).
3. **Network connectivity:** An ensemble of 425 neurons was chosen for a realistic representation of the SCN circadian network (Vasalou and Henson, 2011). All neurons are linked via VIP and GABA neurotransmitter networks generated from a small-world architecture resembling neuron connectivities in the SCN (Vasalou and Henson, 2011) (see Fig.3.1 for example).

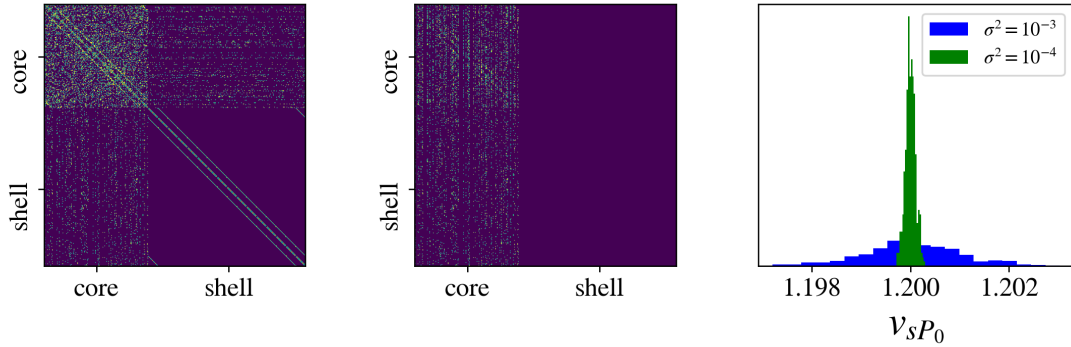
Henson (Henson, 2013) provides a review comparing this model with others in the literature. There is also experimental evidence supporting the model's validity (Vasalou and Henson, 2011; Leloup and Goldbeter, 2003).

In this model, the day-night (light-dark) cycle is modeled as a step function with period  $T_f = T_{light} + T_{dark}$  and duty cycle  $\phi = \frac{T_{light}}{T_f}$ . Light forcing is incorporated in the dynamical system implicitly, by assuming that during the light phase, the AMPA/NDMA and VPAC2 receptors are saturated (in term of model functions:  $b_{Glu_R}(t) = 1, b_{VIP}(t) = 1$  for  $t \in [0, T_{light}]$ ) (Vasalou and Henson, 2011). In contrast to Vasalou *et al.* we assume that the photic effect is uniform across all SCN (both core and shell) neurons.

The effect of Longdaysin can also be incorporated in the model. Longdaysin is a small drug molecule acting as a casein kinase I (CKI) inhibitor. It is hypothesized to increase the time required before the *Per-Cry* complex can enter the nucleus to repress transcription (Hirota et al., 2012; St. John et al., 2014). In our model, this effect can be simulated by different values of the parameter  $k_1$ , the nuclear entry rate of phosphorylated *Per-Cry* complex (St. John et al., 2014). Therefore, in our studies, increased values of Longdaysin translate to lower values of  $k_1$ . Note that, here, we assume that the concentration of Longdaysin is constant and not subject to pharmacodynamics.

The realization of the circadian network studied here includes heterogeneity in the parameter  $v_{sP_0}$ , the basal transcription rate of the *Per* mRNA, across the neurons. This parameter has been shown

to strongly affect the ability of circadian neurons to sustain intrinsic oscillations, while the rhythmic phenotype of *Per* has been shown experimentally to vary (Yamaguchi et al., 2003). Specifically,  $v_{sP_0}$  has been sampled from  $\mathcal{N}(1.2, \sigma^2)$ , similarly to (Vasalou, Herzog, and Henson, 2011) (see Fig.3.1 for example). It is important to mention, that for reproducibility, the random seed used to generate  $v_{sP_0}$  and the small-world networks is always fixed.



**Figure 3.1:** (left) GABA adjacency matrix, (middle) VIP adjacency matrix, (right) histograms sampled from different  $v_{sP_0}$  considered here.

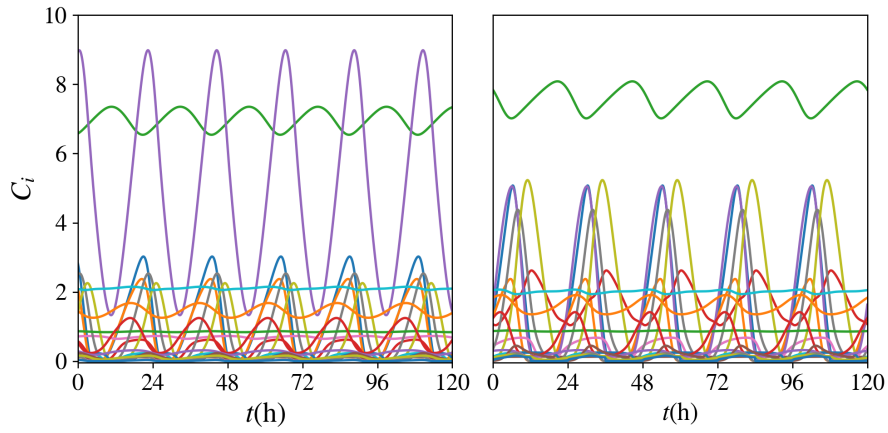
Putting all components together, each circadian neuron is described by 21 Ordinary Differential Equations (ODEs), each of which describes the time evolution of a relevant chemical species. When the entire network is simulated, the resulting system is 8925-dimensional.

### 3.3 Single neuron bifurcation studies

When all neurons in a connected network are homogeneous, the network can in principle behave as each one of its individual neurons. We, therefore, begin our analysis by studying the behavior of a single circadian neuron.

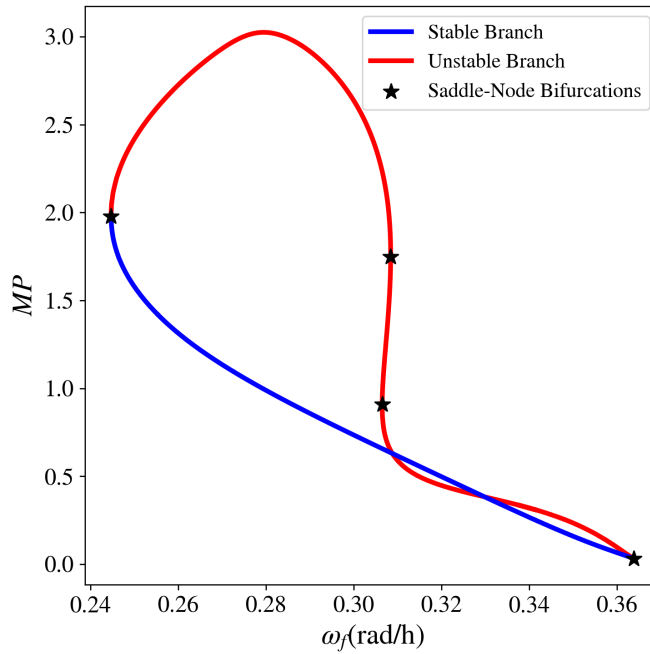
As can be seen from Fig.??, in the absence of a photic stimulus, the circadian neuron will oscillate with its intrinsic frequency, while, in the presence of a photic stimulus, the neuron can get entrained. Notice that at the frequency locked “periodic steady state”, all the chemical species will oscillate with the same frequency but will not necessarily be at the same phase (*e.g.* not all species reach their maximum concentration simultaneously).

To explore entrained periodic solutions of a single circadian neuron, we perform pseudo-arclength



**Figure 3.2:** Timeseries of steady state oscillations of the (left) forced system with natural period  $T_0 = 22.04\text{h}$ , and (right) forced system with  $T_f = 24\text{h}$ .

continuation w.r.t the forcing angular frequency  $\omega_f$  (reminder:  $\omega_f = \frac{2\pi}{T_f}$ ) resulting in bifurcation diagrams like the one in Fig.3.3.



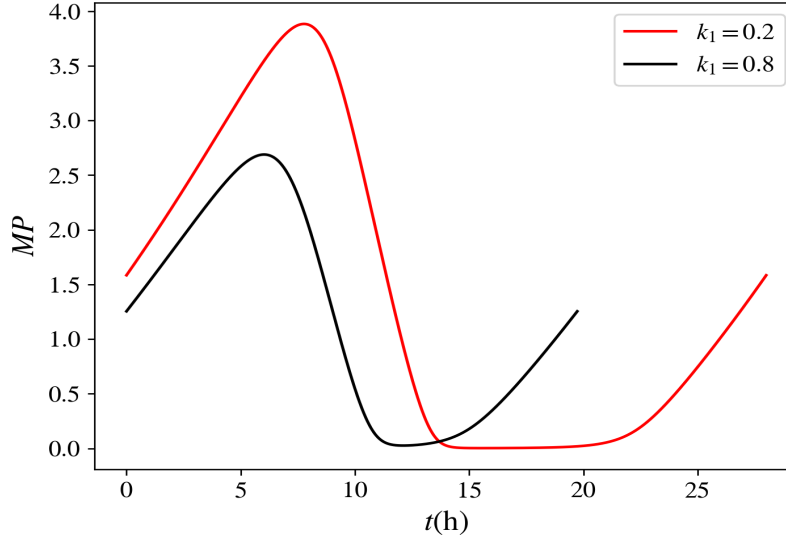
**Figure 3.3:** Bifurcation diagram for periodic steady states under photic stimulus for a single circadian neuron. Here, the  $MP$  (*Per* mRNA) projection is shown. For reference  $\omega_f = 0.262\text{rad/h}$  for  $T_f = 24\text{h}$ .

Here, (and for the rest of continuation studies in Chapter 3) pseudo-arclength continuation is used

(described in Subchapter 2.1.2) with a Newton-Krylov GMRES solver (described in Subchapter 2.1.1.2). The use of the variational approach is avoided due to its computational cost.

As a reminder, in bifurcation studies of entrainment problems, such loops are common when continuing solutions w.r.t. the forcing angular frequency (Tomita and Kai, 1979) and are termed *isolas* (see Subchapter 1.1.1). Each point of the isola corresponds to a fixed point of the stroboscopic map, or, equivalently, to a periodic steady state of the original 21-dimensional ODE system. The limiting values of the bifurcation diagram w.r.t. the forcing angular frequency define *the limits of entrainment* and constitute saddle-node bifurcations of limit cycles. That is why the periodic solutions exchange stability at these points (see Fig. 3.3). Along the unstable branch (red) two additional saddle-node bifurcations are observed, further destabilizing the (already unstable) entrained solution.

To simulate the effect of the drug Longdaysin, we investigate how the synchronization limits change for different values of  $k_1$ . As suggested in (St. John et al., 2014) we expect decreasing values of  $k_1$ , (which correspond to higher doses of Longdaysin) to lead to longer intrinsic (unforced) periods and larger oscillation amplitudes for the non-driven neurons. First, we perform continuation of the *unforced* system w.r.t.  $k_1$  and examine the limit cycles at the two extreme limits of the  $k_1$  interval (Fig.3.4). Then, we select a discrete, representative set of  $k_1$  values, and calculate the limits of entrainment for each one of these values by constructing isolas (as in Fig. 3.3) with respect to the forcing frequency. Note that the nominal value of  $k_1$  according to (Vasalou, Herzog, and Henson, 2011) is  $k_1 = 0.49$  (also included). Note, that in the original model  $k_1$  has units of frequency ( $\text{h}^{-1}$  which will be implied in the following text, for simplicity).

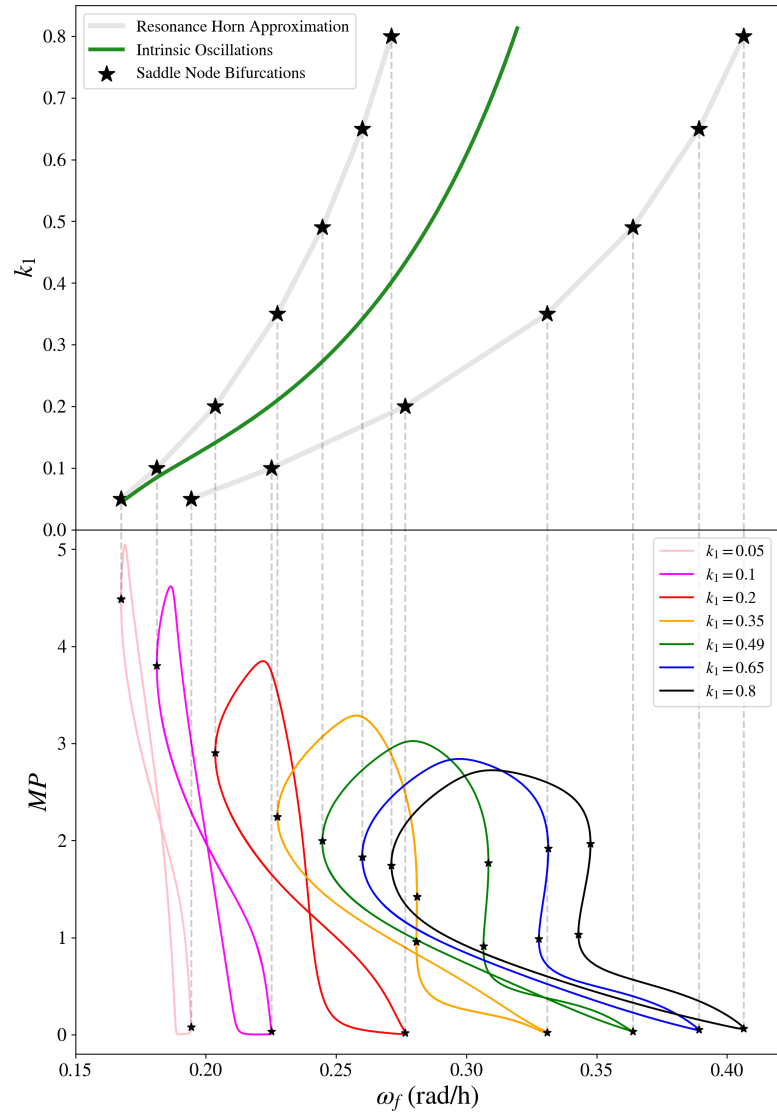


**Figure 3.4:** Unforced  $MP$  oscillations at the periodic steady state of  $k_1 = 0.2$  and  $k_1 = 0.8$ .

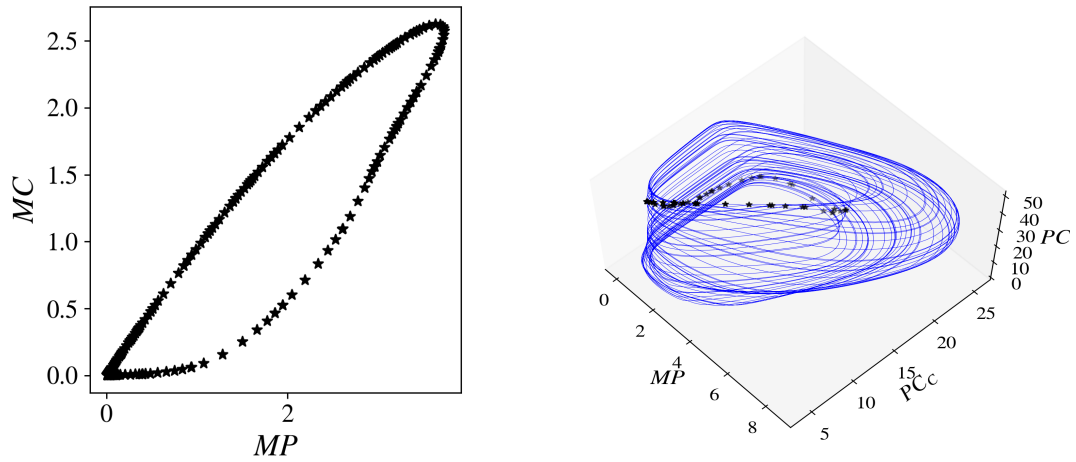
We thus confirm that lower values of  $k_1$  correspond to higher circadian periods and amplitudes of  $MP$  oscillations. It can also be seen from Fig. 3.5 that lower values of  $k_1$  result in tighter entrainment regions. The “natural” period of unforced oscillations always lies within the respective entrainment boundaries, as discussed in (Tomita and Kai, 1979). The bifurcation diagram in the  $k_1 - \omega_f$  space is equivalent to what is known in the literature as a *resonance horn* or *Arnold tongue*, and demarcates the parameter subregion where entrainment by an external forcing signal is feasible.

Outside of the resonance horn in the the  $k_1 - \omega_f$  space, various other dynamic responses are expected, *e.g.* quasiperiodicity, frequency locking, or chaos (Tomita and Kai, 1979). Fig. 3.6 shows quasiperiodic response for low values of  $k_1$ , at  $\omega_f = 0.16746\text{rad/h}$  right after crossing the left saddle-node bifurcation (at  $\omega_f = 0.16749\text{rad/h}$ ). Indeed, as it can be seen in the right panel of Fig. 3.6 the trajectories are now attracted to a high-dimensional torus.

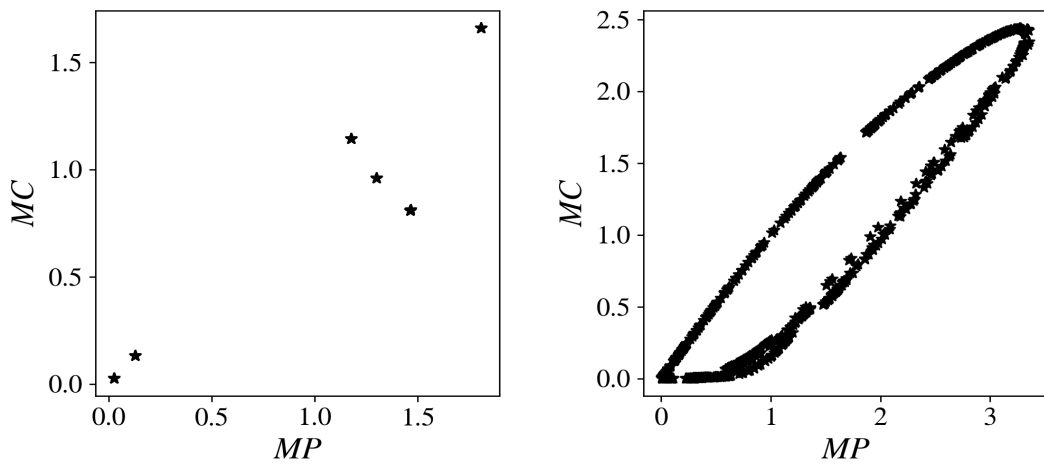
For higher  $k_1$  values it is expected that different phenomena arise (Kevrekidis, Schmidt, and Aris, 1986; Tomita and Kai, 1979). Indeed, for  $k_1 = 0.49$  and  $\omega_f = 0.2446\text{rad/h}$  a period-6 solution arises (frequency locking) after crossing the left bifurcation at  $\omega_f = 0.2447\text{rad/h}$  (left panel, Fig. 3.7).



**Figure 3.5:** Bifurcation diagram for periodic steady states under photic stimulus for varying angular frequency  $\omega_f$  and Longdaysin effect ( $k_1$  value) in the case of a single circadian neuron. At the bottom subfigure, a collection of isolas is shown for discrete values of  $k_1$ . At the top figure, the *resonance horn* is approximated in the  $\omega_f - k_1$  space.



**Figure 3.6:** Loss of entrainment for  $k_1 = 0.05, \omega_f = 0.16746 \text{ rad/h}$ . Near the *tip* of the resonance horn (low  $k_1$ ), quasiperiodicity is observed outside, but close to the entrainment limits. This can be confirmed by plotting iterates of the stroboscopic map (here, shown in the  $MC - MP$  projection), where an invariant circle is observed (left). In the phase portrait representation (right), trajectories are attracted to a torus (here, the  $MP - PC_c - PC$  projection is shown, along with stroboscopic map iterates). Note that  $MP$ : *Per* mRNA,  $MB$ : *Bmal1* mRNA,  $MC$ : *Cry* mRNA,  $PC$ : nonphosphorylated *Cry* protein in the cytosol,  $PC_c$ : nonphosphorylated *Per-Cry* protein complex in the cytosol.

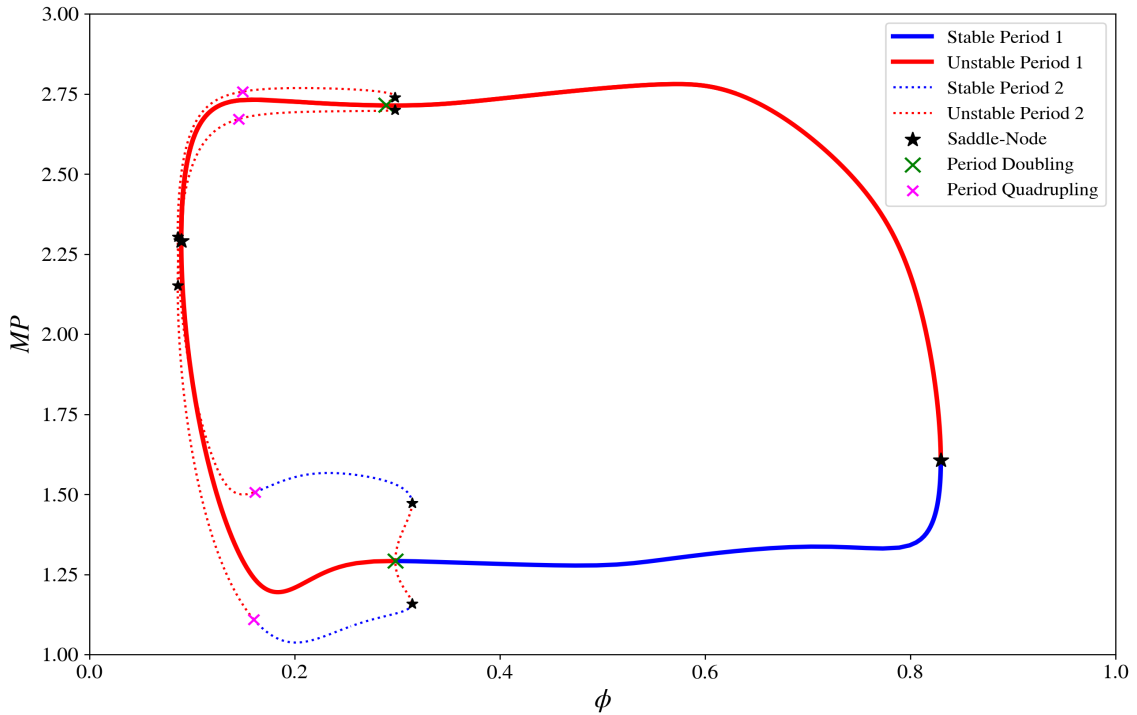


**Figure 3.7:** Loss of entrainment for  $k_1 = 0.49 : \omega_f = 0.2446 \text{ rad/h}$ . (left panel)  $\omega_f = 0.3639 \text{ rad/h}$  (right panel). Stroboscopic map iterates reveal frequency-locking and chaos respectively (here in the  $MC - MP$  projection)

Crossing the other limit of entrainment for  $k_1 = 0.49$  (saddle-node bifurcation at  $\omega_f = 0.3638\text{rad/h}$ ), chaos is observed to emerge for  $\omega_f = 0.3639\text{rad/h}$  (right panel, Fig. 3.7).

Figs. 3.6, 3.7 stand as evidence that periodically forced circadian neurons demonstrate the entire gamut of dynamic responses typical of periodically forced dynamical systems (Tomita and Kai, 1979).

Subsequently, we perform continuation of the periodic steady states w.r.t the duty cycle ( $\phi$ ) for a fixed period (here,  $T_f = 24\text{h}$  or, equivalently,  $\omega_f = 0.268\text{rad/h}$ ). In other words, we investigate entrainment of circadian neurons when the ratio of the length of the day (and, correspondingly, the duration of the night) changes.

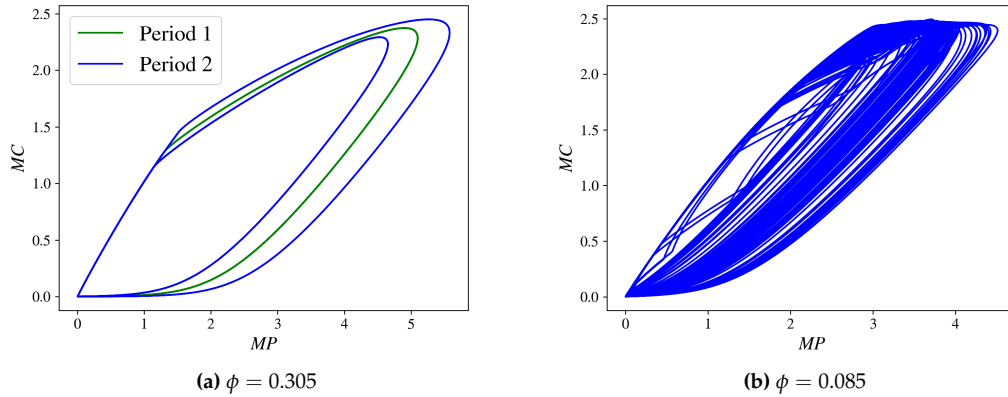


**Figure 3.8:** Bifurcation diagram for periodic steady states w.r.t the duty cycle  $\phi$  under photic stimulus of fixed angular frequency  $\omega_f = 0.268\text{rad/h}$  and without drug intervention ( $k_1 = 0.49$ ) in the case of a single circadian neuron. Notice the additional bifurcations (leading to complex dynamics) towards lower day/night ratios.

As seen in Fig. 3.8 there are entrainment limits w.r.t to the duty cycle as well. As anticipated, circadian neurons are unable to synchronize when the day/night phases become highly unbalanced. However, entrainment is not always lost in exactly the same way as the typical isolas we have encountered thus far.



For high day/night phase ratios, a saddle-node bifurcation marks the upper boundary of entrainment; but at low day/night ratios, the system undergoes a period doubling bifurcation (both the stable and the unstable branch). This means that the circadian neurons will return to the same state after two forcing periods. Interestingly, for some region, stable period-1 and period-2 solutions coexist, which means that circadian neurons are briefly bi-stable (Fig. 3.9a). It is worth noting that the period-2 branches undergo an additional period-doubling bifurcation, leading to period-4 solutions. We hypothesize that this is the beginning of a period-doubling route to chaos (Fig. 3.9b) through a cascade of period-doubling bifurcations.



**Figure 3.9:** Phase portraits at low  $\phi$  values where bistability as well as chaos is observed.

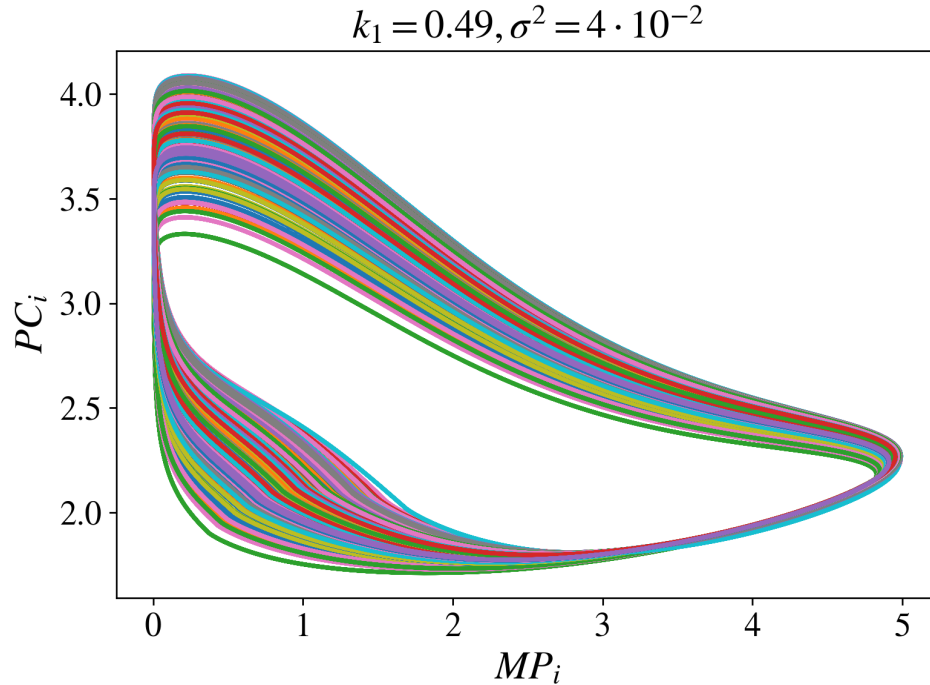
### 3.4 Heterogeneous network bifurcation studies

In a realistic neuronal network each neuron is expected to be unique, or, in terms of computational modeling, to have its own intrinsic parameter values. As described in Subchapter 3.2, here we consider networks of 425 neurons that are heterogeneous w.r.t. the values of the parameter  $v_{s,p_0}$ , for each neuron.

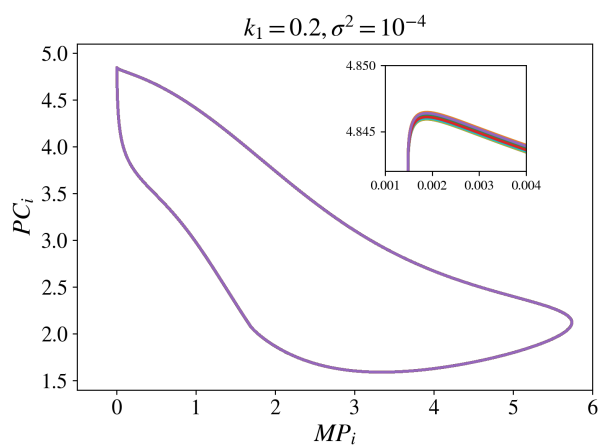
The effect of heterogeneity can be seen when plotting projections of the (now, 8925–dimensional) limit cycle for two variables of each neuron (Fig. 3.10). We plot these limit cycles for two extents of heterogeneity (variance of the -here, normal- distribution of the heterogeneous parameter) and for five different simulated Longdaysin effects ( $k_1$ ). As seen in Figs. 3.10 and 3.11 increasing heterogeneity causes the trajectories of different neurons to move further apart. However, all neurons follow qualitatively similar trajectories since the neuronal network is entrained.

By analogy with Fig.3.5, we can computationally construct a resonance horn for each heterogeneity extent (Fig. 3.12). The entrainment limits do not change much with increasing neuron heterogeneity. This would seem to suggest robustness of the neuronal network to heterogeneity variations. (Komin et al., 2011).

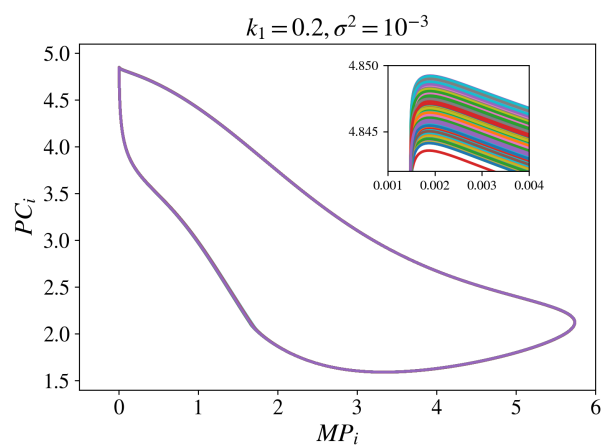
As in the case of single neuron studies, we can explore how entrainment is lost in the neuronal network. As can be seen in Fig. 3.13, after crossing the right saddle-node bifurcation for ( $k_1 = 0.49$  and  $\sigma^2 = 10^{-2}$ ,  $\omega_f = 0.2455\text{rad/h}$ ) a single “rogue” oscillator emerges. By that, we mean that even though most neurons appear to oscillate in synchrony, one of them gradually desynchronizes and starts oscillating “on its own” with varying amplitude. The emergence of such a “rogue” oscillator can be attributed to large variance of the heterogeneity parameter and is associated with bifurcations of the autonomous dynamical system (Bold et al., 2007). In the dynamical systems literature a simple caricature of such a bifurcation is provided by the SNIPER (saddle-node infinite period) bifurcation (Bertalan et al., 2017; Moon and Kevrekidis, 2006).



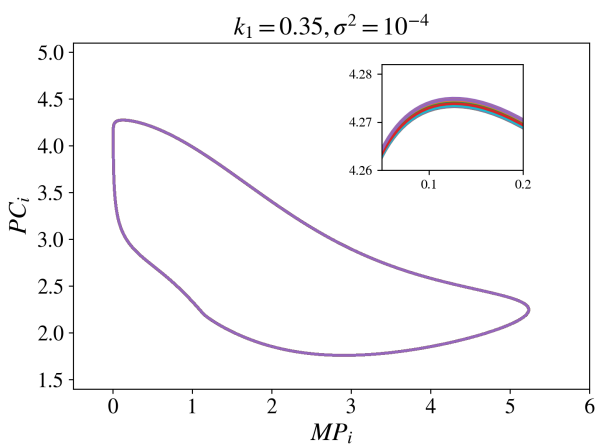
**Figure 3.11:** 2D projections of the 8925–dimensional limit cycle for an even higher value of heterogeneity variance studied here ( $\sigma^2 = 0.04$ ), for comparison purposes. This limit cycle was calculated for  $k_1 = 0.49$ ,  $\phi = 0.5$  with forcing period  $T_f = 24\text{h}$ . This figure should be qualitatively compared to Figs. 3.10e, 3.10f.



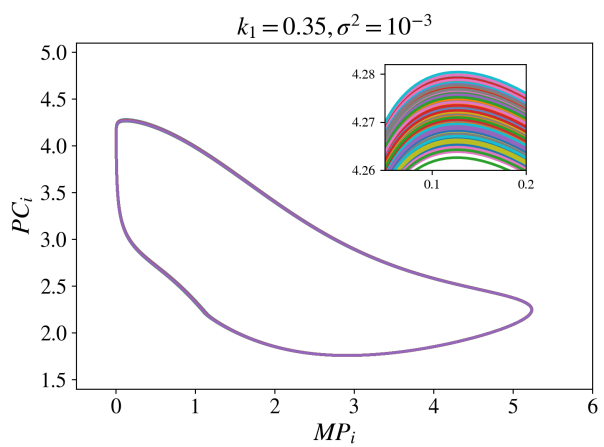
(a)



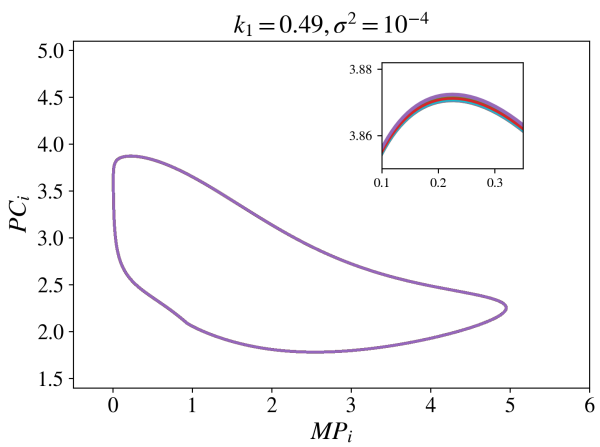
(b)



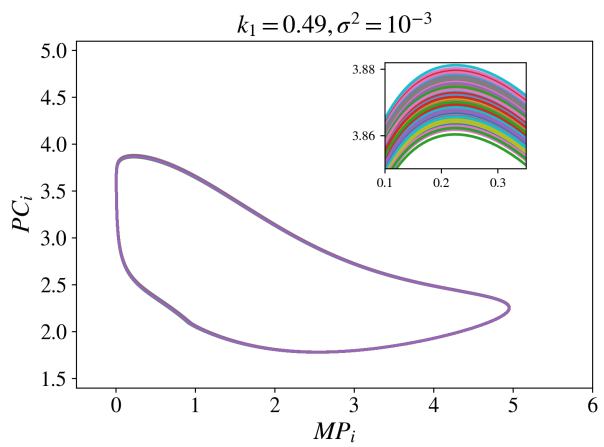
(c)



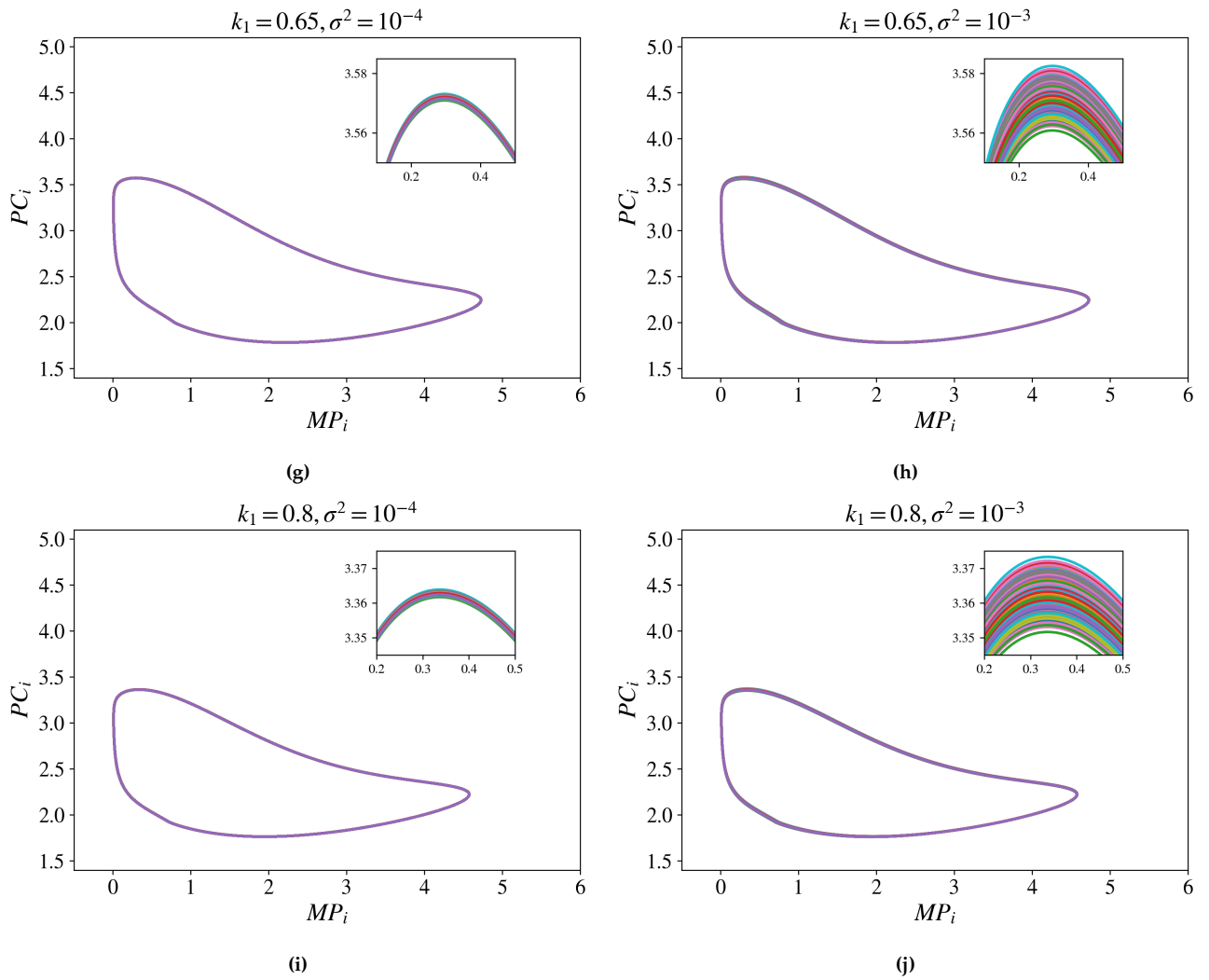
(d)



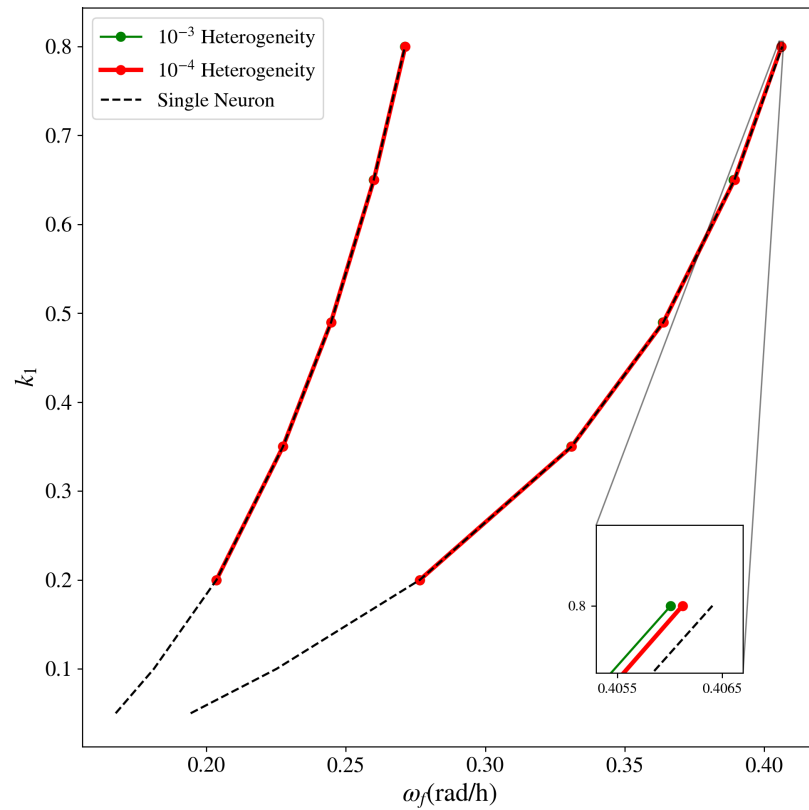
(e)



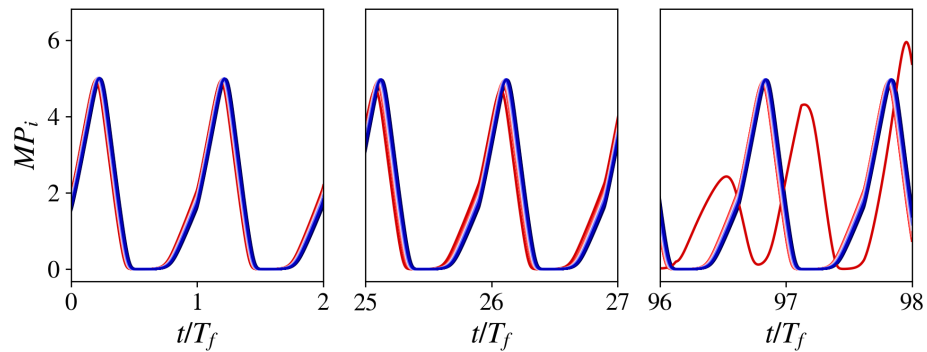
(f)



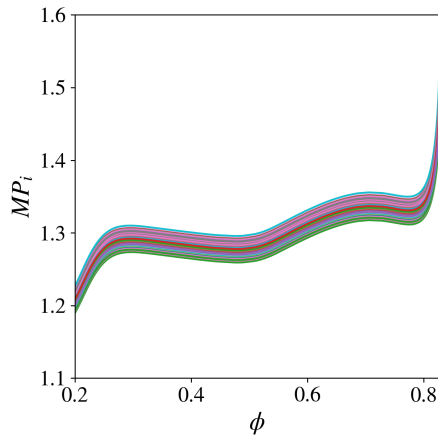
**Figure 3.10:** 2D projections of the 8925–dimensional limit cycles, for  $k_1$  in  $\{0.2, 0.35, 0.49$  (nominal value),  $0.65, 0.8\}$ , for heterogeneity variance  $10^{-3}$  or  $10^{-4}$ , with forcing angular frequency equal to the intrinsic one (for each  $k_1$ ) and for  $\phi = 0.5$ .



**Figure 3.12:** Superimposed resonance horns for the case of a single circadian neuron, and a neuronal networks with varying extents of heterogeneity.



**Figure 3.13:** Loss of entrainment due to the emergence of a single “rogue” oscillator. Three panels show oscillation “snapshots” along a long trajectory. Initially (left panel), all neurons seem to be oscillating in synchrony. After some time (middle panel), the trajectory of one neuron slowly diverges from the rest, and for longer times (right panel) it starts oscillating erratically. Variable  $MP_i$  is reported for every neuron,  $i$  while each neuron is colored by its heterogeneity  $v_{sp_0}$  value; the rogue oscillator has the highest. Forcing parameters and heterogeneity variance value are mentioned in the text.



**Figure 3.14:** Continuation branch of periodic steady states solutions for the neuronal network w.r.t. the duty cycle  $\phi$  for heterogeneity variance of value  $\sigma^2 = 10^{-3}$  and  $k_1 = 0.49$ ,  $T_f = 24h$ . Here the variable  $MP_i$  is reported for every neuron  $i$  at the initial phase of the forcing. This should be compared to the lower branch in Fig. 3.8.

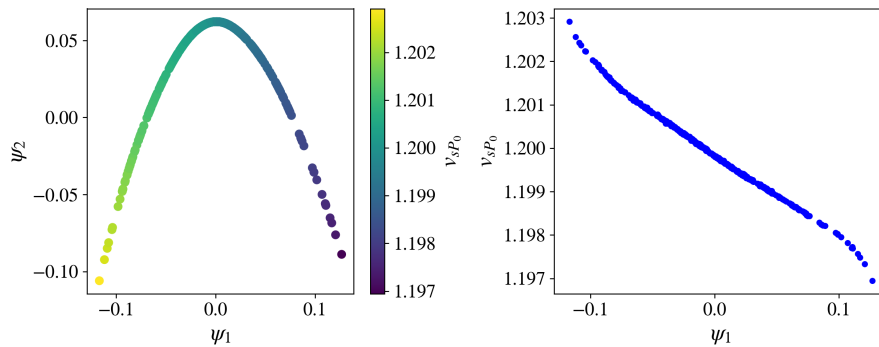
Similarly to Fig.3.8 we can also continue periodic solutions of the high-dimensional, heterogeneous neuronal network w.r.t.  $\phi$ , the day/night ratio (Fig.3.14). As expected, the bifurcation diagram for a single circadian neuron is again qualitatively similar to the large, mildly heterogeneous network diagram.

### 3.5 Latent heterogeneity space

Realistic neuronal networks are characterized by heterogeneity with respect to multiple physical properties. Enumeration and identification of such heterogeneities (and respective parameters for a computational model) from observed neuronal oscillation data is highly nontrivial, and is clearly affected by: (a) the quality and quantity of data; (b) stochasticity/noise inherent to measurements of biological systems; and (c) the complexity of the computational model and assumptions used in its formulation.

With increasing model complexity, it is clear that increasing parameter heterogeneity and variability (both for intrinsic kinetic parameters, and for structural network connectivity parameters) becomes important. When these parameters are not explicitly known, it is the variability of the neuronal behavior itself that encodes it; and so one can obtain a sense of the parameter heterogeneity from the neuronal time-series variability itself. In other words, it should be possible to infer the effective parametric heterogeneity/variability of the entire network by the richness of the variability of the

observed oscillations of individual neurons. Here, we employ a data-driven approach to uncover an *effective* heterogeneity space purely from observed oscillation data. The algorithm we employ is Diffusion Maps (a nonlinear, manifold learning technique), to embed oscillation data in an *effective*, data-driven heterogeneity space Coifman and Lafon, 2006; Thiem et al., 2021; Thiem et al., 2020; Kemeth et al., 2018. The idea of creating such data-driven *emergent spaces* purely from observations was proposed in Kemeth et al., 2018; Kemeth et al., 2022 and it was used in the simple case of Kuramoto-type oscillators in Thiem et al., 2021.



**Figure 3.15:** (left) Data from a single phase of the limit cycle plotted on the space defined by the first two nontrivial diffusion maps eigenvectors. All data points are colored by the *a priori known* heterogeneity parameter  $v_{sP_0}$ . Data are chosen from the case of  $\phi = 0.5, k_1 = 0.49, \sigma^2 = 10^{-3}$  for forcing angular velocity equal to the intrinsic one. (right) The first nontrivial diffusion maps eigenvector plotted against the heterogeneity parameter.

As in Kemeth et al., 2018 we select data from a single phase of the limit cycle (the exact phase does not matter); the data set therefore consists of 425 data points (number of neurons) in a 21-dimensional space (number of variables per neuron). As shown in Fig.3.15 diffusion maps reveals that the (synchronized) neuronal states are effectively one-dimensional, as  $\psi_1$  seems to be the only independent eigenvector. The right panel of Fig.3.15 confirms that the observed data-based emergent heterogeneity descriptor  $\psi_1$  is indeed one-to-one with the (true) intrinsic heterogeneity  $v_{sP_0}$ : the data-driven heterogeneity parametrization is one-to-one with the true physical one.

### 3.6 Discussion and Conclusions

This work was aimed at computationally exploring the limits of entrainment of circadian neurons and circadian neuronal networks. We employ modern techniques from scientific computing, such as matrix-free timestepper-based algorithms, to circumvent limitations inherent in high-dimensional

dynamical systems. Specifically, it shown that using Newton-Krylov GMRES to find fixed points of a 8925–dimensional stroboscopic map is computationally efficient (considering the alternative, i.e. calculating  $8925^2$  entries of a Jacobian). These algorithms enable the exploration of the simulated effect of (i) forcing frequency, (ii) forcing duty cycle, (iii) Longdaysin dosing level, and (iv) neuronal heterogeneity, on the ability of circadian neurons to entrain to the day/night cycle. Lastly, an unsupervised learning algorithm is used to discover an *effective* neuronal heterogeneity space from observed oscillation data.

A wealth of responses to different day/night cycle conditions is demonstrated, such as entrainment, quasiperiodicity, phase-locking and chaos. Linking fundamental concepts of nonlinear dynamics to computational neuroscience can lead to a holistic understanding of circadian dynamics and motivate real world applications. Especially in the case of simulated pharmacological effects, such a model can be thought as a computational “sandbox” for therapeutic (possibly, personalized) interventions to the SCN. Furthermore, combining scientific computing with Machine Learning can provide significant insights into the underlying degrees of freedom of systems as complex as the body’s own clock.



# References

- Tomita, Kazuhisa and Tohru Kai (1979). "Chaotic response of a limit cycle". In: *Journal of Statistical Physics* 21.1, pp. 65–86. ISSN: 1572-9613. DOI: [10.1007/BF01011482](https://doi.org/10.1007/BF01011482). URL: <https://doi.org/10.1007/BF01011482>.
- Kevrekidis, I.G., L.D. Schmidt, and R. Aris (1986). "Some common features of periodically forced reacting systems". In: *Chemical Engineering Science* 41.5, pp. 1263–1276. ISSN: 0009-2509. DOI: [https://doi.org/10.1016/0009-2509\(86\)87099-3](https://doi.org/10.1016/0009-2509(86)87099-3). URL: <https://www.sciencedirect.com/science/article/pii/0009250986870993>.
- Turek, Fred W. and Susan Losee-Olson (1986). "A benzodiazepine used in the treatment of insomnia phase-shifts the mammalian circadian clock". In: *Nature* 321.6066, pp. 167–168. ISSN: 1476-4687. DOI: [10.1038/321167a0](https://doi.org/10.1038/321167a0). URL: <https://doi.org/10.1038/321167a0>.
- Nakano, Shigeru, Kenzo Uchida, Toshikazu Kigoshi, Sadahide Azukizawa, Ryoji Iwasaki, Mihoko Kaneko, and Shinpei Morimoto (1991). "Circadian Rhythm of Blood Pressure in Normotensive NIDDM Subjects: Its Relationship to Microvascular Complications". In: *Diabetes Care* 14.8, pp. 707–711. ISSN: 0149-5992. DOI: [10.2337/diacare.14.8.707](https://doi.org/10.2337/diacare.14.8.707). eprint: <https://diabetesjournals.org/care/article-pdf/14/8/707/341154/14-8-707.pdf>. URL: <https://doi.org/10.2337/diacare.14.8.707>.
- Sack, Robert L., Richard W. Brandes, Adam R. Kendall, and Alfred J. Lewy (2000). "Entrainment of Free-Running Circadian Rhythms by Melatonin in Blind People". In: *New England Journal of Medicine* 343.15, pp. 1070–1077. DOI: [10.1056/NEJM200010123431503](https://doi.org/10.1056/NEJM200010123431503). eprint: <https://doi.org/10.1056/NEJM200010123431503>. URL: <https://doi.org/10.1056/NEJM200010123431503>.
- Leloup, Jean-Christophe and Albert Goldbeter (2003). "Toward a detailed computational model for the mammalian circadian clock". In: *Proceedings of the National Academy of Sciences* 100.12, pp. 7051–7056. DOI: [10.1073/pnas.1132112100](https://doi.org/10.1073/pnas.1132112100). eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.1132112100>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.1132112100>.
- Yamaguchi, Shun, Hiromi Isejima, Takuya Matsuo, Ryusuke Okura, Kazuhiro Yagita, Masaki Kobayashi, and Hitoshi Okamura (2003). "Synchronization of Cellular Clocks in the Suprachiasmatic Nucleus". In: *Science* 302.5649, pp. 1408–1412. DOI: [10.1126/science.1089287](https://doi.org/10.1126/science.1089287). eprint: <https://www.science.org/doi/pdf/10.1126/science.1089287>. URL: <https://www.science.org/doi/abs/10.1126/science.1089287>.
- Leloup, Jean-Christophe and Albert Goldbeter (2004). "Modeling the mammalian circadian clock: Sensitivity analysis and multiplicity of oscillatory mechanisms". In: *Journal of Theoretical Biology* 230.4, pp. 541–562. ISSN: 0022-5193. DOI: <https://doi.org/10.1016/j.jtbi.2004.04.040>. URL: <https://www.sciencedirect.com/science/article/pii/S0022519304002437>.
- Coifman, Ronald R. and Stéphane Lafon (2006). "Diffusion maps". In: *Applied and Computational Harmonic Analysis* 21.1, pp. 5–30. ISSN: 1063-5203. DOI: <https://doi.org/10.1016/>

- j.acha.2006.04.006. URL: <https://www.sciencedirect.com/science/article/pii/S1063520306000546>.
- Moon, Sung Joon and Ioannis G. Kevrekidis (2006). "An Equation-Free Approach to Coupled Oscillator Dynamics: The Kuramoto Model Example". In: *International Journal of Bifurcation and Chaos* 16.07, pp. 2043–2052. DOI: 10.1142/S021812740601588X. eprint: <https://doi.org/10.1142/S021812740601588X>. URL: <https://doi.org/10.1142/S021812740601588X>.
- Bold, Katherine A., Yu Zou, Ioannis G. Kevrekidis, and Michael A. Henson (2007). "An equation-free approach to analyzing heterogeneous cell population dynamics". In: *Journal of Mathematical Biology* 55.3, pp. 331–352. ISSN: 1432-1416. DOI: 10.1007/s00285-007-0086-6. URL: <https://doi.org/10.1007/s00285-007-0086-6>.
- Sack, Robert L., Dennis Auckley, R. Robert Auger, Mary A. Carskadon, Jr Wright Kenneth P., Michael V. Vitiello, and Irina V. Zhdanova (2007a). "Circadian Rhythm Sleep Disorders: Part II, Advanced Sleep Phase Disorder, Delayed Sleep Phase Disorder, Free-Running Disorder, and Irregular Sleep-Wake Rhythm". In: *Sleep* 30.11, pp. 1484–1501. ISSN: 0161-8105. DOI: 10.1093/sleep/30.11.1484. eprint: <https://academic.oup.com/sleep/article-pdf/30/11/1484/13663700/sleep-30-11-1484.pdf>. URL: <https://doi.org/10.1093/sleep/30.11.1484>.
- Sack, Robert L., Dennis Auckley, Robert R. Auger, Mary A. Carskadon, Kenneth P. Wright Jr., Michael V. Vitiello, Irina V. Zhdanova, and of Sleep Medicine American Academy (2007b). "Circadian rhythm sleep disorders: part I, basic principles, shift work and jet lag disorders. An American Academy of Sleep Medicine review". eng. In: *Sleep* 30.11, pp. 1460–1483. ISSN: 0161-8105. DOI: 10.1093/sleep/30.11.1460. URL: <https://pubmed.ncbi.nlm.nih.gov/18041480>.
- To, Tsz-Leung, Michael A. Henson, Erik D. Herzog, and 3rd Doyle Francis J (2007). "A molecular model for intercellular synchronization in the mammalian circadian clock". eng. In: *Biophysical journal* 92.11, pp. 3792–3803. ISSN: 0006-3495. DOI: 10.1529/biophysj.106.094086. URL: <https://pubmed.ncbi.nlm.nih.gov/17369417>.
- Germain, Anne and David J. Kupfer (2008). "Circadian rhythm disturbances in depression". eng. In: *Human psychopharmacology* 23.7, pp. 571–585. ISSN: 1099-1077. DOI: 10.1002/hup.964. URL: <https://pubmed.ncbi.nlm.nih.gov/18680211>.
- Scheer, Frank A. J. L., Michael F. Hilton, Christos S. Mantzoros, and Steven A. Shea (2009). "Adverse metabolic and cardiovascular consequences of circadian misalignment". eng. In: *Proceedings of the National Academy of Sciences of the United States of America* 106.11, pp. 4453–4458. ISSN: 1091-6490. DOI: 10.1073/pnas.0808180106. URL: <https://pubmed.ncbi.nlm.nih.gov/19255424>.
- Vasalou, Christina and Michael A. Henson (2010). "A Multiscale Model to Investigate Circadian Rhythmicity of Pacemaker Neurons in the Suprachiasmatic Nucleus". In: *PLOS Computational Biology* 6.3, pp. 1–15. DOI: 10.1371/journal.pcbi.1000706. URL: <https://doi.org/10.1371/journal.pcbi.1000706>.
- Komin, N, A C Murza, E Hernández-García, and R Toral (2011). "Synchronization and entrainment of coupled circadian oscillators". en. In: *Interface Focus* 1.1, pp. 167–176.
- Vasalou, Christina and Michael A. Henson (2011). "A multicellular model for differential regulation of circadian signals in the core and shell regions of the suprachiasmatic nucleus". In: *Journal of Theoretical Biology* 288, pp. 44–56. ISSN: 0022-5193. DOI: <https://doi.org/10.1016/j.jtbi.2011.08.010>. URL: <https://www.sciencedirect.com/science/article/pii/S002251931100405X>.
- Vasalou, Christina, Erik D. Herzog, and Michael A. Henson (2011). "Multicellular Model for Intercellular Synchronization in Circadian Neural Networks". In: *Biophysical Journal* 101.1,

- pp. 12–20. ISSN: 0006-3495. DOI: <https://doi.org/10.1016/j.bpj.2011.04.051>. URL: <https://www.sciencedirect.com/science/article/pii/S0006349511005613>.
- Hirota, Tsuyoshi, Jae Wook Lee, Peter C. St. John, Mariko Sawa, Keiko Iwaisako, Takako Noguchi, Pagkapol Y. Pongsawakul, Tim Sonntag, David K. Welsh, David A. Brenner, Francis J. Doyle, Peter G. Schultz, and Steve A. Kay (2012). “Identification of Small Molecule Activators of Cryptochrome”. In: *Science* 337.6098, pp. 1094–1097. DOI: [10.1126/science.1223710](https://doi.org/10.1126/science.1223710). eprint: <https://www.science.org/doi/pdf/10.1126/science.1223710>. URL: <https://www.science.org/doi/abs/10.1126/science.1223710>.
- Savvidis, Christos and Michael Koutsilieris (2012). “Circadian Rhythm Disruption in Cancer Biology”. In: *Molecular Medicine* 18.9, pp. 1249–1260. ISSN: 1528-3658. DOI: [10.2119/molmed.2012.00077](https://doi.org/10.2119/molmed.2012.00077). URL: <https://doi.org/10.2119/molmed.2012.00077>.
- Henson, Michael (2013). “Multicellular models of intercellular synchronization in circadian neural networks”. In: *Chaos, Solitons & Fractals* 50, 48–64. DOI: [10.1016/j.chaos.2012.11.008](https://doi.org/10.1016/j.chaos.2012.11.008).
- St. John, Peter C., Tsuyoshi Hirota, Steve A. Kay, and Francis J. Doyle (2014). “Spatiotemporal separation of PER and CRY posttranslational regulation in the mammalian circadian clock”. In: *Proceedings of the National Academy of Sciences* 111.5, pp. 2040–2045. ISSN: 0027-8424. DOI: [10.1073/pnas.1323618111](https://doi.org/10.1073/pnas.1323618111). eprint: <https://www.pnas.org/content/111/5/2040.full.pdf>. URL: <https://www.pnas.org/content/111/5/2040>.
- Bertalan, Tom, Yan Wu, Carlo Laing, C. William Gear, and Ioannis G. Kevrekidis (2017). “Coarse-Grained Descriptions of Dynamics for Networks with Both Intrinsic and Structural Heterogeneities”. In: *Frontiers in Computational Neuroscience* 11. ISSN: 1662-5188. DOI: [10.3389/fncom.2017.00043](https://doi.org/10.3389/fncom.2017.00043). URL: <https://www.frontiersin.org/articles/10.3389/fncom.2017.00043>.
- Vadnie, Chelsea A. and Colleen A. McClung (2017). “Circadian Rhythm Disturbances in Mood Disorders: Insights into the Role of the Suprachiasmatic Nucleus”. In: *Neural Plasticity* 2017, p. 1504507. ISSN: 2090-5904. DOI: [10.1155/2017/1504507](https://doi.org/10.1155/2017/1504507). URL: <https://doi.org/10.1155/2017/1504507>.
- Hastings, Michael H., Elizabeth S. Maywood, and Marco Brancaccio (2018). “Generation of circadian rhythms in the suprachiasmatic nucleus”. In: *Nature Reviews Neuroscience* 19.8, pp. 453–469. ISSN: 1471-0048. DOI: [10.1038/s41583-018-0026-z](https://doi.org/10.1038/s41583-018-0026-z). URL: <https://doi.org/10.1038/s41583-018-0026-z>.
- Kemeth, Felix P., Sindre W. Haugland, Felix Dietrich, Tom Bertalan, Kevin Höhle, Qianxiao Li, Erik M. Bollt, Ronen Talmon, Katharina Krischer, and Ioannis G. Kevrekidis (2018). “An Emergent Space for Distributed Data with Hidden Internal Order through Manifold Learning”. eng. In: *IEEE access : practical innovations, open solutions* 6, pp. 77402–77413. ISSN: 2169-3536. DOI: [10.1109/access.2018.2882777](https://doi.org/10.1109/access.2018.2882777). URL: <https://pubmed.ncbi.nlm.nih.gov/31179198>.
- Huang, Shenzhen, Xinwei Jiao, Dingli Lu, Xiaoting Pei, Di Qi, and Zhijie Li (2020). “Recent advances in modulators of circadian rhythms: an update and perspective”. In: *Journal of Enzyme Inhibition and Medicinal Chemistry* 35.1, pp. 1267–1286. DOI: [10.1080/14756366.2020.1772249](https://doi.org/10.1080/14756366.2020.1772249). eprint: <https://doi.org/10.1080/14756366.2020.1772249>. URL: <https://doi.org/10.1080/14756366.2020.1772249>.
- Thiem, Thomas, Mahdi Kooshkbaghi, Tom Bertalan, Carlo Laing, and Yannis Kevrekidis (2020). “Emergent Spaces for Coupled Oscillators”. In: *Frontiers in Computational Neuroscience* 14. DOI: [10.3389/fncom.2020.00036](https://doi.org/10.3389/fncom.2020.00036).
- Thiem, Thomas N., Felix P. Kemeth, Tom Bertalan, Carlo R. Laing, and Ioannis G. Kevrekidis (2021). “Global and local reduced models for interacting, heterogeneous agents”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.7, p. 073139. DOI: [10.1063/5.0055840](https://doi.org/10.1063/5.0055840).

eprint: <https://doi.org/10.1063/5.0055840>. URL: <https://doi.org/10.1063/5.0055840>.

- Kemeth, Felix P., Tom Bertalan, Thomas Thiem, Felix Dietrich, Sung Joon Moon, Carlo R. Laing, and Ioannis G. Kevrekidis (2022). "Learning emergent partial differential equations in a learned emergent space". In: *Nature Communications* 13.1, p. 3318. ISSN: 2041-1723. DOI: [10.1038/s41467-022-30628-6](https://doi.org/10.1038/s41467-022-30628-6). URL: <https://doi.org/10.1038/s41467-022-30628-6>.
- Ma, Melinda A and Elizabeth H Morrison (2022). "Neuroanatomy, Nucleus Suprachiasmatic". en. In: *StatPearls*. Treasure Island (FL): StatPearls Publishing.

## Chapter 4

# Optimizing Reactors under Dynamic Operation

Work in this Chapter was done in collaboration with Prof. Mihalis Kavousanakis, Prof. Paul J. Dauenhauer and Prof. Ioannis Kevrekidis. The corresponding manuscript (under preparation) is “Computations and Optimization for Programmable Catalysts”.

### 4.1 Motivation

Rate enhancement of heterogeneous catalytic reactors remains a focus in Chemical Engineering research and industry, as it translates to paramount reduction in costs and environmental footprint (Ardagh et al., 2019). It is particularly important to sustainability (e.g. carbon dioxide as feedstock (Rodriguez et al., 2015; Dunwell et al., 2018; Vogt et al., 2019), pollutant degradation (Deka et al., 2013; Fu et al., 2022)), basic chemicals (e.g. ammonia (Singh et al., 2018; Foster et al., 2018), monomers (Stadler et al., 2019)) and energy (e.g. methane production (Vogt et al., 2019), biofuels (Friend and Xu, 2017)). In many cases, the expensive catalysts (e.g. rare earth, noble metals) or extreme operating conditions required for these processes are responsible for a significant percentage of the final cost.

One of the most common approaches in improving heterogeneous catalysis is circumventing or removing limiting processes which constrain the overall reaction rate. For example, the Sabatier

principle describes how either the surface reaction or the desorption step can be limiting, depending on the binding strength between catalyst and adsorbates (Chianelli et al., 2002; Foster et al., 2018; Gathmann, Ardagh, and Dauenhauer, 2022). This trade-off is usually depicted with the so-called volcano plots (Ichikawa, 1990) and naturally implies an optimum, i.e. a catalyst best suited for each reaction. However, this applies to catalysts operating in *static conditions*, i.e. their activity does not change over time. On the contrary, *dynamic* operation of catalytic reactors (and specifically, periodic) can overcome these limitations and lead to great enhancement of the turnover frequency (Ardagh, Abdelrahman, and Dauenhauer, 2019; Ardagh et al., 2020; Ardagh et al., 2019), when tuned properly. Dynamic operation can be achieved through various mechanisms: oscillating feed concentrations (Tomita and Kai, 1979), temperature (Qiao et al., 2008), surface potential (Shetty et al., 2020; Lim, Hülsey, and Yan, 2021), dynamical strain (Wittreich et al., 2022) or light illumination (Lovelett et al., 2021; Sordello et al., 2021; Qi et al., 2020).

Operating a reactor in periodic conditions goes way back: It has been systematically studied in the trail-blazing work of Horn (Horn and Lin, 1967) and Bailey (Bailey, 1974), who introduced the mathematical methodology for it. Tomita and Kai (Tomita and Kai, 1979) explored the response of periodically forced chemical kinetics to different forcing parameters (such as the amplitude and frequency of the signal) and constructed the relevant bifurcation diagrams. Kevrekidis *et al.* (Kevrekidis, Schmidt, and Aris, 1999; Kevrekidis, Aris, and Schmidt, 1986b; Aronson et al., 1986; Kevrekidis, Aris, and Schmidt, 1986a) have rigorously explored the dynamics of periodically forced reactors and constructed detailed bifurcation diagrams for representative cases of kinetics (autocatalytic reactions, Langmuir-Hinshelwood kinetics) and forcing parameters (concentration, temperature). Their work deals with dynamical systems that exhibit oscillations even *without forcing* (i.e. autonomous periodic response), yet the analysis and algorithms used are of more general interest. With more recent advances in experimental methods, (specifically, precise control of the forcing schedule and real-time measurement of the response) computational work was validated against experimental data. Specifically it was shown (Wolff et al., 2003a; Wolff et al., 2003b; Qiao et al., 2008) how periodically forcing a Pt catalyst can lead to significant enhancement of the overall CO oxidation rate. In that case, the forcing was introduced through a localized temperature increase induced by a laser beam. When the laser beam visits each point on the surface on a periodic schedule, steady state oscillations arise.

In the last years, Dynamic Catalysis has gathered renewed interest due to new modeling approaches and experimental capabilities. Specifically, Dauenhauer *et al.* have shown computationally (Ardagh, Abdelrahman, and Dauenhauer, 2019; Ardagh et al., 2020; Ardagh et al., 2019) and experimentally

(Shetty et al., 2020; Gopeesingh et al., 2020) that periodic operation can be achieved by imposing an electrodynamic oscillation on the catalytic surface which causes the binding energies (and consequently, all species' concentrations) to oscillate. When oscillating between two states on either side of the Sabatier optimum, it is possible to continuously “switch” between limiting processes, which, when tuned properly, can greatly improve efficiency.

In this work, we present an algorithmic toolbox to efficiently calculate the response of reactors to external periodic forcing. With the assistance of it, we explore how different forcing parameters affect the reactor's response. This allows us to understand how to optimally tune the extra degrees of freedom that arise when introducing the forcing. It also enables to locate and understand optimal forcing function shapes.

## 4.2 Case studies

In general, we consider two chemical reaction schemes, and three possible reactor configurations:

**Systems:** Unimolecular surface reaction, Parallel surface reaction

**Reactors:** Constant gas pressure reactor, Batch reactor, CSTR

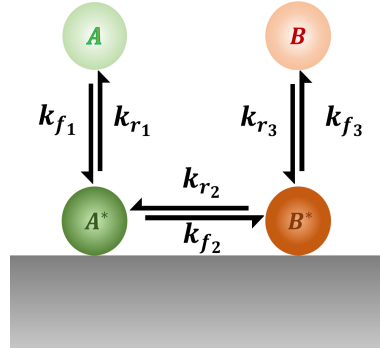
Out of all six combinations, three are considered in this manuscript, which are described in the following.

### Pseudo-steady state unimolecular reaction

First, a simplified 2-dimensional surface catalysis model will be studied. This model is adapted and modified from (Ardagh, Abdelrahman, and Dauenhauer, 2019) assuming constant partial pressures for all species. Therefore, the resulting dynamical system consists of two ODEs describing the time evolution of the surface coverages of species A and B:

$$\frac{d\theta_A}{dt} = k_{f_1} P_A \theta^* - (k_{r_1} + k_{f_2}) \theta_A + k_{r_2} \theta_B \quad (4.1)$$

$$\frac{d\theta_B}{dt} = k_{r_3} P_B \theta^* - (k_{f_3} + k_{r_2}) \theta_B + k_{f_2} \theta_A, \quad (4.2)$$



**Figure 4.1:** Schematic with reaction steps and corresponding rate constants.  $A, B$  are the gas species and  $A^*, B^*$  the adsorbed species.

where  $\theta^* = 1 - \theta_A - \theta_B$  denotes the coverage of empty sites,  $P_A = 99\text{bar}$  is the constant partial pressure of  $A$  and  $P_B = 1\text{bar}$  is the constant partial pressure of  $B$ . Let  $\mathbf{x} = [\theta_A, \theta_B]$  denote the state vector,  $\mathbf{k} = [k_{f_1}, k_{f_2}, k_{f_3}, k_{r_1}, k_{r_2}, k_{r_3}]$  the kinetic parameters vector and  $\mathbf{f}(\mathbf{x}(t); t, \mathbf{k})$  the corresponding vectorfield described by Eqs. 4.1, 4.2.

Most kinetic parameters are a function of the relative binding energy of  $B$  ( $\Delta BE_B$ ), as described in (Ardagh, Abdelrahman, and Dauenhauer, 2019), and in detail, in the following:

The reaction rate constants for all forward elementary reactions shown in Fig. 4.1 are defined as:

$$k_{f_1} = 10^6 e^{-E_{a_{f_1}}/RT_r}, \quad k_{f_2} = 10^{13} e^{-E_{a_{f_2}}/RT_r}, \quad k_{f_3} = 10^{13} e^{-E_{a_{f_3}}/RT_r},$$

and for the respective reverse elementary reactions:

$$k_{r_1} = k_{f_1}/K_1, \quad k_{r_2} = k_{f_2}/K_2, \quad k_{r_3} = k_{f_3}/K_3,$$

where the equilibrium constants are defined as:

$$K_1 = 10^{-7} e^{-\Delta H_1/RT_r}, \quad K_2 = e^{-\Delta H_2/RT_r}, \quad K_3 = 10^7 e^{-\Delta H_3/RT_r},$$

and the activation energies for the forward steps:

$$E_{a_{f_1}} = 0\text{J}, \quad E_{a_{f_2}} = \max\{\beta + \alpha\Delta H_2, \Delta H_2, 0\}, \quad E_{a_{f_3}} = \Delta H_3$$



while the corresponding enthalpies:

$$\Delta H_1 = -BE_A, \Delta H_2 = \Delta H_{ovr} + BE_A - BE_B, \Delta H_3 = BE_B$$

Finally the Binding energies for both  $A$  and  $B$  are:

$$BE_A = \max\{BE_{A_0} + dCorr_A \Delta BE_B(t), 0\}, BE_B = \max\{BE_{B_0} + dCorr_B \Delta BE_B(t), 0\}$$

Constant	Value	Units	Description
$T_r$	423.15	K	Reactor Temperature
$R$	8.314	J/mol/K	Ideal gas constant
$\Delta H_{ovr}$	-20	kJ/mol	Overall reaction enthalpy
$\alpha$	0.8	-	BEP linear-scaling parameter
$\beta$	102	kJ	BEP offset
$BE_{A_0}$	1.3	eV	Reference Binding Energy of A
$BE_{B_0}$	1	eV	Reference Binding Energy of B
$dCorr_A$	0.5	-	Proportionality constant for A
$dCorr_B$	1	-	Proportionality constant for B
$P_A$	99	bar	Reactor partial pressure of A
$P_B$	1	bar	Reactor partial pressure of B
$w_{cat}$	200	mg	Catalyst mass
$\rho_{sites}$	$2 \cdot 10^{-5}$	mol/g	Density of catalytic sites per catalyst weight

**Table 4.1:** Parameters used to simulate the system of Eqs.4.1, 4.2, depicted in Fig. 4.1. Note, that “BEP” stands for Brønsted-Evans-Polyani principle (Akhade et al., 2018).

Note, that from Table 4.1 we have  $N_{sites} = \rho_{sites} \cdot w_{cat}$  (after appropriate conversions)

We will consider the case were oscillations in  $\Delta BE_B$  are enforced externally and the dynamical system becomes non-autonomous as a result. Specifically, the value of  $\Delta BE_B$  will change according to a step forcing function (see Fig. 4.2).

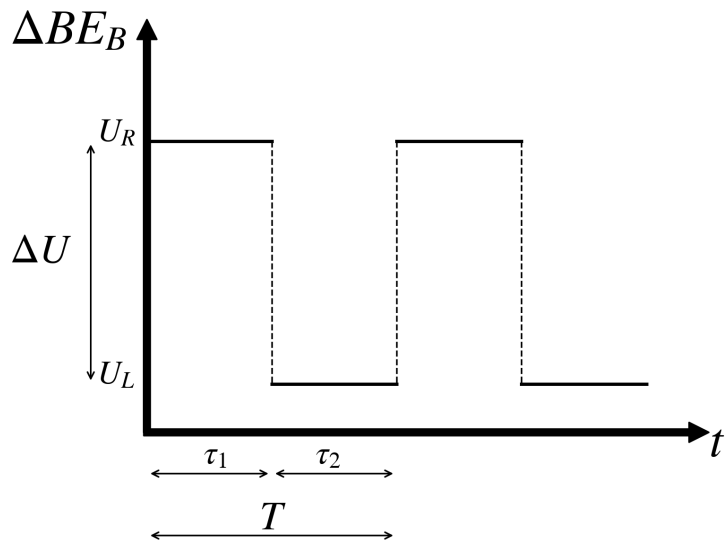


Figure 4.2: Step forcing function

As it can be deduced from Fig.4.2 any step forcing function has four degrees of freedom (equivalently, is a four-dimensional parameter family of shapes). Here, we will choose the following parameters for the parametrization:

- Forcing Frequency  $f = \frac{1}{T}$
- Duty cycle  $\phi = \frac{\tau_1}{T}$
- Amplitude  $\Delta U = U_R - U_L$
- Lower oscillation end-point  $U_L$

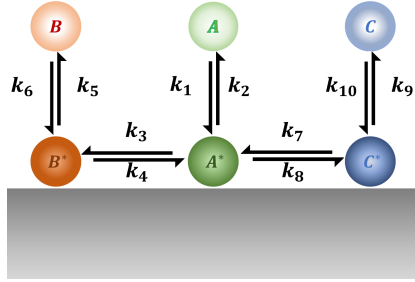
The analytical expression of the step forcing function would be:

$$\Delta BE_B(t) = \begin{cases} U_R( := U_L + \Delta U) & 0 \leq t \bmod T < \phi T \\ U_L & \phi T \leq t \bmod T < T \end{cases}$$

Denoting  $\mathbf{p} = [f, \phi, \phi, \Delta U, U_L]$  we can write  $\Delta BE_B = \Delta BE_B(\mathbf{p})$  or  $\mathbf{k} = \mathbf{k}(\mathbf{p})$ .

#### Batch reactor - parallel reaction pathway

Next, the ODEs and parameters used to model a parallel surface reaction pathway in a batch reactor are presented.



**Figure 4.3:** Schematic with reaction steps and corresponding rate constants.  $A, B, C$  are the gas species and  $A^*, B^*, C^*$  the adsorbed species.

$$\frac{dC_A}{dt} = -k_1 \frac{N_{sites}}{V} p_A \theta^* + k_2 \frac{N_{sites}}{V} \theta_A \quad (4.3)$$

$$\frac{dC_B}{dt} = -k_6 \frac{N_{sites}}{V} p_B \theta^* + k_5 \frac{N_{sites}}{V} \theta_B \quad (4.4)$$

$$\frac{dC_C}{dt} = -k_{10} \frac{N_{sites}}{V} p_C \theta^* + k_9 \frac{N_{sites}}{V} \theta_C \quad (4.5)$$

$$\frac{d\theta_A}{dt} = k_1 p_A \theta^* - (k_2 + k_3 + k_7) \theta_A + k_4 \theta_B + k_8 \theta_C \quad (4.6)$$

$$\frac{d\theta_B}{dt} = k_6 p_B \theta^* + k_3 \theta_A - (k_4 + k_5) \theta_B \quad (4.7)$$

$$\frac{d\theta_C}{dt} = k_{10} p_C \theta^* + k_7 \theta_A - (k_8 + k_9) \theta_C, \quad (4.8)$$

where  $p_X$  is the partial pressure of species  $X$ ,  $N_{sites}$  denotes the moles of available catalytic sites and  $V$  the reactor volume. The state vector now becomes  $\mathbf{x} = [C_A, C_B, C_C, \theta_A, \theta_B, \theta_C]$  and the parameter vector  $\mathbf{k} = [k_1, \dots, k_{10}]$ . Note, that in this case it is the relative binding energy of A oscillating, therefore:

$$\mathbf{k} = \mathbf{k}(\Delta BE_A)$$

All components of  $\mathbf{k}$ , as shown in Fig. 4.3 are defined as:

$$k_1 = \frac{k_B}{T_r h} e^{\Delta S_1 / R},$$

$$k_3 = \frac{k_B}{T_r h} e^{-E_{a_1}/RT_r},$$

$$k_6 = \frac{k_B}{T_r h} e^{-\Delta S_3/R},$$

$$k_7 = \frac{k_B}{T_r h} e^{-E_{a_2}/RT_r},$$

$$k_{10} = \frac{k_B}{T_r h} e^{-\Delta S_5/R},$$

$$k_2 = k_1/K_1, \quad k_4 = k_3/K_2, \quad k_5 = k_6 K_3, \quad k_8 = k_7 K_4, \quad k_9 = k_{10} K_5,$$

where the equilibrium constants are defined as:

$$K_i = e^{-\Delta G_i/RT_r}, \text{ where: } \Delta G_i = \Delta H_i - T_r \Delta S_i, \quad i = 1, \dots, 5$$

and the activation energies:

$$E_{a_{f_i}} = \max\{\beta_i + \alpha_i \Delta H_{2i}, \Delta H_{2i}, 0\}, \quad i = 1, 2$$

while the enthalpies:

$$\Delta H_1 = -BE_1, \quad \Delta H_3 = BE_2, \quad \Delta H_5 = BE_3$$

$$\Delta H_2 = (H_2 - BE_2) - (H_1 - BE_1),$$

$$\Delta H_4 = (H_3 - BE_3) - (H_1 - BE_1).$$

Finally, the Binding Energies:

$$BE_1 = \max\{BE_{A_0} + \Delta BE_A, 0\}$$

$$BE_j = \max\{H_j + \gamma(BE_1 - H_j) - (1 - \gamma_j)\delta_j, 0\}, \quad j = 2, 3$$

All necessary parameters are mentioned in the Table 4.2

Constant	Value	Units	Description
$T_r$	423.15	K	Reactor Temperature
$R$	8.314	J/mol/K	Ideal gas constant
$\alpha$	[0.6, 0.6]	–	BEP linear-scaling parameters
$\beta$	[100, 100]	kJ/mol	BEP offsets
$\gamma$	[2, 0.5]	–	BE proportionality constants
$\delta$	[–1.4, –1.4]	eV	BE common point
$BE_{A_0}$	1	eV	Reference Binding Energy of A
$H_i$	[0, 0, 0]	kJ/mol	Product Enthalpies
$\Delta S_i$	[–135, 0, 135, 0, 135]	kJ/mol/K	Reaction Entropies
$V$	0.0002603	L	Reactor volume
$h$	$6.6210^{-34}$	J/Hz	Planck constant
$k_B$	$1.3810^{-23}$	J/K	Boltzmann constant
$N_{sites}$	$2.7610^{-6}$	mol	Empty catalytic sites
$p_{A_0}$	100	bar	Initial partial gas pressure of A

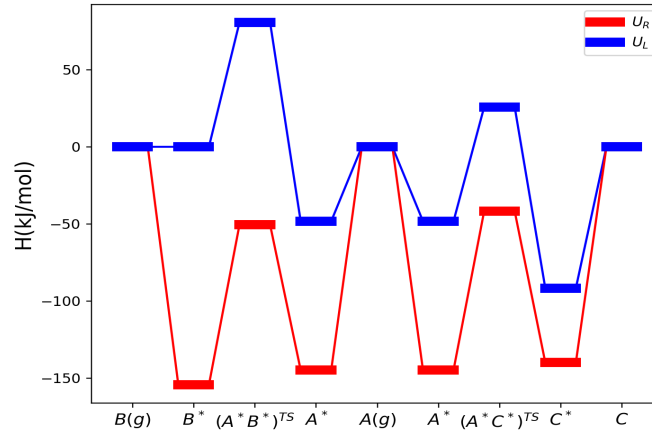
**Table 4.2:** Parameters used to simulate the system of Eqs.4.3 - 4.8 depicted in Fig. 4.3. Note, that “BE” stands for Binding Energies, while “BEP” stands for Brønsted-Evans-Polyani principle (Akhade et al., 2018)

Note that not all ODEs in the above system (Eqs. 4.3-4.8) are independent, due to the additional equation for conservation of mass in the batch reactor:

$$N_t = V(C_A + C_B + C_C + N_{sites}(\theta_A + \theta_B + \theta_C)), \quad (4.9)$$

where  $N_t$  is the constant total moles per reactor volume.

To make our system non-singular (Buckingham, 1914; Aris and Mah, 1963), we remove one of the ODEs from 4.3-4.8 and replace the respective variable through the mass balance (Eq. 4.9). Here, we choose to replace  $C_B$  through Eq.4.9, as it has much different range of values than the rest of the variables. We end up with a well-defined system of 5 ODEs and a reduced state vector  $\mathbf{x}_r = [C_A, C_C, \theta_A, \theta_B, \theta_C]$ . Here, a similar forcing mechanism is introduced, as in the unimolecular surface reaction system, i.e. the forcing is introduced to the kinetic parameters through a step forcing in the relative binding energy of A ( $\Delta BE_A$ ). An example of the energy diagram for  $U_R = 0.5\text{eV}$ ,  $U_L = -0.5\text{eV}$ .



**Figure 4.4:** Energy diagram of the two states of the step forcing function. The state colored red  $U_R$  favors the surface reaction step, while the blue colored state  $U_L$  favors desorption.

#### CSTR - parallel reaction pathway

Lastly, a parallel surface reaction pathway in a CSTR is also considered, adapted from (Ardagh et al., 2020):

$$\frac{dC_A}{dt} = \frac{\dot{q}}{V}(C_A - C_{A_{feed}}) - k_1 \frac{N_{sites}}{V} p_A \theta^* + k_2 \frac{N_{sites}}{V} \theta_A \quad (4.10)$$

$$\frac{dC_B}{dt} = \frac{\dot{q}}{V}(C_B - C_{B_{feed}}) - k_6 \frac{N_{sites}}{V} p_B \theta^* + k_5 \frac{N_{sites}}{V} \theta_A \quad (4.11)$$

$$\frac{dC_C}{dt} = \frac{\dot{q}}{V}(C_C - C_{C_{feed}}) - k_{10} \frac{N_{sites}}{V} p_C \theta^* + k_9 \frac{N_{sites}}{V} \theta_C \quad (4.12)$$

$$\frac{d\theta_A}{dt} = k_1 p_A \theta^* - (k_2 + k_3 + k_7) \theta_A + k_4 \theta_B + k_8 \theta_C \quad (4.13)$$

$$\frac{d\theta_B}{dt} = k_6 p_B \theta^* + k_3 \theta_A - (k_4 + k_5) \theta_B \quad (4.14)$$

$$\frac{d\theta_C}{dt} = k_{10} p_C \theta^* + k_7 \theta_A - (k_8 + k_9) \theta_C, \quad (4.15)$$

where  $\dot{q}$  is the volumetric flow rate and  $C_{X_{feed}}$  is the feed concentration of species X. As in (Ardagh et al., 2020) we consider the case of 1% period-averaged conversion of A. To satisfy this additional constraint, we let the volumetric flow rate  $\dot{q}$  vary. In fact, we can also solve for the volumetric flow rate with the following modification: We can augment the system of ODEs (Eqs. 4.10-4.15) by:

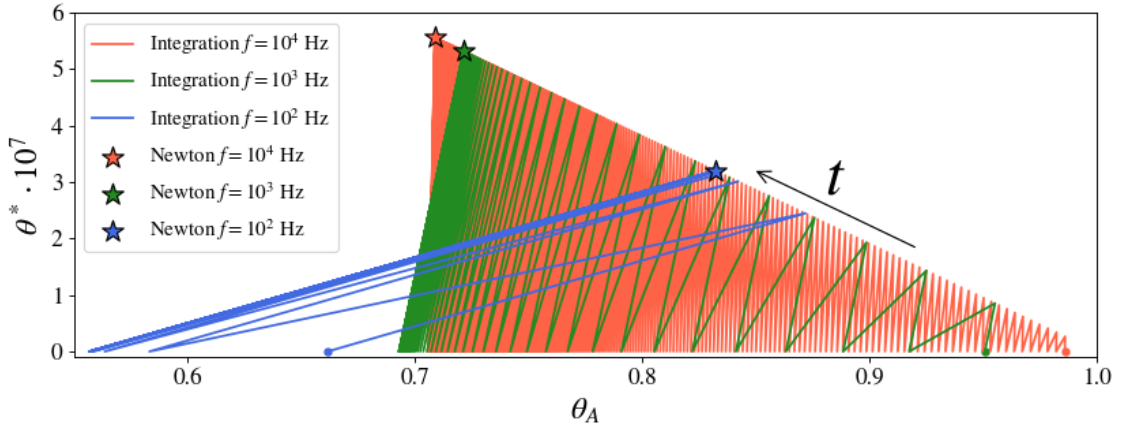
$$\frac{dc}{dt} = \frac{1}{T} \frac{C_{A_{feed}} - C_A}{C_{A_{feed}}}, \quad (4.16)$$

where  $T$  is the forcing period. After integrating this ODE (along with Eqs. 4.10-4.15) the result is:  $c(T) = \frac{C_{A_{feed}} - \frac{1}{T} \int_0^T C_A(t) dt}{C_{A_{feed}}}$ , which is the period averaged conversion of A. All parameters are described in detail in Ardagh et al., 2020. Note, that in this case it is the relative binding energy of A oscillating, therefore:  $\mathbf{k} = \mathbf{k}(\Delta BE_A)$

## 4.3 Discovering periodic steady states

### 4.3.1 Solver efficiency

One can discover steady states by brute-force integration of the system's ODEs, until convergence to a stable periodic steady-state (e.g. by tracking the state vector at a fixed phase, as in (Ardagh et al., 2019)). Alternatively, one can employ the variational approach and *solve* for the limit cycles with Newton's method as fixed points of the stroboscopic map (see Subchapter 2.1.1.1). To compare these two approaches we search for periodic steady states (with the same accuracy) initializing from the same initial state (unloaded catalytic surface) and track some relevant performance metrics. The results can be seen in Fig.4.5. This is performed for three frequency values ( $f = 10^2, 10^3, 10^4$  Hz). These values were chosen *a posteriori* as representative values for the entire range of behaviors.



**Figure 4.5:** Locating periodic steady states with brute-force integration vs. with Newton’s method (variational formulation), for three values of forcing frequencies (colored differently). Integration converging to the same periodic steady state as Newton’s method (denoted as stars). For the integration only the endpoints of each phase are plotted, after the first phase. Note that the rest of the forcing parameters are fixed at  $\phi = 0.5, U_L = -0.1eV, \Delta U = 0.6eV$ .

Frequency (Hz)	$t_{Newton}/t_{Integration}$	Integration Periods
$10^2$	1.32	21
$10^3$	0.20	266
$10^4$	0.03	2164

**Table 4.3:** Performance comparison between brute-force integration and Newton’s method (with the variational approach), for the cases shown in Fig. 4.5.

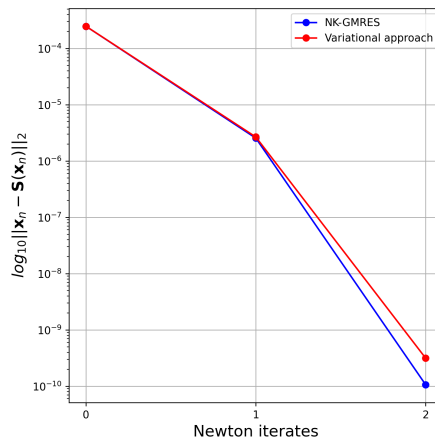
Fig.4.5 and Table 4.3 provide useful information both for algorithmic performance and for the system’s dynamics. To start with, all integration trajectories (Fig.4.5) start from  $(\theta_A, \theta^*) = (0, 1)$  (off plot limits), i.e. with no adsorbates on the catalyst’s surface. These two variables were chosen for better visualization. Upon initialization (from the first phase of the first oscillation), the catalyst rapidly loses most of its empty sites. Following that, the rate of convergence to the periodic steady state strongly depends on the forcing frequency. For small frequencies (equivalently, large periods), such as  $f = 10^2$  Hz, integration converges after a few periods. With increasing forcing frequencies (equivalently, smaller periods), the integration time required grows exponentially (see Table 4.3). This can be understood in terms of the dynamics of each phase: when each phase lasts longer the system has adequate time to move towards the respective steady state before switching to the other phase. This directly affects the time that brute-force integration takes to converge. On the contrary, Newton’s method requires just a couple of iterates for convergence, and the time required is not greatly affected by the forcing



frequency. Therefore, Newton’s method can be much more efficient, especially for larger frequencies (taking, for example no more than 4% of the time compared to brute-force integration, as shown in Table 4.3). It is also important to note, that brute-force integration can locate *only stable* periodic steady states, while Newton’s method (depending on initialization) can also locate unstable steady states. This makes Newton’s method suitable for stability and/or bifurcation analysis. The linear model we are considering here, however, does not have any unstable periodic steady states.

### 4.3.2 Newton-Krylov GMRES

We also present the Newton-Krylov GMRES approach described in Subchapter 2.1.1.2 as an alternative to the variational approach (Subchapter 2.1.1.1). As discussed in Subchapter 2.1.1, this alternative is used to demonstrate that our methodology can be extended to high-dimensional systems while avoiding prohibitive memory requirements.



**Figure 4.6:** Newton convergence using Newton’s method with the variational approach vs. Newton Krylov-GMRES  $f \approx 6667\text{Hz}$ ,  $\phi = 0.50$ ,  $U_L = -1.46\text{eV}$ ,  $\Delta U = 0.6\text{eV}$

NK-GMRES convergence seems quadratic as is the variational approach while the converged solutions agree within the range of numerical accuracy. In fact the iterates of both methods seem consistent, taking into account the differences in the numerical implementation of each. In general, NK-GMRES is not expected to converge as well in general, as this is an *inexact* Newton’s method (Kelley, 2003).

### 4.3.3 Solving for conversion constraints

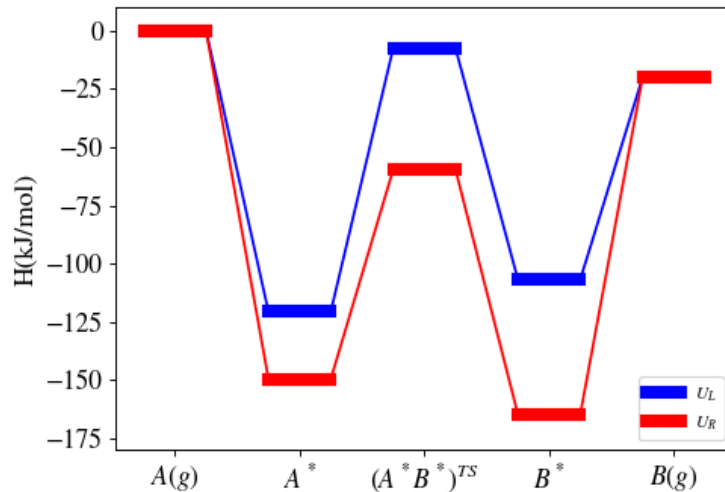
In the case of the CSTR with parallel reaction pathway, the solution the solver finds, needs to satisfy the 1% conversion constraint:  $c(T) - 0.01 = 0$ , where  $c(T)$  is defined by the Eq.4.16. As mentioned, to that end, we let the volumetric flow rate  $\dot{q}$  vary. As a result, the solver is asked to find roots to the following set of equations:

$$\mathbf{R}(\mathbf{x}; \dot{q}) = \mathbf{0} \quad (4.17)$$

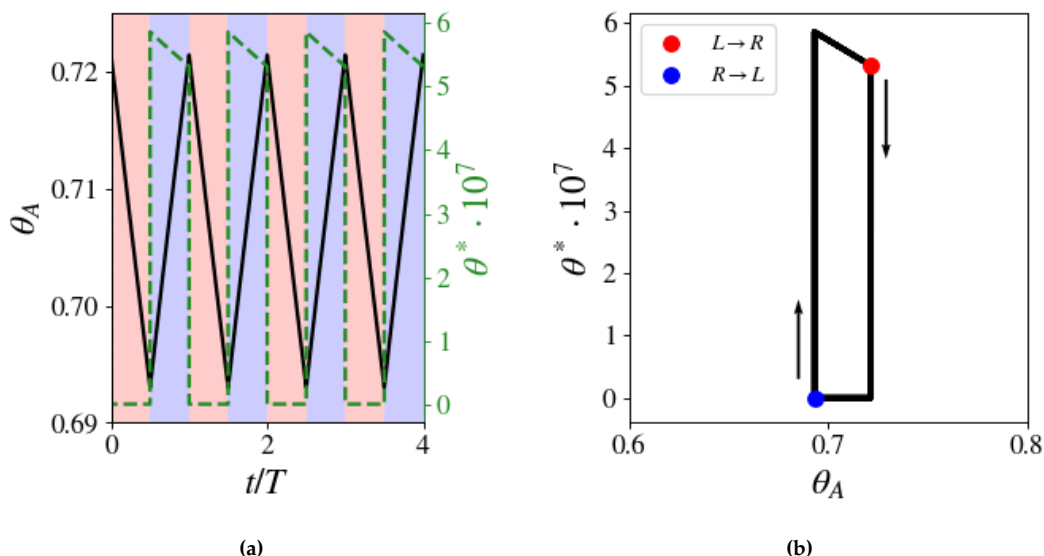
$$c(T) - 0.01 = 0 \quad (4.18)$$

Let  $\mathbf{y} \in \mathbb{R}^7$  be the vector of unknown quantities:  $\mathbf{y} = [\mathbf{x}, \dot{q}]$ . The challenge remaining, is to define the Jacobian of the system Eqs.4.17, 4.18 to be used in Newton's method (Eq. 2.1). Jacobian entries of the form  $\frac{\partial R_i(\mathbf{x})}{\partial x_j}$  can be calculated by the variational approach presented in Subchapter 2.1.1.1. That approach is also valid for  $\frac{\partial R_i}{\partial c}$ ,  $\frac{\partial c(T)}{\partial x_j}$ ,  $\frac{\partial c(T)}{\partial c}$ . The column of the Jacobian corresponding to derivatives w.r.t.  $\dot{q}$ , can be calculated as parametric derivatives (also presented in Subchapter 2.1.1.1).

## 4.4 Understanding the effect of the forcing parameters



**Figure 4.7:** Energy diagram of the two states of the step forcing function. The state colored red  $U_R$  favors the surface reaction step, while the blue colored state  $U_L$  favors desorption.



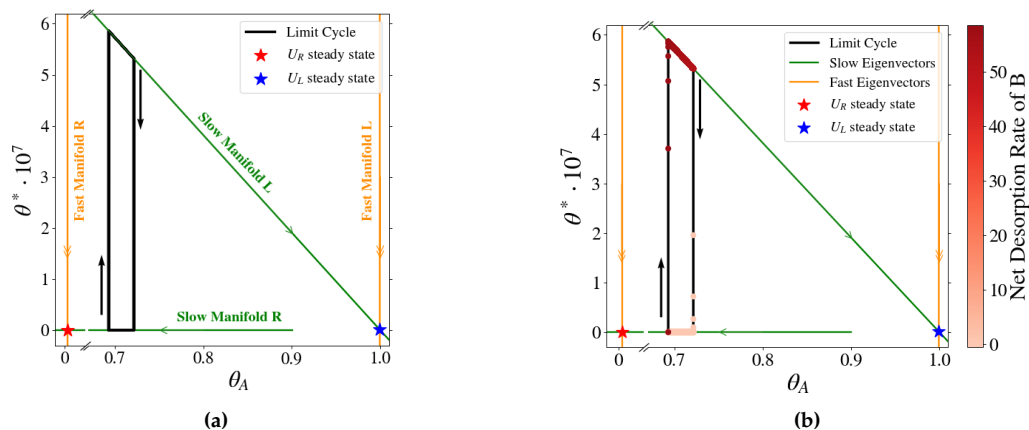
**Figure 4.8:** Oscillation of the binding energies between phases  $R$  and  $L$  lead to oscillations of all chemical species (and functions thereof): (a) Four oscillations of two of the system variables:  $\theta_A$ , the surface coverage of  $A$ , and  $\theta^*$ , the surface coverage of empty sites. Blue and red colors denote the oscillation phase (see 4.7), (b) The limit cycle point of view of the oscillations on the left. The points of phase change are denoted on the limit cycle. Note that these results are obtained for  $f = 10^3 \text{ Hz}$ ,  $\phi = 0.5$ ,  $U_L = -0.1 \text{ eV}$ ,  $\Delta U = 0.6 \text{ eV}$

All system dynamics can be described in terms of six processes: Adsorption of  $A$ , Desorption of  $A$ , Forward and Backward surface reaction, Adsorption of  $B$ , and Desorption of  $B$ . Adsorption of both species does not require activation, in contrast to the rest of the processes.

During phase  $R$ , the surface reaction energy barrier is reduced, and the desorption of  $B$  energy barrier is increased. Therefore, this phase favors the surface reaction rate. On the contrary, phase  $L$  favors the desorption of  $B$ , and the surface reaction rate is reduced. In other words, phase  $R$  induces a bottleneck on the desorption of  $B$ , while during phase  $L$  the bottleneck is shifted to the surface reaction.

The limiting process of each oscillation phase is directly reflected on the surface coverages of the chemical adsorbates (here, we choose again to study the surface coverage of  $A$  and the surface coverage of empty sites - Fig.4.8a). During phase  $R$  (red) limited desorption restricts the number of empty sites on the catalyst (low  $\theta^*$ ), while the  $A$  adsorbates are reacting to  $B$  adsorbates. During phase  $L$  (blue) desorption is no longer limiting and empty surface sites appear. At the same time, the now difficult surface reaction causes accumulation of  $A$  adsorbates (Fig. 4.8a). The limit cycle point of view is more helpful as it removes the time dependence and makes it easier to show periodic steady states (Fig.4.8b).

As seen in Fig.4.9, the static steady states for each dynamic catalysis phase provide very useful



**Figure 4.9:** Oscillations of the chemical species during Dynamic Catalysis: (a) Static steady states and their slow and fast manifolds are shown on the limit cycle figure, (b) The net desorption rate of B of a number of points along the limit cycle. These points are separated by a fixed time step.

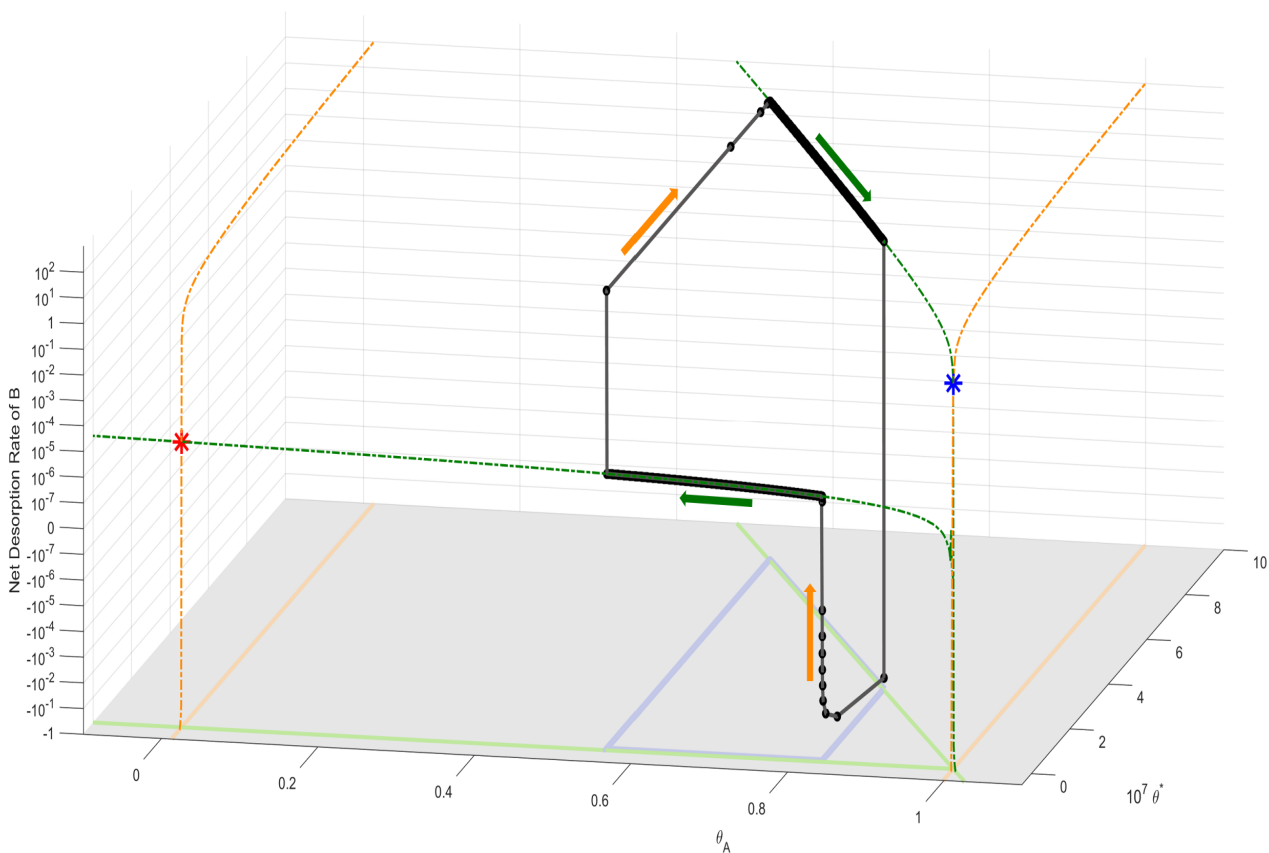
information about the limit cycle. As expected, the limit cycle assumes intermediate surface coverages w.r.t the two static steady states. Each of the equilibria described by the static steady states describes dynamics extremely favoring either adsorbate A or adsorbate B, as the rate of the entire process is restricted to the limiting rate of each static steady state. The composition of these states is shown in Table 4.4.

State	$(\theta_A, \theta^*)$ Coordinates	Eigenvalues
Phase R	$(0.0132, 1.35 \cdot 10^{-13})$	$(-10^8, -81.9)$
Phase L	$(9.994, 1.23 \cdot 10^{-9})$	$(-10^8, -194.8)$

**Table 4.4:** Static steady states and their eigenvalues.

The eigenvalues of both static steady states (Table 4.4) confirm that both of them are stable and reveal a significant separation in timescales. Knowing the eigendirections of the respective eigenvectors, we can plot the slow and fast stable manifolds for both steady states. **These manifolds inform us about the slow and fast directions on the limit cycle.** This is confirmed by the density of (equally spaced in time) points on the limit cycle (Fig.4.9b): The vast majority of points can be found where the limit cycle is moving along slow manifolds and towards static steady states. This is confirmed in Fig.4.5 where the endpoint of each oscillation close to the steady state can be consistently found on the slow manifolds (alternating between the two).

Fig. 4.9b associates different parts of the limit cycle with the net desorption rate of B. The net

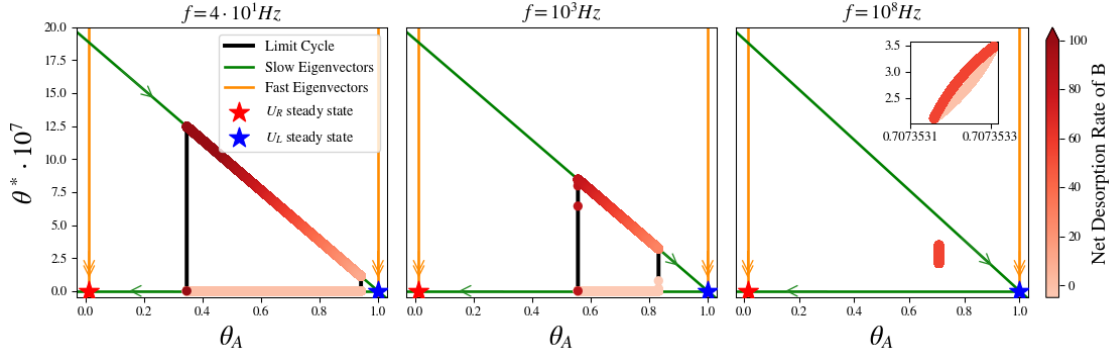


**Figure 4.10:** 3D view of the limit cycle ( $f = 10^2\text{Hz}$ ,  $\phi = 0.5$ ,  $U_L = -0.1\text{eV}$ ,  $\Delta U = 0.6\text{eV}$ ) where the Net Desorption rate of B is now the z-axis. Steady states are denoted as stars (red for phase R and blue for L). Slow manifolds are colored green and fast ones, orange. Direction arrows on different parts of the limit cycle are colored analogously. The projection of the limit cycle is seen on the floor of the 3D plot.

desorption rate of B is significantly greater during phase L. This fits with the qualitative understanding we already have about that phase: Phase L promotes desorption of B, by decreasing the energy barrier required for it. However, the closer we move to the static steady state, the lesser the net desorption rate of B. This can be clearly seen in the Fig.4.10 where the same system is presented in 3D. As a metric for each forcing parameter configuration, we will use the period-averaged net desorption rate of B, defined as:  $\overline{NDR_B} = \frac{1}{T} \int_0^T NDR_B(t) dt$ .

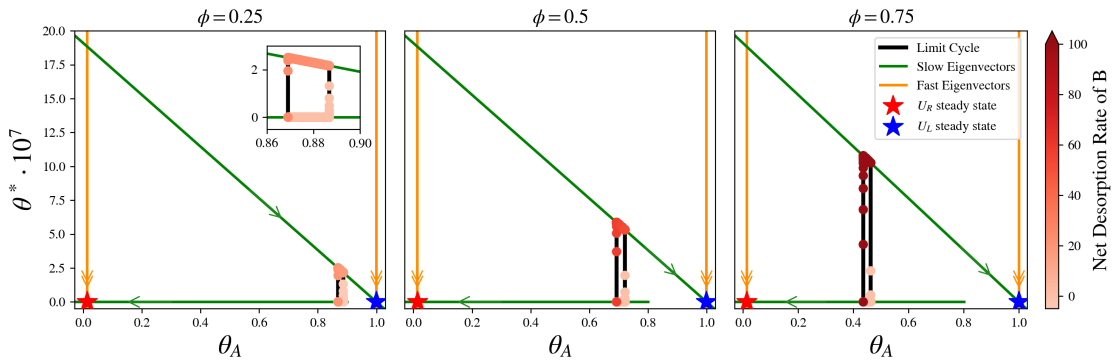
It is interesting to notice in Fig.4.10 that both the static steady states have an extremely low net desorption rate of B. This can be explained by the fact that the dynamics of each static system are fully constrained by the respective limiting process (desorption or reaction). This observation further

corroborates the benefits of Dynamic Catalysis, by juxtaposing static and dynamic performance.



**Figure 4.11:** Investigating the effect of forcing frequency: (left)  $f = 10\text{Hz}$ , (middle)  $f = 10^3\text{Hz}$ , (right)  $f = 10^8\text{Hz}$  all for  $\phi = 0.5$ ,  $U_L = -0.1\text{eV}$ ,  $\Delta U = 0.6\text{eV}$ .

First we explore the effect of the forcing frequency on the dynamics. Fig.4.11 shows the effect of extreme values of the forcing frequency. In the case of great reduction of frequency (left panel in Fig.4.11), the system has enough time to move towards the respective static steady state of each phase. As a result, the system spends considerable time on the slow manifolds, especially close to the static steady states (where the dynamics get exponentially slow). Even if the system reaches higher net desorption rates of B, it stays more time in less productive states and, therefore, the result is a lower period-averaged net desorption rate of B ( $\overline{NDR}_B = 23.2\text{s}^{-1}$ ). The characteristic times of the slow manifolds confirm this conclusion ( $2f < \lambda_{L,slow}, 2f < \lambda_{R,slow}$ ). In the case of great increase of frequency (right panel in Fig.4.11) the extremely short oscillation periods do not allow the system to reach the slow manifolds. In fact, the oscillation periods are even shorter than the fast characteristic times of both static steady states ( $2f > \lambda_{L,fast}, 2f > \lambda_{R,fast}$ ). This means that the system doesn't even have time to move along the fast manifolds. However, the period-averaged net desorption rate of B is not affected ( $\overline{NDR}_B = 27.7\text{s}^{-1}$ ) compared to the lower frequencies, such as that in the middle panel of Fig.4.11 ( $\overline{NDR}_B = 27.7\text{s}^{-1}$ ).



**Figure 4.12:** Investigating the effect of the duty cycle frequency: (left)  $\phi = 0.25$ , (middle)  $\phi = 0.5$ , (right)  $\phi = 0.75$  all for  $f = 10^3 \text{ Hz}$ ,  $U_L = -0.1 \text{ eV}$ ,  $\Delta U = 0.6 \text{ eV}$

Fig.4.12 shows the effect of changing the duty cycle  $\phi$  while keeping all the other forcing parameters fixed. In the case where phase R lasts longer (right panel in Fig.4.12) the limit cycle is located closer to the static steady state of phase R (where the dynamics are slower). Necessarily, during phase L, the limit cycle is further away from the static steady state of phase L, which means higher net desorption rate of B. However, this highly productive state doesn't last long and it is on average worse ( $\overline{NDR_B} = 26.0 \text{ s}^{-1}$ ) than the case of equally lasting phases ( $\overline{NDR_B} = 27.7 \text{ s}^{-1}$ ). On the contrary, in the case when L phase lasts longer (left panel of Fig.4.12), the opposite effect is observed: the limit cycle approaches the static steady state of phase L but now neither phase is productive enough to make a difference ( $\overline{NDR_B} = 17.3 \text{ s}^{-1}$ ).

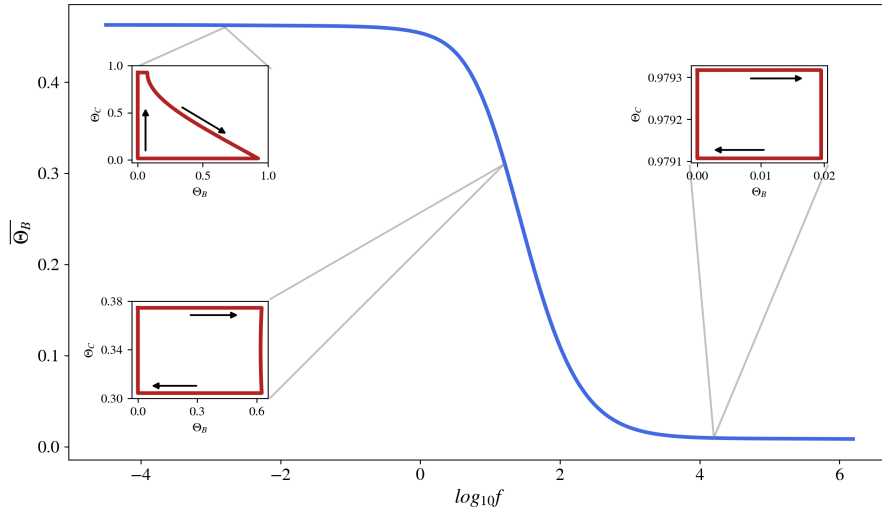
In general, when changing the duty cycle  $\phi$  a very interest trade-off arises: more productive states (such as those in the right panel of Fig.4.12) can be only reached for a shorter time duration, so that the system always keeps a "safe distance" from the L static steady state. This behavior indicates the existence of an optimum w.r.t. the duty cycle.

Similar computational examples can be performed to study the effects of the amplitude  $\Delta U$  and the lower oscillation endpoint ( $U_L$ ). Note that when comparing such cases, the steady states, their eigenvalues and stable manifolds would be different every time.

## 4.5 Continuation

Pseudo-arclength continuation (see Subchapter 2.1.2) allows the efficient exploration of behaviors along a large parameter range. Here we employ the commercial software AUTO 07p (Doedel et al., 2007;

Doedel et al., 1999) to explore how the batch reactor described in Subchapter 4.2 responds to different forcing frequencies.



**Figure 4.13:** Continuation of periodic steady states w.r.t the forcing frequency using the software AUTO 07p (Doedel et al., 1999). Inset figures show representative projections of the limit cycles. All other forcing parameters were kept fixed at  $\phi = 0.50$ ,  $U_L = -0.5eV$ ,  $\Delta U = 1eV$ .

In accordance to the analysis in Subchapter 4.4 we observe a strong dependence of the resulting dynamics to the forcing frequency. As the projections of the limit cycles suggest, at very low frequencies (or high periods) the system has adequate time to move towards the static steady states, and that's why the catalytic surface ends up being covered with either one of the two adsorbed products at each phase. On the contrary, when the oscillations are too fast, the surface coverages don't change by a lot, and in fact, the dominant surface species is the one that is most difficult to get desorbed "on average" (i.e. species C, see Fig. 4.4).

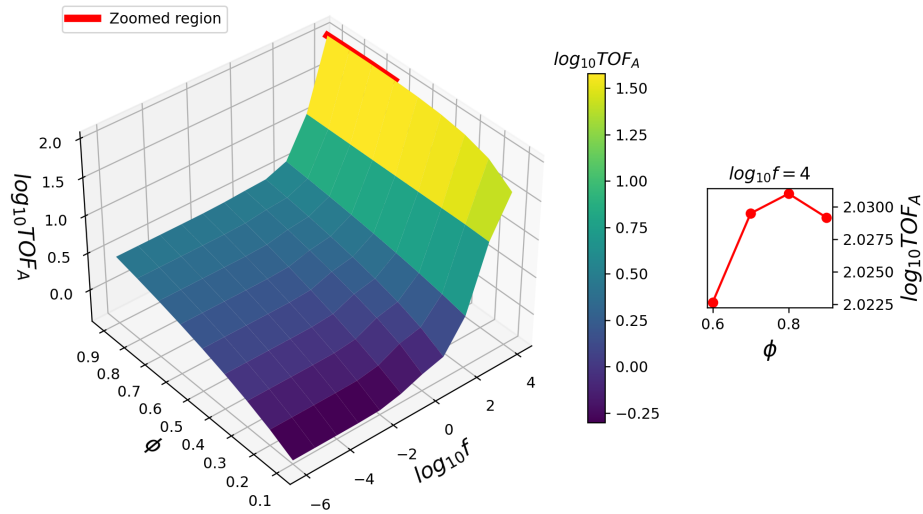
Next, we perform continuation for a CSTR reactor with a parallel reaction pathway (described in Subchapter 4.2). In the case of a CSTR reactor turnover frequencies can be used to quantify performance:

$$TOF_X(\mathbf{p}) = \frac{\dot{q}}{TN_{sites}} \int_0^T C_X(t; \mathbf{p}) dt, \quad (4.19)$$

for any species X. In Eq. 4.5,  $\mathbf{p}$  is the forcing parameters vector,  $\dot{q}$  is the volumetric flow rate of the CSTR,  $T$  is the forcing frequency (component of  $\mathbf{p}$ ),  $N_{sites}$  is the number of empty catalytic sites and  $C_X$  is the concentration of X, a component of the state vector.



With the results of Ardagh *et al.* (Ardagh et al., 2020), as a starting point we explore how the turnover frequency of A ( $TOF_A$ ) changes w.r.t. two forcing parameters. Although the duty cycle was not considered in (Ardagh et al., 2020), here we show that tuning it can further improve  $TOF_A$ . To show this, we select the best performing  $U_L$  from Fig. 3 in Ardagh et al., 2020 (i.e.  $U_L = -1.46eV$ ), choose the same fixed amplitude  $\Delta U = 0.6eV$  and vary the frequency  $f$  and duty cycle  $\phi$ . To approximate the  $TOF_A$  surface, a “scaffolding” is made by evaluating it at a regular grid in the  $(\log_{10}f - \phi)$  parameter space.



**Figure 4.14:** Two parameter solution surface approximated by a grid of points where periodic steady states are found using Newton’s method (here, reporting the turnover frequency of A for each such state). Red line on left panel corresponds to the zoomed region on the right. All calculations were performed for fixed  $U_L = -1.46eV$ ,  $\Delta U = 0.6eV$ .

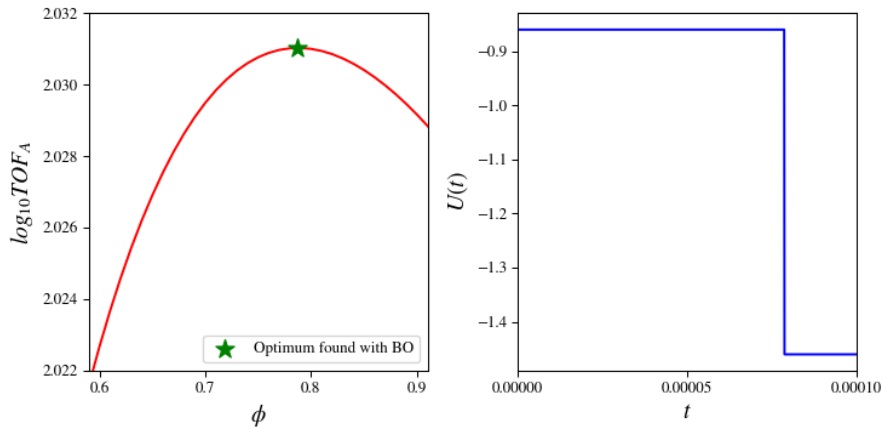
As it can be seen in Fig.4.14 the CSTR is performing better at higher forcing frequencies, something that is confirmed in Fig. 3 of (Ardagh et al., 2020). We expect that this is associated to the characteristic times of the two steady states, as explained in Subchapter 4.4. However, there is an optimum with respect to the duty cycle, which we hypothesize is attributed to a trade-off similar to the one presented in Fig.4.12. As a result, we achieve a maximum turnover frequency of  $107.4s^{-1}$  which can be compared to the one found in Ardagh et al., 2020, i.e.  $101s^{-1}$ .

## 4.6 Step forcing function optimization

After understanding Dynamic Catalysis, finding solutions efficiently and continuing them, the next natural step is to focus on optimization. Specifically, we will deal with maximizing the turnover frequency of A ( $TOF_A$ ). The optimization problem can be formulated as:

$$\mathbf{p}^* = \underset{\mathbf{p}}{\operatorname{argmax}}\{TOF_A(\mathbf{p})\} \quad (4.20)$$

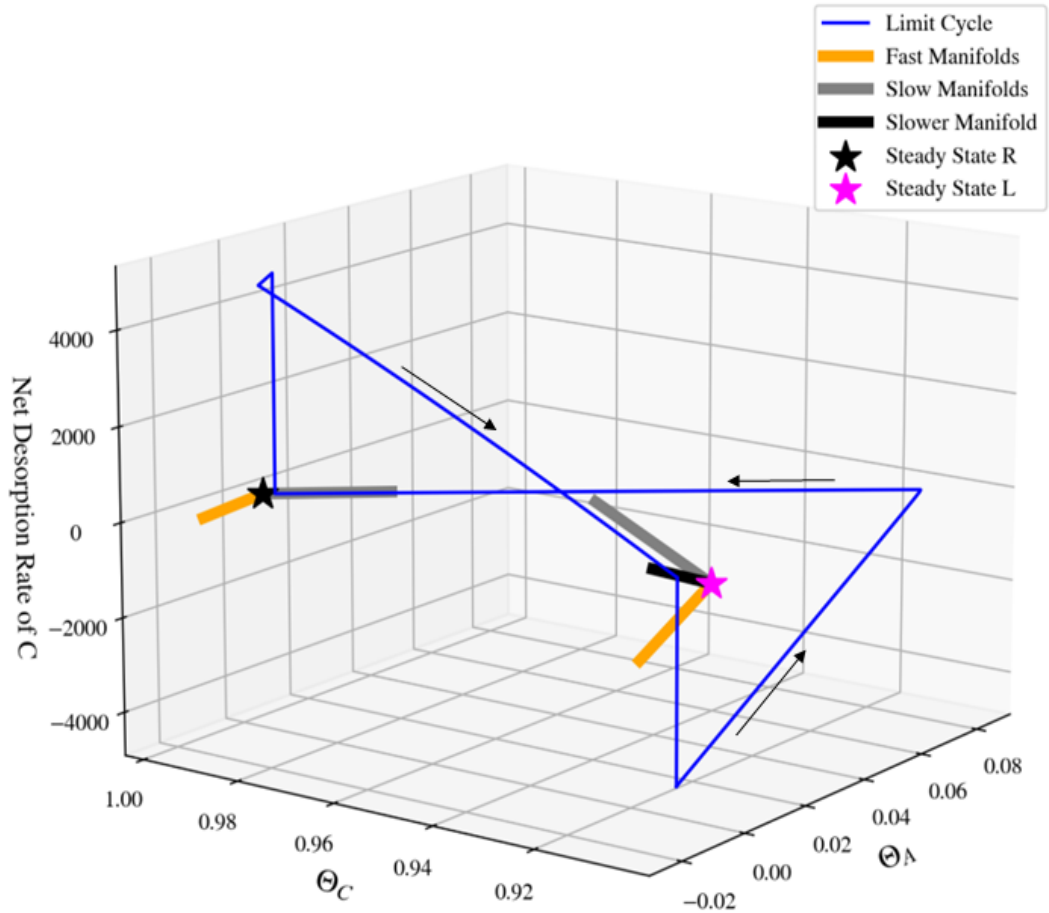
We will first seek to validate our approach against the results in Fig. 4.14 using the Bayesian Optimization framework discussed in Subchapters 2.2.3.1 and 2.2.3.2. For that purpose, one-parameter continuation is performed w.r.t  $\phi$  for  $\log_{10}f = 4$  (the solution branch corresponding to the right panel of Fig.4.14).



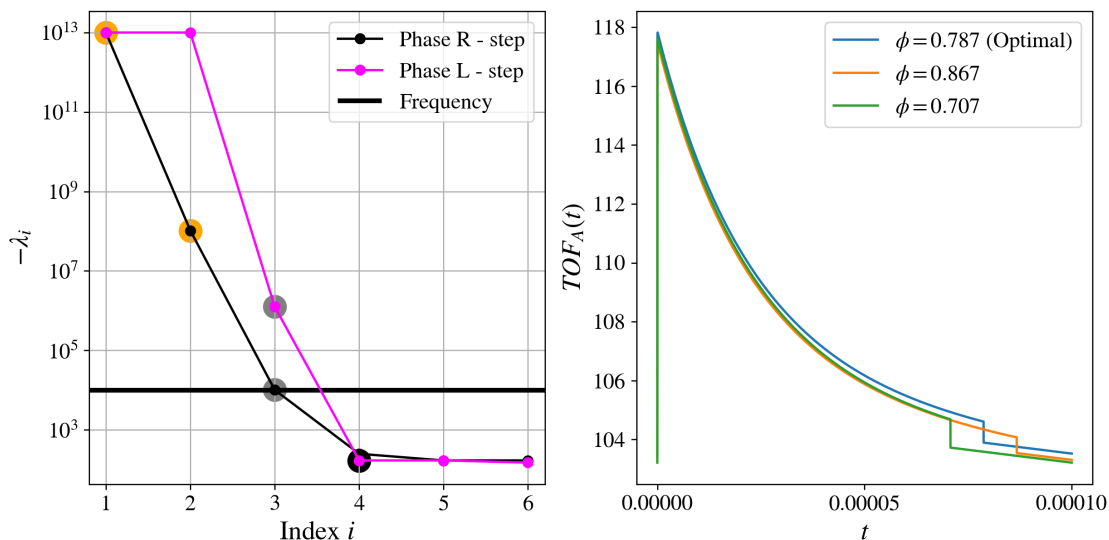
**Figure 4.15:** (left) Optimization result on top of the one-parameter continuation branch (for comparison, see right panel, Fig. 4.14). BO optimum location in agreement with what was expected from continuation, (right) optimal forcing function shape.

As Fig. 4.15 suggests, we are able to validate the BO approach: the optimum is located at the optimum suggested by the continuation results (specifically,  $\mathbf{p} = (\log_{10}f^*, \phi^*) = (4, 0.787)$ ). After finding the optimal forcing parameters, we will utilize insights from Subchapter 4.4 to *understand why* it is optimal.

As it can be seen in Fig.4.16 the eigendirections of the static steady states provide useful information about the directions of the limit cycle. As expected from the analysis in Subchapter 4.4 the limit cycle spends a lot of time on the slow manifolds at each phase. However, in this case, there are multiple manifolds, corresponding to multiple timescales. Furthermore, the manifolds are nonlinear and the



**Figure 4.16:** The limit cycle corresponding to the optimal forcing function shape found with Bayesian Optimization (Fig.4.15). Also plotted, are the two static steady states and all relevant eigendirections/ stable manifolds. The black arrows on the limit cycle denote the direction of time.



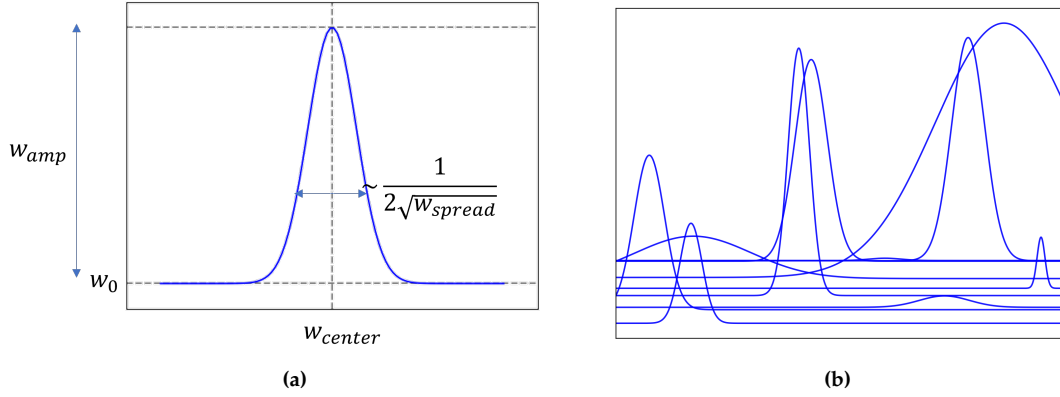
**Figure 4.17:** (left) Eigenvalues for the two static steady states, on top forcing frequency. The eigenvalues are colored according to their corresponding relevant eigendirections shown in Fig.4.16 (if any). (right) The turnover frequency over one period for the optimum found with BO and for less or greater  $\phi$  values.

limit cycle high-dimensional, rendering visualization difficult; we are restricted in the qualitative study of projections. Interestingly, the interplay of many slow manifolds on the limit cycle can be observed, such the one here for phase  $L$ . The respective eigenvalues (Fig.4.17, left panel) can inform us about the relation between the characteristic times of each eigendirection and the forcing frequency. As expected, the important slow manifolds have comparable characteristic times to the imposed forcing (see left panel Fig.4.17).

A closer examination of the timeseries in the right panel of Fig.4.17 allows us to understand the existence of an optimum w.r.t. the duty cycle  $\phi$ . It is evident that the optimum arises due to a trade-off between residence times in different phases: staying longer or shorter in the reaction promoting phase  $R$  (which also promotes  $A$  adsorption) results in lower final  $TOF_A$  and because this is a shooting problem, the overall rate will be less eventually.

## 4.7 Radial Basis Function optimization

What has been done so far, is optimizing the *shape* of the forcing function, in an (at most) 4-dimensional family of step functions. It is possible to add degrees of freedom, (and necessarily increase the overall complexity), if we consider a versatile set of basis functions that can parametrize shapes, for



**Figure 4.18:** (a) A single Radial Basis Function (RBF) parametrized by four parameters:  $w_0, w_{amp}, w_{center}, w_{spread}$ , (b) Random sample of RBFs.

example, Radial Basis Functions (RBFs) (Park and Sandberg, 1991).

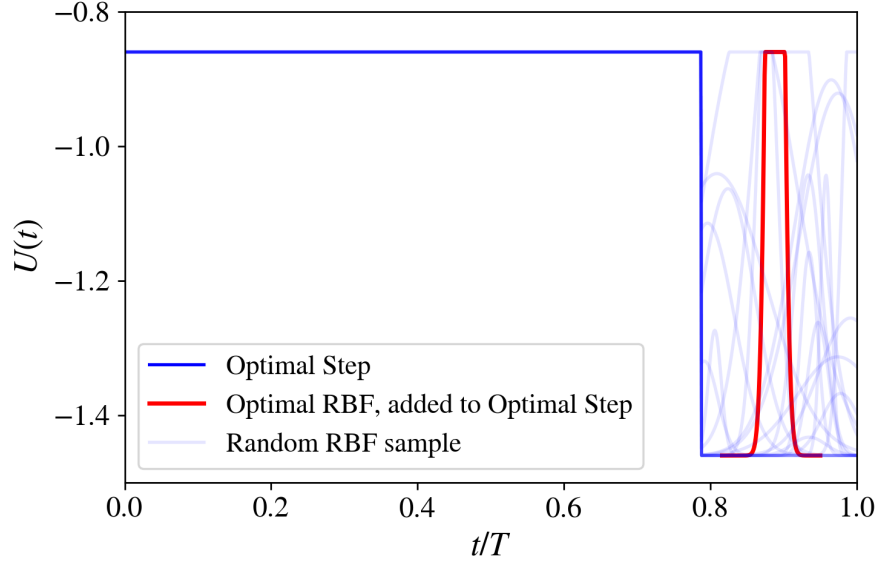
In the context of Dynamic Catalysis, the relative binding energy of A can be expressed with RBFs, as:

$$\Delta BE_A(t) = w_0 + \sum_{i=1}^N w_{amp_i} e^{w_{spread_i}(t-w_{center_i})^2} \quad (4.21)$$

This would result in  $3N + 1$  - dimensional family of shapes for  $N$  RBFs. The meaning of the parameters, can be intuitively understood from Fig.4.18. Denoting  $\mathbf{w} = [w_0, w_{amp_1}, \dots, w_{amp_N}, w_{center_1}, \dots, w_{center_N}, w_{spread_1}, \dots, w_{spread_N}]$  the optimization problem now becomes:

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmax}} \{TOF_A(\mathbf{w})\} \quad (4.22)$$

To establish the benefits of expressing forcing shapes with RBFs, we will attempt to further improve the optimum step forcing function shape found with BO (Fig.4.15) by **adding one radial basis function on the  $L$  phase and optimizing its shape.**



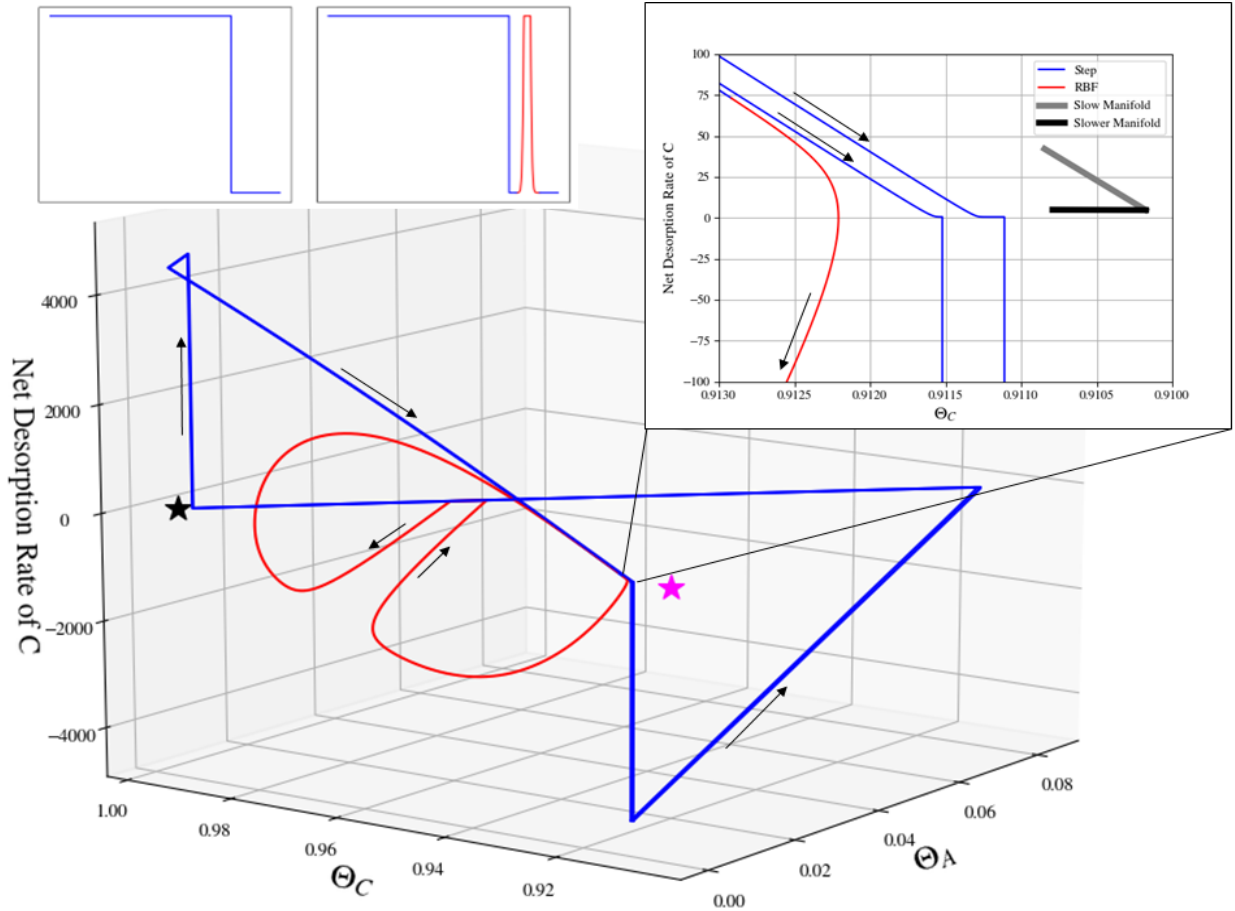
**Figure 4.19:** Out of all RBFs that can be added to the  $L$  phase of the already optimal step forcing function, BO locates the one that maximizes  $TOf_A$ .

Fig.4.19 shows a sample of RBFs that can be added on the  $L$  phase of the oscillation. This can be imposed by restricting  $w_{center}$  in  $[\phi^*, 1]$ , where  $\phi^* = 0.788$  as found in Fig.4.15. Note that the rest of the step function shape parameters are kept fixed:  $f = 10^4\text{Hz}$ ,  $\Delta U = 0.6\text{eV}$ ,  $U_L = -1.46\text{eV}$ . To make sure that these values are respected, we fix  $w_0 = 0$  and truncate any RBF at  $U_R = U_L + \Delta U$ . As for the amplitude of the RBF  $w_{amp}$ , we arbitrarily bound it in  $[0, 2]$  and the spread  $w_{spread}$  at  $[10, 10^4]$ . These choices are arbitrary, yet reasonable: too high values of  $w_{amp}$  seem to have little effect on the shape, because of the truncation, and too high values of  $w_{spread}$  create too large potential gradients.

The optimum found (shown in the right panel of Fig.4.19) is  $TOF_A(\mathbf{w}^*) = 112.5\text{s}^{-1}$ ,  $\mathbf{w}^* = [0, 2, 0.866, 3.76]$ . In this case, the addition of the RBF has led to further improvement of the performance of the catalytic reactor. We argue that this improvement is not the result of additional *residence* in one of the two phases, but the result of an optimal *shape*. As evidence, we consider the case of an equivalent forcing function, for which all forcing parameters are the same as in the case of the optimal step forcing function  $[f, U_L, \Delta U] = [10^4\text{Hz}, -1.46\text{eV}, 0.6\text{eV}]$  but now the duty cycle is  $\phi = 0.787 + 0.022$ , where 0.022 is the *additional* residence time of the RBF in state  $U_R$  (See plateau on top in right panel of Fig.4.19). We find that this configuration performs worse ( $TOF_A = 107.39\text{s}^{-1}$ ) than in the case of separate step and RBF.

Our hypothesis is that optimal shapes are the ones best fitted to guide the dynamics of the system to most productive states. To provide more evidence, we plot a projection of the limit cycles of the optimal

step and the optimal step and RBF, and turn to the eigendirection analysis for insights.

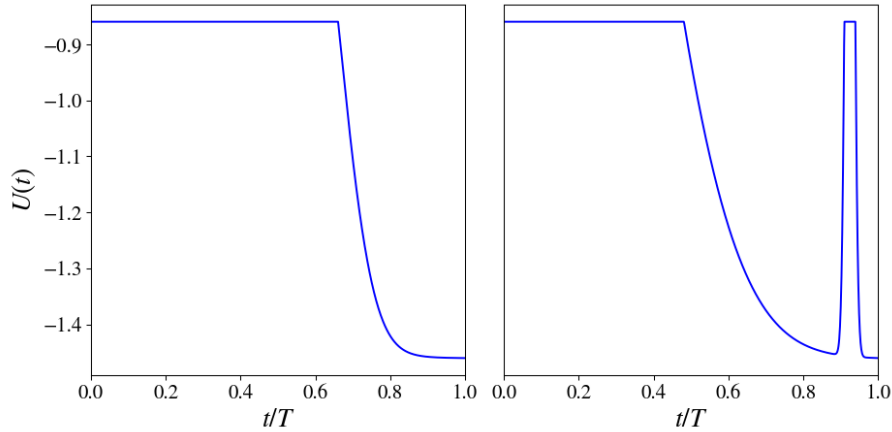


**Figure 4.20:** Limit cycle projections for the two optimization solutions found: one for the optimal step forcing function (blue) and one for the optimal additional RBF (red) with the optimal step forcing function. The left inset shows these two optimal forcing functions. The right inset shows a zoomed region where the trajectories of the two limit cycles diverge, along with the relevant eigendirections. Black arrows denote the time direction. Note that the steady states (and respective eigenvalues/eigenvectors) will be slightly different for the two cases shown here, but we include only one for simplicity.

As expected and confirmed by Fig.4.20 these two cases have similar dynamics for most of the time, i.e. the respective limit cycles move mostly along the same slow eigendirections. In the case of the additional RBF (colored red), the limit cycle suddenly diverges and follows a different trajectory leading back to the slow eigendirection of phase  $R$  (coinciding with the plateau at the right panel of Fig.4.19). After a while, the trajectory returns close to the point of divergence to join the trajectory of the step forcing function (colored blue). While the former trajectory was kicked off its course, the latter one ended up at an extremely slower and less “productive” eigendirection. This can be better seen in the

inset of Fig.4.20. In other words, the optimal additional RBF prevented the trajectory from spending useful time in an unproductive state. This observation further corroborates the case that it is the *shape* of the forcing function that matters, as it can be selected to optimally assist the system dynamics to stay in more productive states.

The next step is to completely rely on RBFs for the construction of forcing function shapes. To demonstrate this, we formulate a new optimization problem similar to Eq.4.22 with the superposition of two radial basis functions. We choose to keep  $w_0$  fixed at  $-1.46\text{eV}$ , and also  $f = 10^4\text{Hz}$ . The chosen value of the fixed frequency is based on our knowledge of the characteristic times. We also choose to keep the rest of the constraints used in the optimization of one RBF, with the exception of the  $w_{center}$  parameter: we contain the first RBF on the first 60% of the period and the second at the last 60%. This constraint allows for *some* overlap of the RBFs but avoids problematic RBF permutations. The resulting optimization problem is over a 6-dimensional family of shapes.



**Figure 4.21:** Two of the converged solutions of optimization with two RBFs. The converged solutions approximate (left) the step forcing function (right) the step forcing function with the additional RBF, both previously examined.

As it can be seen from Fig.4.21, BO converges to “familiar” shapes: the left panel of Fig.4.21 resembles the optimal step forcing function shape in Fig.4.15, while the right panel of Fig.4.21 resembles the step and added RBF shape in Fig.4.19. This similarity is reflected on similar values of  $TOF_A$ . The above observation demonstrates how RBFs can be used as an universal approximator for any shape, and thus, can be superimposed to construct forcing functions of any complexity.



## 4.8 Discussion and Conclusions

Dynamic Catalysis has the potential to significantly improve the performance of catalytic reactors and facilitate the transition to efficient and sustainable production of valuable chemicals. In this work, various case studies from recent literature were sourced to get deeper insights into how Dynamic Catalysis works and what really matters when tuning the extra degrees of freedom. Combining established numerical methods with modern Machine Learning schemes, we demonstrated efficient location of solutions, continuation of such solutions and performance optimization.

An interesting future direction is to explore efficient optimization of more complex forcing function shapes (e.g. with more radial basis functions, or functions of another basis). As the dimensionality of the optimization problem increases, we hypothesize that **reduced** Bayesian optimization can significantly accelerate convergence (Pozharskiy et al., 2020). We believe there is significant benefit in employing Global optimization approaches for Gaussian Processes used in Bayesian Continuation and Optimization (Schweidtmann et al., 2021).

It is also useful to study different objective functions, such as reactor selectivities (defined as  $S_X = \frac{TOF_X}{\sum_i TOF_i}$ ), or multiple objective functions (e.g. as Pareto fronts - multiobjective optimization (Miettinen, 1998)). Apart from that, Bayesian Continuation is a “side product” of this work which, we believe, has many promising future directions/extensions.

# References

- Buckingham, E. (1914). "On Physically Similar Systems; Illustrations of the Use of Dimensional Equations". In: *Phys. Rev.* 4 (4), pp. 345–376. DOI: 10.1103/PhysRev.4.345. URL: <https://link.aps.org/doi/10.1103/PhysRev.4.345>.
- Aris, Rutherford and R. H. S. Mah (1963). "Independence of Chemical Reactions". In: *Industrial & Engineering Chemistry Fundamentals* 2.2, pp. 90–94. DOI: 10.1021/i160006a002. eprint: <https://doi.org/10.1021/i160006a002>. URL: <https://doi.org/10.1021/i160006a002>.
- Horn, F. J. M. and R. C. Lin (1967). "Periodic Processes: A Variational Approach". In: *Industrial & Engineering Chemistry Process Design and Development* 6.1, pp. 21–30. ISSN: 0196-4305. DOI: 10.1021/i260021a005. URL: <https://doi.org/10.1021/i260021a005>.
- Bailey, J. E. (1974). "Periodic Operation Of Chemical Reactors: A Review". In: *Chemical Engineering Communications* 1.3, pp. 111–124. DOI: 10.1080/00986447408960421. eprint: <https://doi.org/10.1080/00986447408960421>. URL: <https://doi.org/10.1080/00986447408960421>.
- Tomita, Kazuhisa and Tohru Kai (1979). "Chaotic response of a limit cycle". In: *Journal of Statistical Physics* 21.1, pp. 65–86. ISSN: 1572-9613. DOI: 10.1007/BF01011482. URL: <https://doi.org/10.1007/BF01011482>.
- Aronson, D. G., R. P. McGehee, I. G. Kevrekidis, and R. Aris (1986). "Entrainment regions for periodically forced oscillators". In: *Phys. Rev. A* 33 (3), pp. 2190–2192. DOI: 10.1103/PhysRevA.33.2190. URL: <https://link.aps.org/doi/10.1103/PhysRevA.33.2190>.
- Kevrekidis, I.G., R. Aris, and L.D. Schmidt (1986a). "Forcing an entire bifurcation diagram: Case studies in chemical oscillators". In: *Physica D: Nonlinear Phenomena* 23.1, pp. 391–395. ISSN: 0167-2789. DOI: [https://doi.org/10.1016/0167-2789\(86\)90145-4](https://doi.org/10.1016/0167-2789(86)90145-4). URL: <https://www.sciencedirect.com/science/article/pii/0167278986901454>.
- Kevrekidis, I.G., R. Aris, and L.D. Schmidt (1986b). "The stirred tank forced". In: *Chemical Engineering Science* 41.6, pp. 1549–1560. ISSN: 0009-2509. DOI: [https://doi.org/10.1016/0009-2509\(86\)85237-X](https://doi.org/10.1016/0009-2509(86)85237-X). URL: <https://www.sciencedirect.com/science/article/pii/000925098685237X>.
- Ichikawa, Shinichi (1990). "Volcano-shaped curves in heterogeneous catalysis". In: *Chemical Engineering Science* 45.2, pp. 529–535. ISSN: 0009-2509. DOI: [https://doi.org/10.1016/0009-2509\(90\)87039-U](https://doi.org/10.1016/0009-2509(90)87039-U). URL: <https://www.sciencedirect.com/science/article/pii/000925099087039U>.
- Park, J. and I. W. Sandberg (1991). "Universal Approximation Using Radial-Basis-Function Networks". In: *Neural Computation* 3.2, pp. 246–257. DOI: 10.1162/neco.1991.3.2.246.
- Miettinen, K. M. (1998). *Nonlinear Multiobjective Optimization*. Dordrecht: Kluwer Academic Publishers. ISBN: 0-7923-8278-1. URL: <https://pure.iiasa.ac.at/id/eprint/5443/>.
- Doedel, Eusebius, A. Champneys, Thomas Fairgrieve, Yu Kuznetsov, Björn Sandstede, and Xianjun Wang (1999). "AUTO 97: Continuation And Bifurcation Software For Ordinary Differential Equations (with HomCont)". In.

- Kevrekidis, I.G., L.D. Schmidt, and R. Aris (1999). "I - \*Some Common Features of Periodically Forced Reacting Systems \*Reprinted with the permission of Elsevier Science from Chemical Engineering Science, volume 41, number 5, pp. 1263–1276, 1986. Copyright 1986 Elsevier Science." In: *Mathematical Modeling*. Ed. by Rutherford Aris. Vol. 1. Process Systems Engineering. Academic Press, pp. 227–251. DOI: [https://doi.org/10.1016/S1874-5970\(99\)80019-8](https://doi.org/10.1016/S1874-5970(99)80019-8). URL: <https://www.sciencedirect.com/science/article/pii/S1874597099800198>.
- Chianelli, R.R, G Berhault, P Raybaud, S Kasztelan, J Hafner, and H Toulhoat (2002). "Periodic trends in hydrodesulfurization: in support of the Sabatier principle". In: *Applied Catalysis A: General* 227.1, pp. 83–96. ISSN: 0926-860X. DOI: [https://doi.org/10.1016/S0926-860X\(01\)00924-3](https://doi.org/10.1016/S0926-860X(01)00924-3). URL: <https://www.sciencedirect.com/science/article/pii/S0926860X01009243>.
- Kelley, Carl (2003). *Solving nonlinear equations with Newton's method - fundamentals of algorithms*. ISBN: 978-0-89871-546-0.
- Wolff, J., A. G. Papathanasiou, H. H. Rotermund, G. Ertl, X. Li, and I. G. Kevrekidis (2003a). "Gentle Dragging of Reaction Waves". In: *Phys. Rev. Lett.* 90 (1), p. 018302. DOI: [10.1103/PhysRevLett.90.018302](https://doi.org/10.1103/PhysRevLett.90.018302). URL: <https://link.aps.org/doi/10.1103/PhysRevLett.90.018302>.
- Wolff, J., A. G. Papathanasiou, HH Rotermund, G. Ertl, X. Li, and I. G. Kevrekidis (2003b). "Local manipulation of catalytic surface reactivity". In: *Journal of Catalysis* 216.1-2, pp. 246–256.
- Doedel, Eusebius J., Thomas F. Fairgrieve, Björn Sandstede, Alan R. Champneys, Yuri A. Kuznetsov, and Xianjun Wang (2007). *AUTO-07P: Continuation and bifurcation software for ordinary differential equations*. Tech. rep.
- Qiao, L., X. Li, I. G. Kevrekidis, C. Punckt, and H. H. Rotermund (2008). "Enhancement of surface activity in CO oxidation on Pt(110) through spatiotemporal laser actuation". In: *Phys. Rev. E* 77 (3), p. 036214. DOI: [10.1103/PhysRevE.77.036214](https://doi.org/10.1103/PhysRevE.77.036214). URL: <https://link.aps.org/doi/10.1103/PhysRevE.77.036214>.
- Deka, Upakul, Ines Lezcano-Gonzalez, Bert M. Weckhuysen, and Andrew M. Beale (2013). "Local Environment and Nature of Cu Active Sites in Zeolite-Based Catalysts for the Selective Catalytic Reduction of NOx". In: *ACS Catalysis* 3.3, pp. 413–427. DOI: [10.1021/cs300794s](https://doi.org/10.1021/cs300794s). eprint: <https://doi.org/10.1021/cs300794s>. URL: <https://doi.org/10.1021/cs300794s>.
- Rodriguez, José A., Ping Liu, Dario J. Stacchiola, Sanjaya D. Senanayake, Michael G. White, and Jingguang G. Chen (2015). "Hydrogenation of CO2 to Methanol: Importance of Metal-Oxide and Metal-Carbide Interfaces in the Activation of CO2". In: *ACS Catalysis* 5.11, pp. 6696–6706. DOI: [10.1021/acscatal.5b01755](https://doi.org/10.1021/acscatal.5b01755). URL: <https://doi.org/10.1021/acscatal.5b01755>.
- Friend, Cynthia M. and Bingjun Xu (2017). "Heterogeneous Catalysis: A Central Science for a Sustainable Future". In: *Accounts of Chemical Research* 50.3, pp. 517–521. ISSN: 0001-4842. DOI: [10.1021/acs.accounts.6b00510](https://doi.org/10.1021/acs.accounts.6b00510). URL: <https://doi.org/10.1021/acs.accounts.6b00510>.
- Akhade, Sneha A., Robert M. Nidzyn, Gholamreza Rostamikia, and Michael J. Janik (2018). "Using Brønsted-Evans-Polanyi relations to predict electrode potential-dependent activation energies". In: *Catalysis Today* 312, pp. 82–91. ISSN: 0920-5861. DOI: <https://doi.org/10.1016/j.cattod.2018.03.048>. URL: <https://www.sciencedirect.com/science/article/pii/S0920586118303122>.
- Dunwell, Marco, Wesley Luc, Yushan Yan, Feng Jiao, and Bingjun Xu (2018). "Understanding Surface-Mediated Electrochemical Reactions: CO2 Reduction and Beyond". In: *ACS*

- Catalysis* 8.9, pp. 8121–8129. DOI: 10.1021/acscatal.8b02181. URL: <https://doi.org/10.1021/acscatal.8b02181>.
- Foster, Shelby L., Sergio I. Perez Bakovic, Royce D. Duda, Sharad Maheshwari, Ross D. Milton, Shelley D. Minter, Michael J. Janik, Julie N. Renner, and Lauren F. Greenlee (2018). “Catalysts for nitrogen reduction to ammonia”. In: *Nature Catalysis* 1.7, pp. 490–500. ISSN: 2520-1158. DOI: 10.1038/s41929-018-0092-7. URL: <https://doi.org/10.1038/s41929-018-0092-7>.
- Singh, Aayush R., Joseph H. Montoya, Brian A. Rohr, Charlie Tsai, Aleksandra Vojvodic, and Jens K. Nørskov (2018). “Computational Design of Active Site Structures with Improved Transition-State Scaling for Ammonia Synthesis”. In: *ACS Catalysis* 8.5, pp. 4017–4024. DOI: 10.1021/acscatal.8b00106. URL: <https://doi.org/10.1021/acscatal.8b00106>.
- Ardagh, M. Alexander, Omar A. Abdelrahman, and Paul J. Dauenhauer (2019). “Principles of Dynamic Heterogeneous Catalysis: Surface Resonance and Turnover Frequency Response”. In: *ACS Catalysis* 9.8, pp. 6929–6937. DOI: 10.1021/acscatal.9b01606. URL: <https://doi.org/10.1021/acscatal.9b01606>.
- Ardagh, M. Alexander, Turan Birol, Qi Zhang, Omar A. Abdelrahman, and Paul J. Dauenhauer (2019). “Catalytic resonance theory: superVolcanoes, catalytic molecular pumps, and oscillatory steady state”. In: *Catal. Sci. Technol.* 9 (18), pp. 5058–5076. DOI: 10.1039/C9CY01543D. URL: <http://dx.doi.org/10.1039/C9CY01543D>.
- Stadler, Bernhard M., Christoph Wulf, Thomas Werner, Sergey Tin, and Johannes G. de Vries (2019). “Catalytic Approaches to Monomers for Polymers Based on Renewables”. In: *ACS Catalysis* 9.9, pp. 8012–8067. DOI: 10.1021/acscatal.9b01665. URL: <https://doi.org/10.1021/acscatal.9b01665>.
- Vogt, Charlotte, Matteo Monai, Gert Jan Kramer, and Bert M. Weckhuysen (2019). “The renaissance of the Sabatier reaction and its applications on Earth and in space”. In: *Nature Catalysis* 2.3, pp. 188–197. ISSN: 2520-1158. DOI: 10.1038/s41929-019-0244-4. URL: <https://doi.org/10.1038/s41929-019-0244-4>.
- Ardagh, M. Alexander, Manish Shetty, Anatoliy Kuznetsov, Qi Zhang, Phillip Christopher, Dionisios G. Vlachos, Omar A. Abdelrahman, and Paul J. Dauenhauer (2020). “Catalytic resonance theory: parallel reaction pathway control”. In: *Chem. Sci.* 11 (13), pp. 3501–3510. DOI: 10.1039/C9SC06140A. URL: <http://dx.doi.org/10.1039/C9SC06140A>.
- Gopeesingh, Joshua, M. Alexander Ardagh, Manish Shetty, Sean T. Burke, Paul J. Dauenhauer, and Omar A. Abdelrahman (2020). “Resonance-Promoted Formic Acid Oxidation via Dynamic Electrocatalytic Modulation”. In: *ACS Catalysis* 10.17, pp. 9932–9942. DOI: 10.1021/acscatal.0c02201. URL: <https://doi.org/10.1021/acscatal.0c02201>.
- Pozharskiy, Dmitry, Noah J. Wichrowski, Andrew B. Duncan, Grigorios A. Pavliotis, and Ioannis G. Kevrekidis (2020). *Manifold Learning for Accelerating Coarse-Grained Optimization*. DOI: 10.48550/ARXIV.2001.03518. URL: <https://arxiv.org/abs/2001.03518>.
- Qi, Ji, Joaquin Resasco, Hossein Robatjazi, Isabel Barraza Alvarez, Omar Abdelrahman, Paul Dauenhauer, and Phillip Christopher (2020). “Dynamic Control of Elementary Step Energetics via Pulsed Illumination Enhances Photocatalysis on Metal Nanoparticles”. In: *ACS Energy Letters* 5.11, pp. 3518–3525. DOI: 10.1021/acsenenergylett.0c01978. URL: <https://doi.org/10.1021/acsenenergylett.0c01978>.
- Shetty, Manish, M. Alexander Ardagh, Yutong Pang, Omar A. Abdelrahman, and Paul J. Dauenhauer (2020). “Electric-Field-Assisted Modulation of Surface Thermochemistry”. In: *ACS Catalysis* 10.21, pp. 12867–12880. DOI: 10.1021/acscatal.0c02124. URL: <https://doi.org/10.1021/acscatal.0c02124>.

- Lim, Chia Wei, Max J. Hülsey, and Ning Yan (2021). “Non-Faradaic Promotion of Ethylene Hydrogenation under Oscillating Potentials”. In: *JACS Au* 1.5, pp. 536–542. DOI: [10.1021/jacsau.1c00044](https://doi.org/10.1021/jacsau.1c00044). URL: <https://doi.org/10.1021/jacsau.1c00044>.
- Lovelett, Robert J., Evan M. Zhao, Makoto A. Lalwani, Jared E. Toettcher, Ioannis G. Kevrekidis, and José L. Avalos (2021). “Dynamical Modeling of Optogenetic Circuits in Yeast for Metabolic Engineering Applications”. In: *ACS Synthetic Biology* 10.2, pp. 219–227. DOI: [10.1021/acssynbio.0c00372](https://doi.org/10.1021/acssynbio.0c00372). URL: <https://doi.org/10.1021/acssynbio.0c00372>.
- Schweidtmann, Artur M., Dominik Bongartz, Daniel Grothe, Tim Kerkenhoff, Xiaopeng Lin, Jaromil Najman, and Alexander Mitsos (2021). “Deterministic global optimization with Gaussian processes embedded”. In: *Mathematical Programming Computation* 13.3, pp. 553–581. ISSN: 1867-2957. DOI: [10.1007/s12532-021-00204-y](https://doi.org/10.1007/s12532-021-00204-y). URL: <https://doi.org/10.1007/s12532-021-00204-y>.
- Sordello, F., F. Pellegrino, M. Prozzi, C. Minero, and V. Maurino (2021). “Controlled Periodic Illumination Enhances Hydrogen Production by over 50% on Pt/TiO<sub>2</sub>”. In: *ACS Catalysis* 11.11, pp. 6484–6488. DOI: [10.1021/acscatal.1c01734](https://doi.org/10.1021/acscatal.1c01734). eprint: <https://doi.org/10.1021/acscatal.1c01734>. URL: <https://doi.org/10.1021/acscatal.1c01734>.
- Fu, Yukui, Zhuo Yin, Lei Qin, Danlian Huang, Huan Yi, Xigui Liu, Shiyu Liu, Mingming Zhang, Bisheng Li, Ling Li, Wenjun Wang, Xuerong Zhou, Yixia Li, Guangming Zeng, and Cui Lai (2022). “Recent progress of noble metals with tailored features in catalytic oxidation for organic pollutants degradation”. In: *Journal of Hazardous Materials* 422, p. 126950. ISSN: 0304-3894. DOI: <https://doi.org/10.1016/j.jhazmat.2021.126950>. URL: <https://www.sciencedirect.com/science/article/pii/S030438942101918X>.
- Gathmann, Sallye R., M. Alexander Ardagh, and Paul J. Dauenhauer (2022). “Catalytic resonance theory: Negative dynamic surfaces for programmable catalysts”. In: *Chem Catalysis* 2.1, pp. 140–163. ISSN: 2667-1093. DOI: <https://doi.org/10.1016/j.checat.2021.12.006>. URL: <https://www.sciencedirect.com/science/article/pii/S2667109321003766>.
- Wittreich, Gerhard R., Shizhong Liu, Paul J. Dauenhauer, and Dionisios G. Vlachos (2022). “Catalytic resonance of ammonia synthesis by simulated dynamic ruthenium crystal strain”. In: *Science Advances* 8.4, eabl6576. DOI: [10.1126/sciadv.abl6576](https://doi.org/10.1126/sciadv.abl6576). eprint: <https://www.science.org/doi/pdf/10.1126/sciadv.abl6576>. URL: <https://www.science.org/doi/abs/10.1126/sciadv.abl6576>.

## Chapter 5

# Learning Chemotactic PDEs with Machine Learning

Understanding and predicting the behavior of a complex biological system (such as an ensemble of cells) is always a challenge. In the case of multi-agent systems, when the single-agent dynamics are known (possibly from first principles), the system can be studied and simulated at the microscale. Macroscale behavior naturally arises from simulating sufficiently large agent ensembles. It is sometimes possible to derive macroscale partial differential equations from the dynamics of the individual agents (Erban and Othmer, 2004). Such PDE-level descriptions are particularly attractive, as we are usually interested in the evolution of only a few, important macroscopic variables, rather than of the behavior of each individual “microscopic” agent. Importantly, one also needs to know which (and how many) macroscopic variables/observables are sufficient to usefully construct a closed macroscopic evolution equation (e.g. (Lee et al., 2020)).

For systems of great complexity, an accurate macroscopic PDE may be out of reach. One could only gather (full or partial) information from experiments and/or fine-scale, individual based/stochastic simulations. This calls for a data-driven approach to “discover” a macroscopic law for a “coarse” PDE, solely from spatiotemporal data (experimental or computational movies). Such a data-driven PDE can then be exploited towards the following purposes:

1. Predicting the time evolution starting from different initial conditions or in out-of-sample spatiotemporal domains. This is particularly attractive when it is not easy to probe the system and extract such profiles “on demand” from experiments or simulations.
2. Reconstructing the full behavior of the system even when only partial information is at hand (i.e. when we do not have data for all important macroscopic variables).
3. If a qualitatively correct but quantitatively inaccurate macroscopic model is available, a quantitative data-driven model can help probe and even understand different components of the system’s behavior (Lee et al., 2022). This can be a way to shed light on the fundamental physical laws of the studied system (explainability).

In this Chapter, data-driven PDEs are constructed for **the chemotactic motility of bacteria**, i.e. their ability to direct multicellular motion along chemical gradients. This phenomenon is central to environmental, medical and agricultural processes (Bhattacharjee et al., 2021). Since the pioneering work of Adler (Adler, 1969), chemotaxis has been extensively studied, decoding complex mechanisms ranging from biochemistry and molecular genetics (Parkinson, 1976; Parkinson, 1980) to the inter- (Boyd, Krikos, and Simon, 1981; Sourjik and Berg, 2002) and intra-cellular signaling (Liu and Parkinson, 1989; Heit et al., 2002) and from the sensory adaptation in response to external stimuli (Segel et al., 1986; Othmer and Schaap, 1998) to the motor structure and the flagellum-related motility (Cluzel, Surette, and Leibler, 2000) and scaling up to the emergent collective behaviour (Wu et al., 2006). Depending on the level of information and spatio-temporal scale of analysis, a vast number of mathematical models have been proposed ranging from the molecular/individual (Emonet et al., 2005; Coburn et al., 2013; Othmer, Xin, and Xue, 2013; Rousset and Samaey, 2013; Yasuda, 2017) to the continuum/macroscopic scale (Patlak, 1953; Keller and Segel, 1971; Erban and Othmer, 2004; Bellomo et al., 2010; Othmer, Xin, and Xue, 2013; Franz and Erban, 2013; Bellomo et al., 2022) (for an extensive review of both modelling approaches see (Tindall et al., 2008b; Tindall et al., 2008a; Othmer, Xin, and Xue, 2013; Bellomo et al., 2022). The celebrated Keller-Segel (Keller and Segel, 1971) PDE derived for the macroscopic description of the population density evolution, coupled with the concomitant chemoattractant field, constitutes the cornerstone in the field. In its simplest form, the dependence of the cell density  $b(x, t)$  in space and time evolves according to

$$\frac{\partial b}{\partial t} = \nabla \cdot (D \nabla b - \chi(s) b \nabla s), \quad (5.1)$$

coupled with appropriate boundary conditions; In Eq. 5.1,  $D$  is the diffusion coefficient,  $\chi$  is the *chemotactic coefficient* and  $s$  is the substrate distribution. This model explicitly describes cell motility

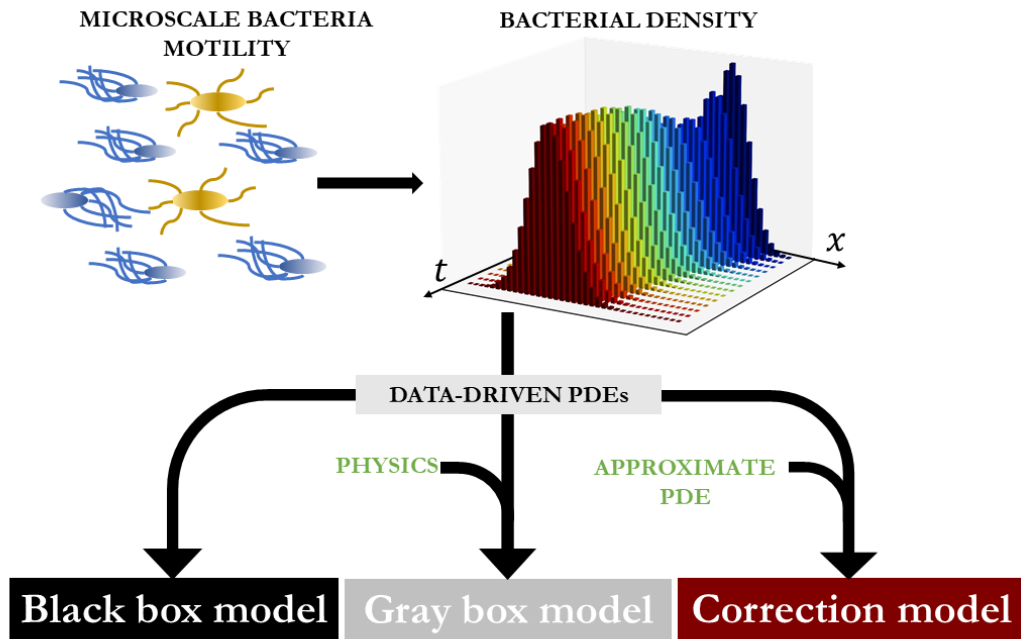
through two terms: a **diffusion term** (usually isotropic) and a **chemotactic term**, which encapsulates the response of the bacteria in the presence of a substrate, i.e. chemoattractant/chemorepellent field. This response includes signal transduction dynamics and properties of cellular chemoreceptors. In this term, the function  $\chi : \mathbb{R} \rightarrow \mathbb{R}$  can be tuned for different kinds of chemotactically relevant substances and their spatial profile. Most importantly, the sign of  $\chi$  distinguishes chemoattractants vs. chemorepellents. In this work, we will deal only with chemoattractants (and specifically chemonutrients).

In fact, the effect of the substrate's distribution  $s(x, t)$  and the bacterial density  $b(x, t)$  on both  $D$  and  $\chi$  are not generally known. In order to obtain an expression in closed-form and then attempt to fuse experimental observations and bio-physical insight, several assumptions are made, resulting to different closed-form approximations. For example, in the original paper of Keller and Segel, (Keller and Segel, 1971) it is assumed that  $D = D(s)$  and  $\chi = \chi(s)$ , i.e., that both depend on the distribution of the substrate  $s$ . Assuming  $D = D(b)$ ,  $\chi = \chi(b)$ , the Keller-Segel PDE reduces to a Fokker-Planck equation, while for constant diffusion  $D$  and constant chemotactic  $\chi$  coefficient, we obtain the Smoluchowski equation (for a review of different closures and models refer to (Othmer and Schaap, 1998; Chavanis, 2008; Erban and Othmer, 2004; Othmer, Xin, and Xue, 2013; Painter, 2019)).

In this work, we demonstrate a toolbox of Data Mining methodologies for learning different forms of *the law* of macroscopic chemotactic partial differential equations, from three types of data sets (in order of decreasing available information and assumptions):

- i Data derived from simulations of *known* chemotactic PDEs (Subchapter 5.1). In this case, it is possible to explore different learning modalities and validate our approach.
- ii Data derived from Agent-Based Modeling (Monte Carlo simulations), where each bacterium is simulated as a stochastic agent (Subchapter 5.2). In this case, it is shown how data-driven PDEs can predict emergent behaviors from microscopic computational experiments. It is also shown, how data-driven PDEs can exploit (different levels of) information from approximate models.
- iii Fluorescence data from chemotaxis experiments (Subchapter 5.3). Having validated our approaches in the two previous cases, a data-driven model is learned from *real-world* chemotactic motility data, where practical considerations become important.





**Figure 5.1:** Overview of proposed algorithmic pipeline: from individual bacterium motility patterns to data-driven, possibly partially physics-informed, surrogate models for chemotactic Partial Differential Equations (PDEs).

This work falls in the general category of nonlinear system identification using data-driven, Machine Learning-assisted surrogate models. Neural Networks have repeatedly demonstrated successes in learning nonlinear Ordinary (Rico-Martinez, Anderson, and Kevrekidis, 1994) or Partial Differential Equations (Lee et al., 2020; González-García, Rico-Martínez, and Kevrekidis, 1998; Siettos, Bafas, and Boudouvis, 2002; Siettos and Bafas, 2002; Alexandridis et al., 2002). More recently, with the increased accessibility of powerful computational hardware and the computational efficiency of Machine Learning algorithms, nonlinear system identification has attracted a lot of attention (Galaris et al., 2022; Raissi, Perdikaris, and Karniadakis, 2019; Kemeth et al., 2020; Lee et al., 2020; Vlachas et al., 2020; Chen et al., 2021; Karniadakis et al., 2021) and has motivated the design of novel approaches and architectures. Notable among these approaches are Neural ODEs (Chen et al., 2019) and Convolutional Neural Networks (LeCun, Bengio, et al., n.d.; Rao et al., 2022), sparse identification (Brunton, Proctor, and Kutz, 2016) and effective dynamics identification (Vlachas et al., 2018; Vlachas et al., 2022).

Work in this Chapter was done in collaboration with Dr. Seungjoon Lee, Dr. Juan Bello-Rivas, Prof. Constantinos Siettos, Prof. Sujit Datta, Dr. Tapomoy Bhattacharjee and Prof. Ioannis Kevrekidis. It

includes submitted work in the manuscript “Learning black- and gray-box chemotactic PDEs/closures from agent based Monte Carlo simulation data” (Lee et al., 2022) and the manuscript (to be submitted) “Data-driven Discovery of Chemotactic Migration of Bacteria via Machine Learning”.

## 5.1 PDE simulations

As mentioned in the introduction of Chapter 5, the first chemotaxis case study is data collected from simulations of an *a priori known* PDE. This approach is used to validate the proposed data-driven PDE methodologies, and explore different modalities which can be useful in real world scenarios, such as the one in Subchapter 5.3. In the following, the analytical PDE model will be presented (Subchapter 5.1.1) and the specific data-driven PDE approaches explained (Subchapter 5.1.2). Results (Subchapter 5.1.3) and Conclusions (Subchapter 5.1.4) follow.

### 5.1.1 PDE model

Despite the generality and applicability of the Keller-Segel model (Eq. 5.1), the chemotactic term is in many cases intractable. To model chemotactic motion of *Escherichia coli* (*E. coli*) in heterogeneous porous media, Bhattacharjee et al., have used the following extension of the Keller-Segel model (Bhattacharjee et al., 2021):

$$\begin{aligned}\frac{\partial b}{\partial t} &= D_b \Delta b - \chi_0 \nabla \cdot [b \nabla \log F_1(c)] + b \gamma F_2(c) \\ \frac{\partial c}{\partial t} &= D_c \Delta c - b \kappa F_2(c) \\ \nabla \mathbf{J}_b(0, t) \cdot \hat{\mathbf{n}} &= 0, \nabla \mathbf{J}_b(R, t) \cdot \hat{\mathbf{n}} = 0 \\ \nabla c(0, t) \cdot \hat{\mathbf{n}} &= 0, \nabla c(R, t) \cdot \hat{\mathbf{n}} = 0,\end{aligned}\tag{5.2}$$

where in radial coordinates  $b(r, t)$  is the bacterial density,  $c(r, t)$  is the chemonutrient concentration,  $D_b$  is the bacterial diffusion coefficient,  $D_c$  is the chemonutrient diffusion coefficient,  $R$  (in the boundary conditions) is the overall domain radius,  $F_1(c) = \frac{1+c/c_-}{1+c/c_+}$  and  $F_2(c) = \frac{c}{c+c_{char}}$ ,  $\mathbf{J}_b$  is the bacterial density flux ( $\mathbf{J}_b = D_b \nabla b - \chi_0 b \nabla \log F_1(c)$ ) and  $\hat{\mathbf{n}}$  is the normal vector at the domain boundaries. Note that in this case, no flux boundary conditions imply zero gradients for both fields at the circular/cylindrical boundary. In this model, bacteria bias their motion towards the chemonutrient which they can consume (second equation). In that process, and for initial conditions corresponding to the experimental data presented here, they exhibit a macroscopic coherent propagating “bacterial wave”. Experiments can be

performed to individually estimate the parameters  $D_b, \chi_0, c_-, c_+, D_c, c_{char}, \gamma$ , as in (Bhattacharjee et al., 2021) (also see Table 5.1).

### 5.1.2 Data driven PDEs

In this Subchapter, the main data-driven tools/algorithms will be described and their use justified. These tools are used to train and test several versions of surrogate macroscopic data-driven description of the chemotactic PDE(s).

Consider a system described by  $d$  macroscopic scalar variable fields  $(u^{(1)}, \dots, u^{(d)})$ . Assuming a one-dimensional domain along the vector  $\hat{\mathbf{x}}$  (for an example in cylindrical coordinates see Figure. 5.1) discretized through  $m$  points in space ( $x$ ) and  $n$  points in time ( $t$ ), we are given  $m \cdot n$  data points in  $\mathbb{R}^d$ . Using interpolation/numerical differentiation, we can estimate the temporal derivatives  $u_t^{(1)}, \dots, u_t^{(d)}$ , as well as various order derivatives in space (first,  $u_x^{(1)}, \dots, u_x^{(d)}$ , second  $u_{xx}^{(1)}, \dots, u_{xx}^{(d)}$ , etc). We assume that we know *a priori* the largest order of relevant spatial derivatives (here, two) (Li et al., 2003), the coordinate system, and the boundary conditions (here, zero flux). We also assume that the spatiotemporal discretization satisfies the necessary criteria for a numerically converged PDE solution. Given these derivatives, we can compute all relevant local operators, such as:  $\nabla \mathbf{u}^{(i)}, \Delta u^{(i)}, i \in \{1, \dots, d\}$ . In Cartesian coordinates these operators are simply related to the spatial derivatives; but in curvilinear coordinates, or when the evolution occurs on curved manifolds, the relation between spatial derivatives and local operators needs a little more care.

We consider physical Euclidean space  $\mathbb{R}^3$  (regarded as a Riemannian manifold with Euclidean metric expressed as  $g = (dx)^2 + (dy)^2 + (dz)^2$  in Cartesian coordinates  $(x, y, z)$ ). The gradient,  $\text{grad } f$ , of a smooth function  $f: \mathbb{R}^3 \rightarrow \mathbb{R}$  is a vector pointing in the direction at which  $f$  grows at its maximal rate and whose length is said maximal rate —note that this definition is independent of the system of coordinates on  $\mathbb{R}^3$ . The phase flow of a smooth vector field  $v: \mathbb{R}^3 \rightarrow \mathbb{R}^3$  can be regarded as the motion of a fluid in space. The divergence of  $v$ , denoted by  $\text{div } v$ , is the outflow of fluid per unit volume at a given point. Again, the definition is coordinate frame independent; the expression  $\text{div } v = \frac{\partial v_1}{\partial x} + \frac{\partial v_2}{\partial y} + \frac{\partial v_3}{\partial z}$  is valid in Cartesian coordinates. The Laplacian is the composition of the gradient followed by the divergence (in other words,  $\Delta = \text{div grad}$ ), again independently of the choice of coordinates.

Our objective is to learn functions  $f_i: \mathbb{R}^{3d \cdot m} \rightarrow \mathbb{R}^m, i \in \{1, \dots, d\}$  such that:

$$u_t^{(i)} = f_i(u^{(1)}, \dots, u^{(d)}, \nabla \mathbf{u}^{(1)} \cdot \hat{\mathbf{x}}, \dots, \nabla \mathbf{u}^{(d)} \cdot \hat{\mathbf{x}}, \Delta u^{(1)}, \dots, \Delta u^{(d)})$$

This is a **black box expression** for the time derivative of a macroscopic field expressed as a function of the relevant lower order coordinate-independent local spatial operators, operating on the fields. After training (after successfully learning this function based on data) integrating this model numerically can reproduce spatiotemporal profiles in the training set, and even hopefully predict them beyond the training set. The arguments of  $f_i$  will be the features (or input vectors) and  $u_t^{(i)}$  will be the target (or output vector) of our data-driven model. Note that, usually, not all features are informative in the learning of  $f_i$  (in other words, only some orders of the spatial derivatives appear in the PDE right-hand-side). Also, note that not all macroscopic variables  $u^{(i)}$  are always necessary for learning  $f_j, j \neq i$ . In the spirit of the Whitney and Takens embedding theorems (Whitney, 1936; Takens, 1981), short histories of some of the relevant variable profiles may “substitute” for missing observations/measurements of other relevant variables.

A similar approach can be implemented when we have knowledge of a term/ some of the terms **but not of the rest of the terms** of the right-hand side. In the specific context of chemotaxis, we are interested in learning just the chemotactic term, i.e. functions  $g_i : \mathbb{R}^{3d-m} \rightarrow \mathbb{R}^m, i \in \{1, \dots, d\}$  such that:

$$u_t^{(i)} - D^{(i)}\Delta u^{(i)} = g_i(u^{(1)}, \dots, u^{(d)}, \nabla \mathbf{u}^{(1)} \cdot \hat{\mathbf{x}}, \dots, \nabla \mathbf{u}^{(d)} \cdot \hat{\mathbf{x}}, \Delta u^{(1)}, \dots, \Delta u^{(d)}),$$

where  $D^{(i)}$  is an *a priori* known diffusivity. This is now a **gray box model** for the macroscopic PDE, and is particularly useful in cases where an (effective) diffusion coefficient is easy to determine, possibly from a separate set of experiments or simulations (Lee et al., 2022). Again, as for black box models, gray boxes can also be formulated in the case of partial information, i.e. when not all fields  $u^{(i)}$  are known, by leveraging history information of the known variables.

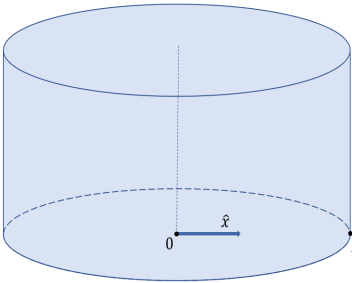
Note that in our specific case of **cylindrical coordinates**, of interest in this paper, if only the component along the radial dimension (with unit vector denoted  $\hat{\mathbf{r}}$ ) is important, the right-hand-side of a PDE will explicitly depend on the local radius as well. In the above formulation, this is incorporated in the Laplacian term  $\Delta u^{(i)} = \frac{1}{r} \frac{\partial}{\partial r} (r \frac{\partial u^{(i)}}{\partial r})$ . The construction of relevant differential operators in any coordinate system using ideas from Exterior Algebra is discussed in Subchapter 5.1.4.

### 5.1.3 Results

Numerical simulations of Eq.5.2, to provide training data for our data-driven identification approach, were performed using COMSOL Multiphysics 5.5 (Multiphysics, 1998), for the spatiotemporal domain

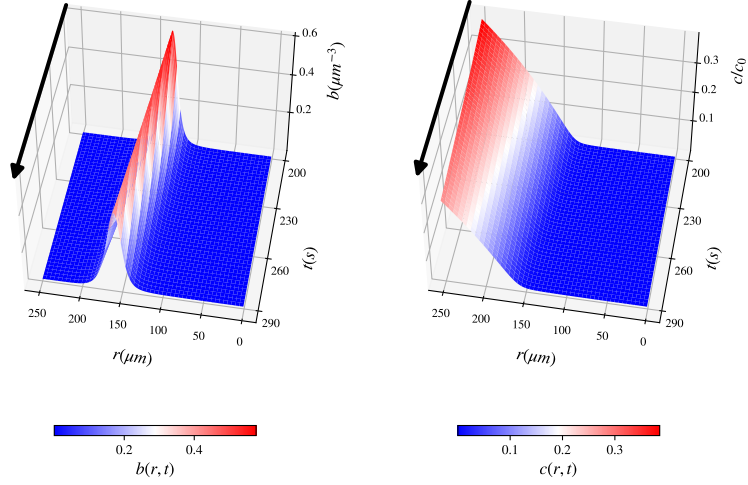
$(t, r) \in [0, 300] \times [0, 1000]$ , with initial conditions  $b(r, 0) = b_0 e^{-r^2/2\sigma^2}$ ,  $c(r, 0) = c_0$ . Note that for all subsequent learning cases, the training data are (a suitable subset) in  $(t, r) \in [200, 230] \times [0, 250]$  (therefore,  $m \cdot n \leq 15,801$  in  $\mathbb{R}^2$ ) and the model is validated by simulation in  $(t, r) \in [200, 290] \times [0, 250]$ . with spatial resolution  $dr = 0.5\mu\text{m}$  with the results reported every  $\delta t = 0.1\text{s}$  (relative tolerance set at  $10^{-7}$ ). Integration was performed using Finite Element Method and a MUMPS solver (Multiphysics, 1998). The parameters of Eq.5.2 used in the simulation are included in Table 5.1 and the results in Fig.5.3.

Parameter	Value	Unit
$D_b$	2.325	$\mu\text{m}^2/\text{s}$
$\chi_0$	17.9	$\mu\text{m}^2/\text{s}$
$c_-$	1	$\mu\text{M}$
$c_+$	30	$\mu\text{M}$
$c_{char}$	1	$\mu\text{M}$
$\gamma$	0	$\mu\text{M}/\text{s}/\mu\text{m}^3$
$D_c$	800	$\mu\text{m}^2/\text{s}$
$\kappa$	3000	$\mu\text{M}/\text{s}/\mu\text{m}^3$
$b_0$	0.95	$1/\mu\text{m}^3$
$\sigma$	42.62	$\mu\text{M}$
$c_0$	10	$\text{mM}$
$R$	17.5	$\text{mm}$



**Figure 5.2 & Table 5.1:** Parameters used for the PDE simulation of the Extended Keller-Segel Model and a schematic representation of the computational domain.

A variety of Machine Learning-enabled data-driven models was considered; they are listed in Table 5.2; the relevant notation is summarized in the Table caption. Note that for each model, two supervised learning algorithms are used to validate the results: An Artificial Neural Network and Gaussian Process Regression.



**Figure 5.3:** PDE simulations representing the ground truth data: (left)  $b(r, t)$  field and (right)  $c(r, t)$  field. For clarity, arrows are added to denote the direction of time.

Model	Surrogate Function	Known Fields	Known RHSs	Output	Algorithm
Black-box for 2 PDEs	$f_{GP}, h_{GP}$	$b(r, t), c(r, t)$	–	$b_t, c_t$	GPR
	$f_{NN}, h_{NN}$	$b(r, t), c(r, t)$	–	$b_t, c_t$	ANN
Black-box for 1 PDE	$f_{GP}$	$b(r, t), c(r, t)$	$c_t$	$b_t$	GPR
	$f_{NN}$	$b(r, t), c(r, t)$	$c_t$	$b_t$	ANN
Black-box, delays	$f_{GP}^{partial}$	$b(r, t), \text{history}$	–	$b(r, t + \Delta t)$	GPR
	$f_{NN}^{partial}$	$b(r, t), \text{history}$	–	$b(r, t + \Delta t)$	ANN
Gray-box	$g_{GP}$	$b(r, t), c(r, t)$	$c_t$	$b_t - D_b \Delta b$	GPR
	$g_{NN}$	$b(r, t), c(r, t)$	$c_t$	$b_t - D_b \Delta b$	ANN
Gray-box, delays	$g_{GP}^{partial}$	$b(r, t), \text{history}$	–	$b(r, t + \Delta t) - D_b \Delta b(r, t)$	GPR
	$g_{NN}^{partial}$	$b(r, t), \text{history}$	–	$b(r, t + \Delta t) - D_b \Delta b(r, t)$	ANN

**Table 5.2:** Listing of the data-driven models explored. Notation:  $f, h$  denote surrogate functions for the entire RHS of the  $b$ - and  $c$ -PDE respectively, while  $g$  denotes a surrogate for the chemotactic term. Subscripts “GPR” and “NN” denote Gaussian Process Regression and Artificial Neural Network respectively.  $\Delta t$  denotes the time delay used in all models with partial information.

Black box learning of both PDEs with an ANN (from known fields  $b(r, t), c(r, t)$ )

$$\begin{bmatrix} b_t \\ c_t \end{bmatrix} = \begin{bmatrix} f \\ h \end{bmatrix} = \mathbf{F}_{\text{NN}}(b, \nabla \mathbf{b} \cdot \hat{\mathbf{r}}, \Delta b, c, \nabla \mathbf{c} \cdot \hat{\mathbf{r}}, \Delta c) \quad (5.3)$$

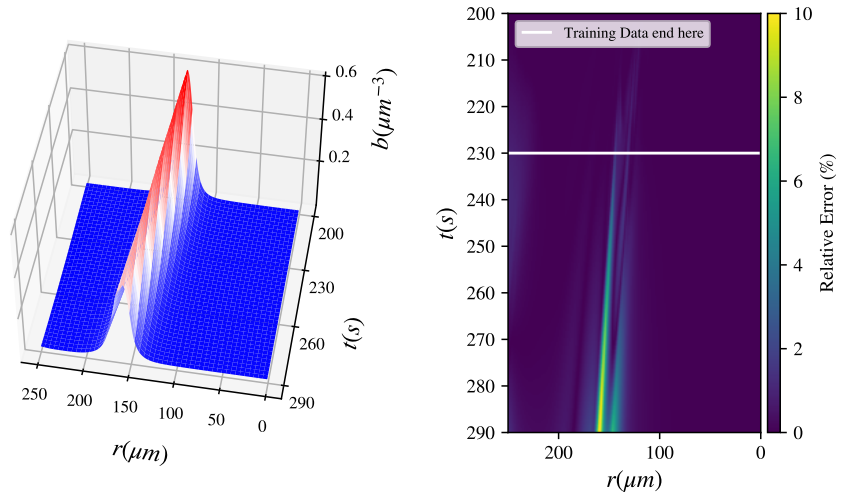


Figure 5.4: Black-box learning of both PDEs with an Artificial Neural Network: (left) Integration results for the **first** data-driven PDE (for  $b(r, t)$ ) and (right) relative error (%).

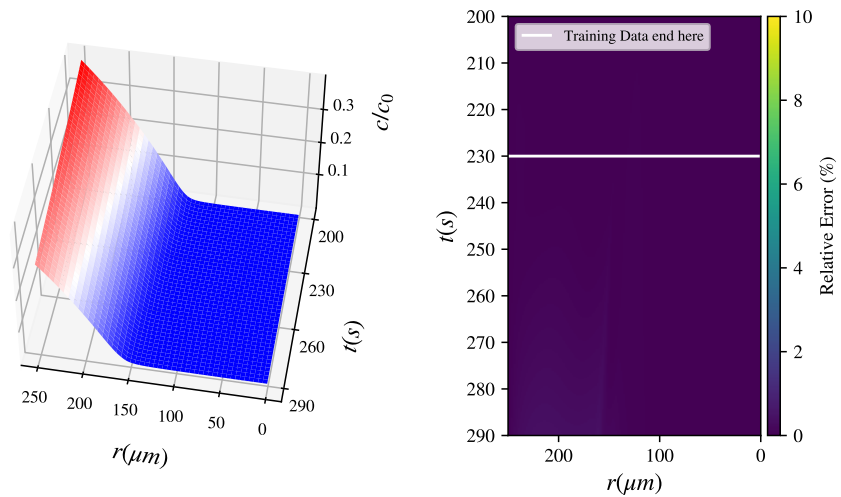
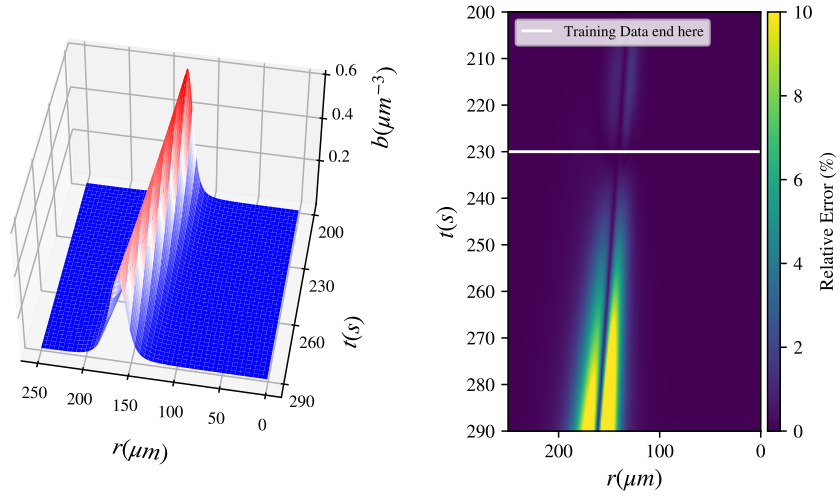


Figure 5.5: Black-box learning of both PDEs with an Artificial Neural Network: (left) Integration results for the **second** data-driven PDE (for  $c(r, t)$ ) and (right) relative error (%).

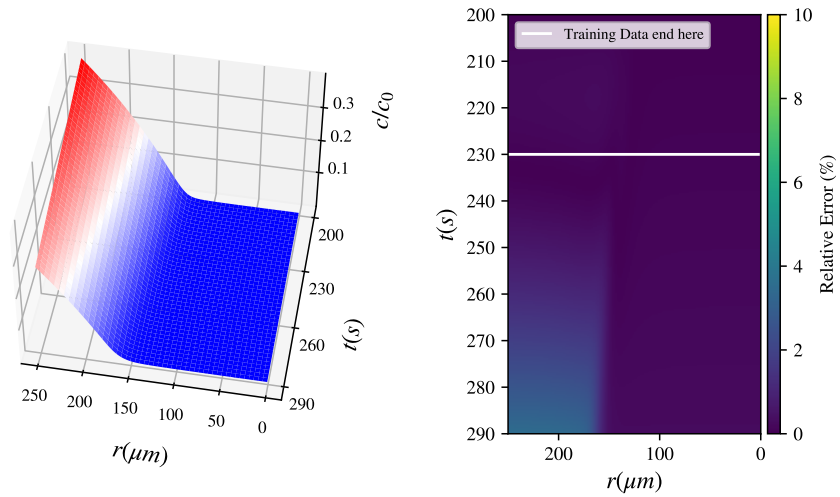


Black box learning of both PDEs with GPR (from known fields  $b(r, t), c(r, t)$ )

$$\begin{bmatrix} b_t \\ c_t \end{bmatrix} = \begin{bmatrix} f_{GP}(b, \nabla \mathbf{b} \cdot \hat{\mathbf{x}}, \Delta b, c, \nabla \mathbf{c} \cdot \hat{\mathbf{x}}, \Delta c) \\ h_{GP}(b, \nabla \mathbf{b} \cdot \hat{\mathbf{x}}, \Delta b, c, \nabla \mathbf{c} \cdot \hat{\mathbf{x}}, \Delta c) \end{bmatrix} \quad (5.4)$$



**Figure 5.6:** Black-box learning of both PDEs with Gaussian Process Regression: (left) Integration results for the **first** data-driven PDE (for  $b(r, t)$ ) and (right) relative error (%). Note that the white horizontal line separates the training data set from the rest of the validation data set.



**Figure 5.7:** Black-box learning of both PDEs with Gaussian Process Regression: (left) Integration results for the **second** data-driven PDE (for  $c(r, t)$ ) and (right) relative error (%).

With training data from both  $b(r, t)$  and  $c(r, t)$  fields, a neural network was trained of the form described in Eq.5.3 in Python’s Tensorflow platform (Abadi et al., 2015). Training was performed with a feedforward neural network consisting of two hidden layers, each with 18 tanh-activated neurons. An Adams optimizer (Kingma and Ba, 2014) was used with a MSE loss function. The neural network hyperparameters were empirically tuned: 2048 epochs and 0.02 learning rate. After training, the data-driven PDE was integrated with a commercial BDF integrator, as implemented in Python’s *solve\_ivp* (Shampine and Reichelt, 1997), with relative and absolute tolerances at  $(10^{-4}, 10^{-7})$ . The initial profile for integration was supplied by our simulation data, and the boundary conditions set to no flux. The Jacobian of the data-driven PDE was provided via automatic differentiation.

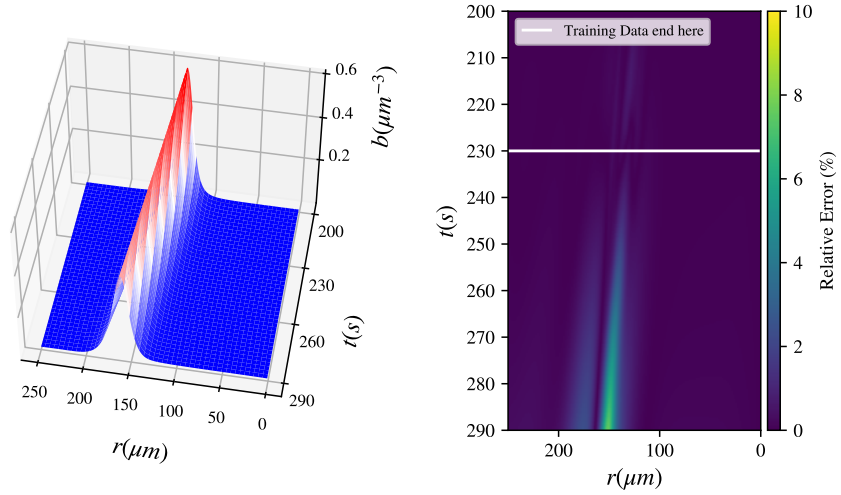
Figures 5.4, 5.5 show how the data-driven PDEs are able to learn the laws of time evolution of both  $b(r, t)$  and  $c(r, t)$ . The data-driven models can be used to reproduce the training data and successfully extrapolate as far as we attempted (here, up to  $t = 290$ s).

GPR was trained in Python’s scikit-learn framework (Pedregosa et al., 2011) with hyperparameter  $\alpha = 10^{-4}$  and an L-BFGS-B optimizer (Zhu et al., 1997). Temporal profiles used in training were subsampled (2 used in GP for every 5 in ANN). The resulting data-driven PDE was integrated similarly to ANN (without the Jacobian of the right-hand-side).

Figures 5.6, 5.7 show how the data-driven PDEs are able to learn the laws of time evolution of both  $b(r, t)$  and  $c(r, t)$  through Gaussian Process Regression. The data-driven models can be used to reproduce the training data and successfully extrapolate as far as we attempted (here, up to  $t = 290$ s).

**Black box learning of  $b_t$  with an ANN with known  $c_t$  (from known fields  $b(r, t), c(r, t)$ )**

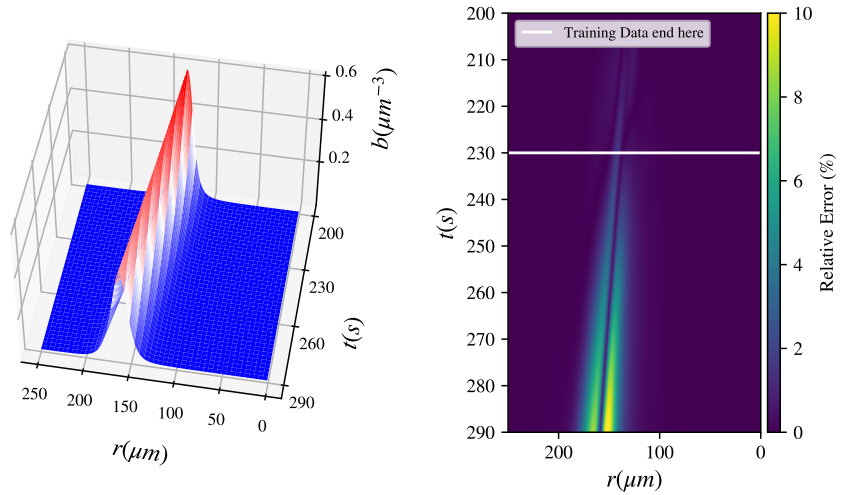
$$b_t = f_{NN}(b, \nabla \mathbf{b} \cdot \hat{\mathbf{r}}, \Delta b, c, \nabla \mathbf{c} \cdot \hat{\mathbf{r}}, \Delta c) \quad (5.5)$$



**Figure 5.8:** Black-box learning with a Neural Network: (left) Integration results for the data-driven PDE and (right) % relative error.

**Black box learning of  $b_t$  with GPR with known  $c_t$  (from known fields  $b(r, t), c(r, t)$ )**

$$b_t = f_{GP}(b, \nabla \mathbf{b} \cdot \hat{\mathbf{r}}, \Delta b, c, \nabla \mathbf{c} \cdot \hat{\mathbf{r}}, \Delta c) \quad (5.6)$$



**Figure 5.9:** Black-box learning with Gaussian Process Regression: (left) Integration results for the data-driven PDE and (right) % relative error.

Figures 5.8, 5.9 showcase learning of *one of the two* data-driven PDEs when the second PDE is known.

Here, the chemonutrient ( $c(r, t)$ ) PDE is assumed known, and the bacterial density evolution PDE is learned with a neural network or Gaussian Process Regression.

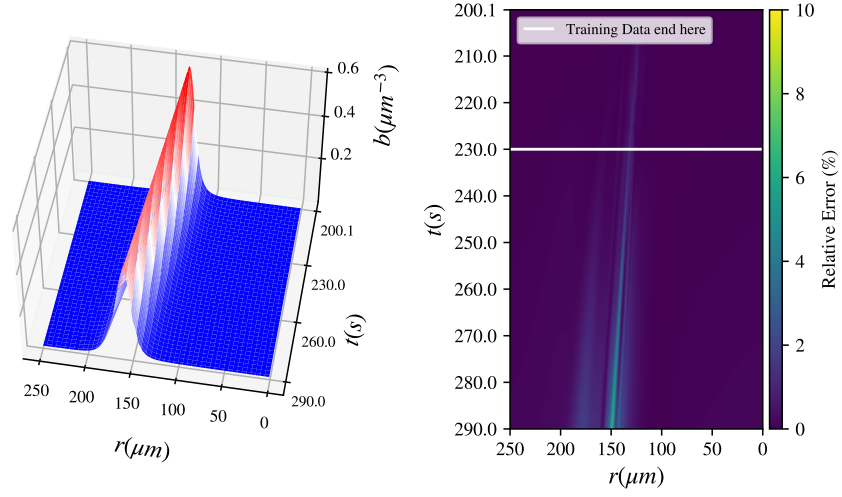
After the initial success of the previous section, we decided to attempt a computational experiment, based on the spirit of the Takens embedding for finite-dimensional dynamical systems (Takens, 1981; Whitney, 1936; Stark et al., 1997; Stark, 1999; Stark et al., 2003; Sauer, Yorke, and Casdagli, 1991).

The idea here is that, if only observations of a few (or even only one) variables involved are available, one can use history of these observations (“time-delay” measurements) to create a useful latent space in which to embed the dynamics -and in which, therefore, to learn a surrogate model with less variables, but involving also history of these variables (Packard et al., 1980; Aeyels, 1981). There are important assumptions here: finite (even low) dimensionality of the long-term dynamics, something easier to contemplate for ODEs, but possible for PDEs with attracting, low dimensional (possibly inertial) manifolds for their long-term dynamics. There is also the assumption that the variable whose values and history we measure is a *generic observer* for the dynamics on this manifold. One can always claim that, if a 100-point finite different discretization of our problem is deemed accurate (so, for two fields, 200 degrees of freedom), then the current discretized observation of one of the two fields (100 measurements) plus three delayed observations of it ( $3 \times 100$ ) plus possibly one more measurement give us enough variables for a useful latent space in which to learn a surrogate model. Here we attempted to do it with much less: at each discretization point we attempted keeping the current  $b(r, t)$  field measurement and its spatial derivatives, and added only some history (the values and spatial derivatives at the previous moment in time). The equation below is written in discrete time form (notice the dependence of the field at the next time step from two previous time steps); it can be thought of as a discretization of a *second order* partial differential equation for the  $b(r, t)$  field, based on its current value and its recent history.

**Black box ANN learning of a single field evolution equation, with only partial information (only the field  $b(r, t)$  is observed).**

$$b(t_{k+1}) = b(t_k) + \Delta t f_{NN}^{partial}(b(t_k), (\nabla \mathbf{b} \cdot \hat{\mathbf{r}})(t_k), (\Delta b)(t_k), b(t_{k-1}), (\nabla \mathbf{b} \cdot \hat{\mathbf{r}})(t_{k-1}), (\Delta b)(t_{k-1})), \quad (5.7)$$

with  $\Delta t = t_{k+1} - t_k$ , for any time point  $t_k, k \geq 1$ .

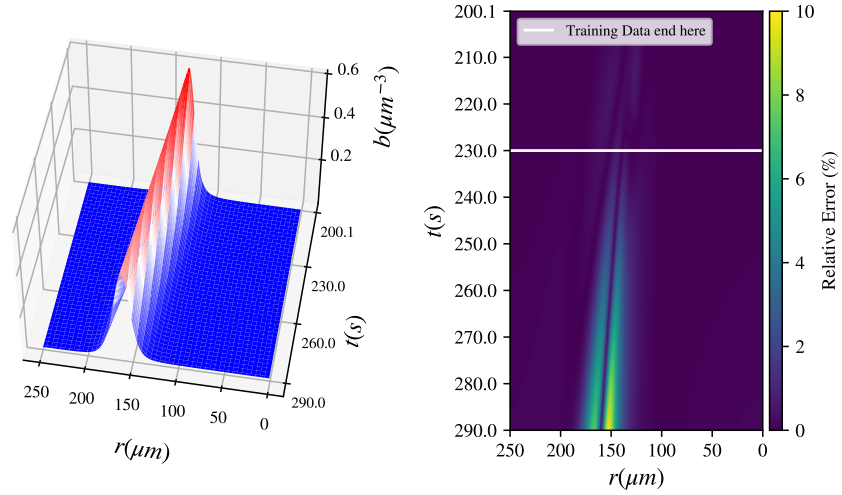


**Figure 5.10:** Black-box partial-information learning with a Neural Network: (left) Integration results for the data-driven PDE and (right) % relative error.

**Black box GPR learning of a single field evolution equation, with only partial information (only the field  $b(r, t)$  is observed).**

$$b(t_{k+1}) = b(t_k) + \Delta t f_{GP}^{partial}(b(t_k), (\nabla \mathbf{b} \cdot \hat{\mathbf{r}})(t_k), (\Delta b)(t_k), b(t_{k-1}), (\nabla \mathbf{b} \cdot \hat{\mathbf{r}})(t_{k-1}), (\Delta b)(t_{k-1})), \quad (5.8)$$

with  $\Delta t = t_{k+1} - t_k$ , for any time point  $t_k, k \geq 1$ .

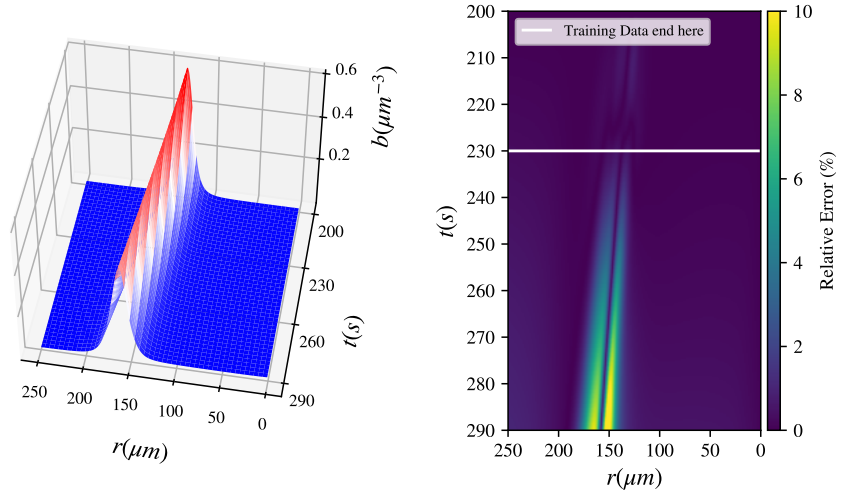


**Figure 5.11:** Black-box partial-information learning with Gaussian Process Regression: (left) Integration results for the data-driven PDE and (right) % relative error.

Figures 5.10, 5.11 demonstrate learning a data-driven (discrete in time here) evolution equation for the bacterial density  $b(r, t)$  when only data for  $b(r, t)$  are at hand (partial information). Even though we know the existence of another, physically coupled field, we cannot sample from it, so we replace its effect on the  $b(r, t)$  field through a functional dependence on the history of  $b(r, t)$ . Simulation of the resulting model was successful in reproducing the training data and extrapolating beyond them in time.

**Gray box learning with an ANN -  $c_t$  known (with fields  $b(r, t), c(r, t)$  known).**

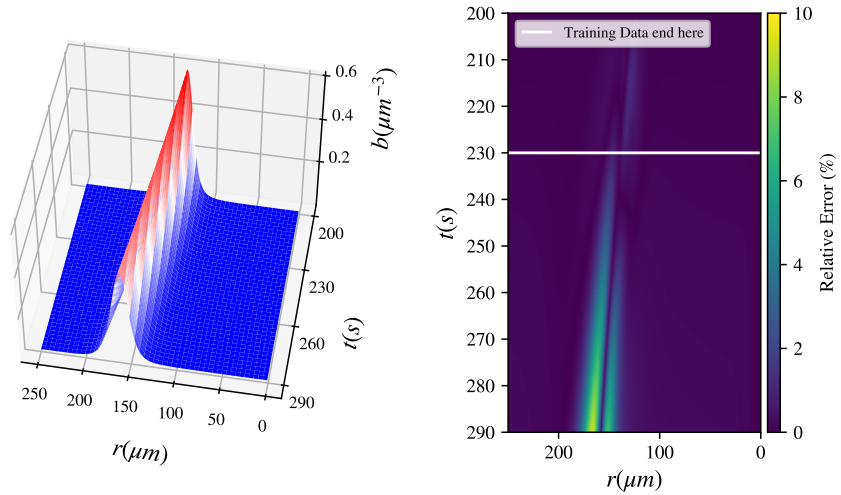
$$b_t - D_b \Delta b = g_{NN}(b, \nabla \mathbf{b} \cdot \hat{\mathbf{r}}, \Delta b, c, \nabla \mathbf{c} \cdot \hat{\mathbf{r}}, \Delta c) \quad (5.9)$$



**Figure 5.12:** Gray-box learning with a Neural Network: (left) Integration results for the data-driven PDE and (right) % relative error.

**Gray box learning with GPR -  $c_t$  known (with fields  $b(r, t), c(r, t)$  known).**

$$b_t - D_b \Delta b = g_{GP}(b, \nabla \mathbf{b} \cdot \hat{\mathbf{r}}, \Delta b, c, \nabla c \cdot \hat{\mathbf{r}}, \Delta c) \quad (5.10)$$



**Figure 5.13:** Gray-box learning with Gaussian Process Regression: (left) Integration results for the data-driven PDE and (right) % relative error.

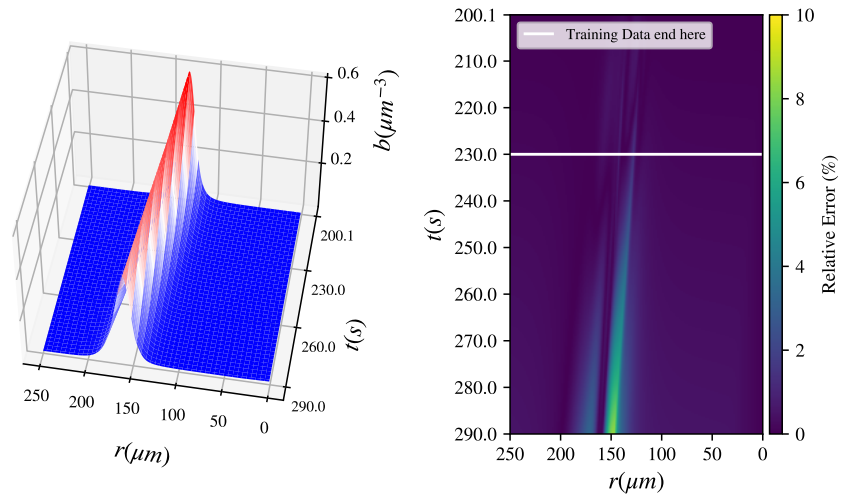
Figures 5.12, 5.13 show the performance of gray-box models, where only the chemotactic term of the

bacteria density PDE is considered unknown. For this gray-box model, the effective diffusion coefficient for the bacterial density is considered known. In principle, one can also hardwire the knowledge of the *functional form of this term* in the loss of the neural network, and thus obtain an estimate of this diffusivity in addition to learning the chemotactic term in the PDE.

**Gray box ANN learning of the chemotactic term, with only partial information (only the field  $b(r, t)$  is observed).**

$$b(t_{k+1}) = b(t_k) + \Delta t (D_b \Delta b(t_k) + g_{NN}^{partial}(b(t_k), (\nabla \mathbf{b} \cdot \hat{\mathbf{r}})(t_k), (\Delta b)(t_k), b(t_{k-1}), (\nabla \mathbf{b} \cdot \hat{\mathbf{r}})(t_{k-1}), (\Delta b)(t_{k-1}))), \quad (5.11)$$

with  $\Delta t = t_{k+1} - t_k$ , for any time point  $t_k, k \geq 1$ .



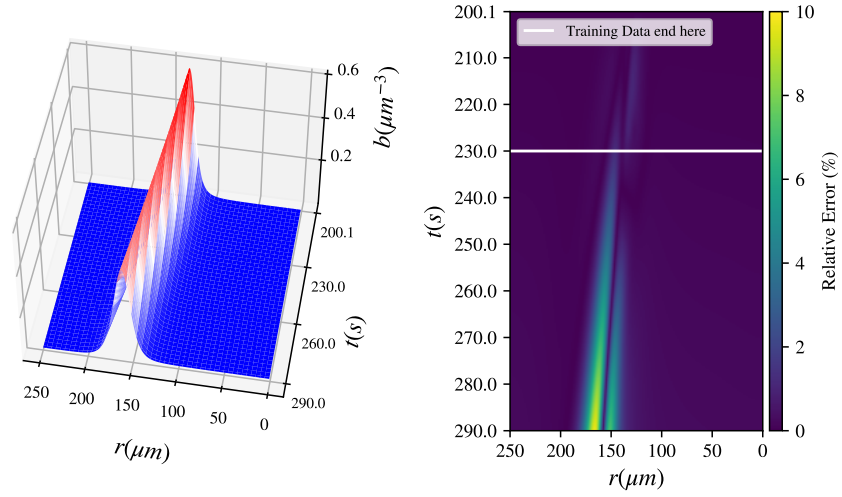
**Figure 5.14:** Gray-box partial-information learning with a Neural Network: (left) Integration results for the data-driven PDE and (right) % relative error.

**Gray box GPR learning of the chemotactic term, with only partial information (only the field  $b(r, t)$  is observed).**



$$b(t_{k+1}) = b(t_k) + \Delta t (D_b \Delta b(t_k) + g_{GP}^{partial}(b(t_k), (\nabla \mathbf{b} \cdot \hat{\mathbf{r}})(t_k), (\Delta b)(t_k), b(t_{k-1}), (\nabla \mathbf{b} \cdot \hat{\mathbf{r}})(t_{k-1}), (\Delta b)(t_{k-1}))), \quad (5.12)$$

with  $\Delta t = t_{k+1} - t_k$ , for any time point  $t_k, k \geq 1$ .



**Figure 5.15:** Gray-box partial-information learning with Gaussian Process Regression: (left) Integration results for the data-driven PDE and (right) % relative error.

Figures 5.14 , 5.15 demonstrate how Gray box models can also be constructed with partial information, i.e. with knowledge only of the bacteria density field  $b(r, t)$ .

**Estimating the chemonutrient field given measurements of the bacterial density ( $b(r, t)$  and its recent history).**

Surrogate Function	Known Fields	Known RHSs	Output	Algorithm
$C_{GP}$	$b(r, t)$ , history	–	$c(r, t)$	GPR
$C_{NN}$	$b(r, t)$ , history	–	$c(r, t)$	ANN

**Table 5.3:** Summary of all data-driven models for C-learning.

Following up the above success in using Takens’ embeddings (and more generally, Whitney embeddings) (Takens, 1981; Whitney, 1936) for low-dimensional long-term dynamics, we now attempt to estimate (i.e. create a nonlinear observer - a “soft sensor” of) the chemonutrient field from local measurements of the bacterial fields and its history (Altaf et al., 2017; Farhat, Lunasin, and Titi, 2017). More specifically, we will attempt to train a neural network to learn (in a data driven manner)  $c(r_i, t_k)$  as a function of some local space time information:

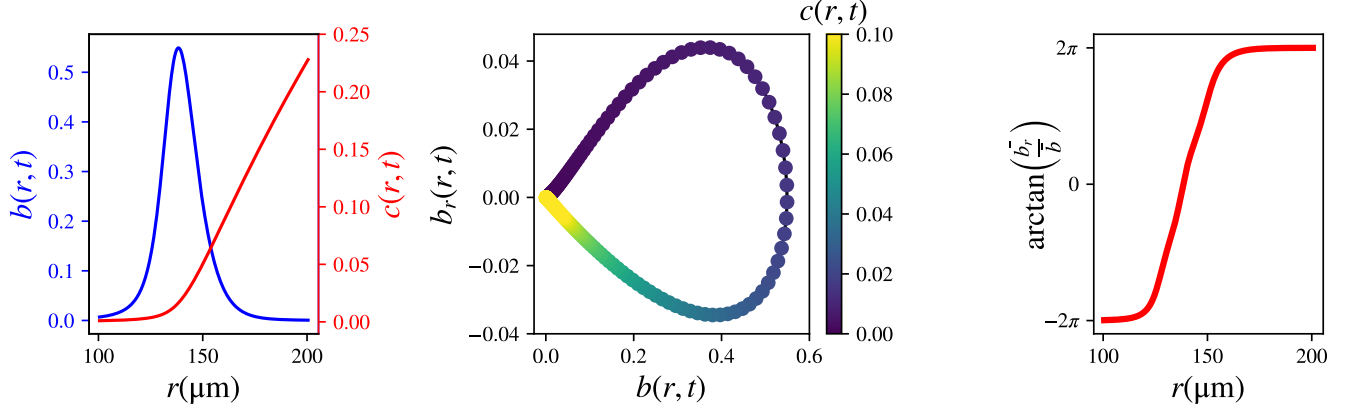
$$c(r_i, t_k) = C_{NN}(b(r_i, t_k), (\nabla \mathbf{b} \cdot \hat{\mathbf{r}})(r_i, t_k), (\Delta b)(r_i, t_k), \\ b(r_i, t_{k-1}), (\nabla \mathbf{b} \cdot \hat{\mathbf{r}})(r_i, t_{k-1}), (\Delta b)(r_i, t_{k-1})), \quad (6a)$$

for any discretization point in space  $r_i$  and time point  $t_k, k \geq 1$ .

Indeed, if the long-term simulation data live on a low-dimensional, say  $m$ -dimensional manifold, then  $2m + 1$  generic observables suffice to embed the manifold, and then learn *any* function on the manifold in terms of these observables. Here we will attempt an experiment with a neural network that uses a *local* parametrization of this manifold, testing if such a local parametrization can be learned (in the spirit of the nonlinear discretization schemes of Brenner et al. (Bar-Sinai et al., 2019)).

There is, however, one particular technical difficulty: because the long-term dynamics of our problem appear in the form of travelling waves, both the front and the back of the wave appear practically *flat* – the spatial derivatives are all approximately zero, and a simple neural network cannot easily distinguish, from local data, if we find ourselves in the flat part *in front* of the wave or *behind* the wave. We therefore constructed an additional local observable, capable of bringing up the difference of flat profiles “before” and flat profiles “after” the wave. Indeed, when the data represent the spatiotemporal evolution of a traveling wave (as in our training/testing data set), we expect a **singularity close to**  $b(r, t) = 0$ . This singularity arises at both “tails” of the traveling bacterial wave where  $b$  and its derivatives all become asymptotically zero. Clearly, however, the  $c$  field is dramatically different on the two “flat bacteria” sides. When learning such a function **locally**, to circumvent this singularity, we came up with a transformation of two of the inputs:  $(b, \nabla \mathbf{b} \cdot \hat{\mathbf{r}}) \rightarrow \left( b, \arctan\left(\frac{\overline{(\nabla \mathbf{b} \cdot \hat{\mathbf{r}})}}{b}\right) \right)$ , where the bar symbol denotes an affine transformation of the respective entire feature vector to the interval  $[-1, 1]$ . This transformation brings points at different sides of the aforementioned singularity at different ends of a line, exploiting

their difference in sign (see Fig.5.16).

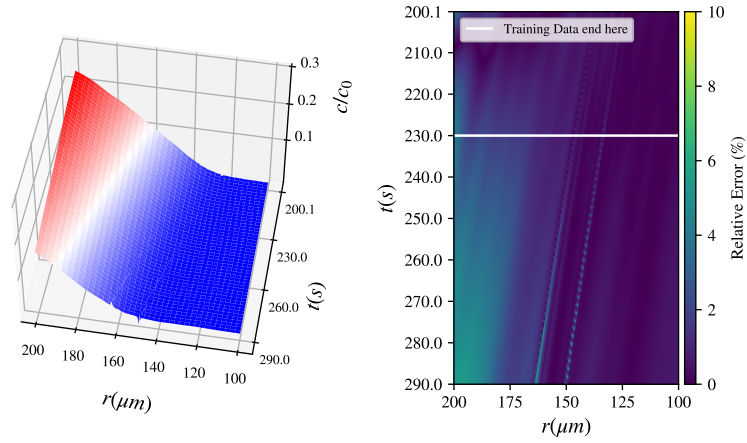


**Figure 5.16:** Transformation on the inputs: (left) representative profiles of both fields  $b(r,t), c(r,t)$ , (middle) visualization of the singularity, (right) transformed variable.

Then, the Neural Network is trained to learn the estimator (nonlinear observer) of the chemonutrient field as:

$$c(r_i, t_k) = C_{NN}(b(r_i, t_k), \arctan\left(\frac{(\nabla \mathbf{b} \cdot \hat{\mathbf{r}})(r_i, t_k)}{b(r_i, t_k)}\right), (\Delta b)(r_i, t_k), \\ b(r_i, t_{k-1}), (\nabla \mathbf{b} \cdot \hat{\mathbf{r}})(r_i, t_{k-1}), (\Delta b)(r_i, t_{k-1})), \quad (6b)$$

for any discretization point in space  $r_i$  and time point  $t_k, k \geq 1$ .



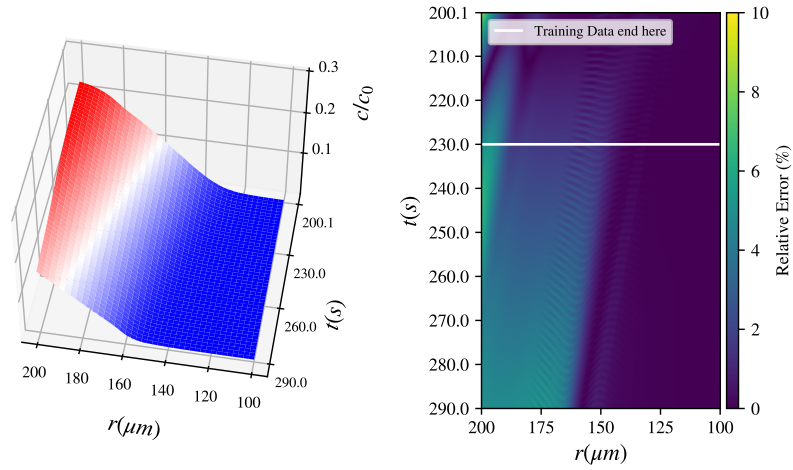
**Figure 5.17:** Learning the  $c$ -field with a Neural Network: (left) Field prediction and (right) % relative error.

Analogously, for the GPR learning approach:

$$c(r_i, t_k) = C_{GP}(b(r_i, t_k), \arctan \left( \frac{(\nabla \mathbf{b} \cdot \hat{\mathbf{r}})(r_i, t_k)}{b(r_i, t_k)} \right), (\Delta b)(r_i, t_k),$$

$$b(r_i, t_{k-1}), (\nabla \mathbf{b} \cdot \hat{\mathbf{r}})(r_i, t_{k-1}), (\Delta b)(r_i, t_{k-1})),$$

for any discretization point in space  $r_i$  and time point  $t_k, k \geq 1$ .



**Figure 5.18:** Learning the  $c$ -field with Gaussian Process Regression: (left) Field prediction and (right) % relative error.

As it can be seen from Figs.5.17, 5.18 through the model in Eqs.6b, ?? it was possible to provide reasonable predictions for the chemonutrient field.

## 5.1.4 Discussion and Conclusions

A lot of interesting observations can be drawn from comparing different models or methodologies in Subchapter 5.1.3.

- For the model in Eq.5.3, where both PDEs are data-driven it is obvious that the  $c(r, t)$  PDE is easier to learn and predict than the  $b(r, t)$  PDE, both for the ANN and the GPR. This can be understood in terms of complexity of these two target functions: The  $b_t$  expression is highly nonlinear (owing mostly to the logarithmic chemotactic term) and complex, as it depends on

most of the input features. On the contrary the  $c_t$  expression only depends on a handful of inputs and is less complicated.

- When learning both PDEs (model in Eq.5.3), it is straightforward to employ an ANN for multiple output prediction. However, multiple output GP (or co-kriging) is especially non-trivial Wang and Chen, 2015 and therefore, multiple single-output GPs are preferred instead.
- When comparing the ANN methodology with the GPR, for the same model, it can be seen that ANN results in more accurate predictions than GPR. may be attributed to the ANN's versatility and efficiency in capturing the nonlinearities and complexities of any target function. It is also important to note that, due to memory constraints, GPR was trained only with an appropriately chosen subset of the training data. This could cause the loss of accuracy in long term predictions. However, it is important to note, that the error in GPR is always *smooth*, owing to the smoothness of the Gaussian kernel (see Subchapter 2.2.1.1).
- Comparing the predictions for  $b_t$  in Figs.5.4, 5.6 and Figs. 5.8, 5.9 it can be seen that when only *one* of the PDEs is data-driven, the predictions are more accurate. This can be understood in terms of error accumulation during integration: when both PDEs are data-driven, both of the PDEs contribute to prediction error which will propagate along the integration trajectory.
- In the case of partial information, the model was trained with a single (therefore, fixed) delay ( $t_k - t_{k-1}$ ), and this imposes important restrictions on how we can advance the data-driven model in time. A natural way to do this, is with a Forward Euler scheme with a timestep equal to the delay used in training, as explicitly shown in Eqs. 5.7, 5.8, 5.11, 5.12.

In this work, it was demonstrated how data-driven models are able to learn PDEs from bacterial migration data (even in the case of partial information). It is also possible to learn just one term of a PDE, or a certain PDE out of a set of coupled PDEs. These data-driven models were able to reproduce spatiotemporal profiles used for training and extrapolate further in time. This work showcases that data-driven PDEs are a versatile tool which can be adjusted and implemented to many different problem settings, data sets or learning objectives. It can be especially useful (if not necessary) when the derivation of an analytical PDE is cumbersome or when there is no capacity for a large number of simulations or experiments. In fact, data-driven PDEs can be used to estimate transients from different (nearby) Initial Conditions, for different Boundary Conditions or spatial domains. Apart from that, learning data-driven PDEs is one of the most compact and generalizable ways to learn a system's behavior from data.

Future work includes removing some of the assumptions used here to make data-driven PDEs more robust and generalizable. For example, it is possible to limit the number of independent, relevant inputs in a data-driven way, using dimensionality reduction, automatic relevance determination or other feature importance methods (Rasmussen and Williams, 2005; Ghorbani, Abid, and Zou, 2019).

Another important direction is to train these models in a *coordinate - free* way (Luk and Grosse, 2020; Weiler et al., 2021) using tools from Exterior Calculus. Here, we give a brief outlook on the use of differential operators arising from exterior calculus (Flanders, 1989; Lee, 2009) to create a dictionary of features in which to express learned operators: The gradient, curl, divergence, and Laplacian all arise by combining the exterior derivative  $d$ , the metric tensor  $g$ , the exterior product  $\wedge$  and the inner product of differential forms  $\langle \cdot, \cdot \rangle$ , the Hodge star operator  $\star$ , as well as the musical isomorphisms  $\sharp$  and  $\flat$ . Consequently, we can extend the problem of learning  $f$  in

$$u = f(u, \text{grad } u, \text{div grad } u, \dots)$$

to the more general setting of learning a function  $f$  defined by compositions of the operators mentioned above. For instance, the Navier-Stokes equations (Gurtin, 1981) can be written in coordinate-free form as (Wilson, 2011)

$$\begin{cases} \frac{\partial \omega}{\partial t} = -\star(\omega \wedge \star d\omega) - \nu d\star d\omega + \frac{1}{2}d\langle \omega, \omega \rangle - dp \\ d\star \omega = 0, \end{cases}$$

where  $\nu \geq 0$  is the viscosity and  $p$  is the pressure.

The use of exterior calculus and exterior differential systems (Bryant et al., 1991) in Physics-informed neural networks is currently growing (Sitzmann et al., 2020; Weiler et al., 2021; Jenner and Weiler, 2021; Bronstein et al., 2021) and a more in-depth study of this framework is an interesting future direction.

## 5.2 Agent based modeling

After establishing and validating our approach with data collected from PDE simulations in Subchapter 5.1, we turn to data from microscale simulations: high-fidelity data, produced by a detailed biophysical Monte-Carlo model for the motility of *E. coli* bacteria. The parameters of this model are calibrated from experimental data (Berg and Brown, 1972; Larsen et al., 1974; Maeda et al., 1976; Block, Segall, and Berg, 1983; Block, Segall, and Berg, 1982; Ishihara et al., 1983; Spiro, Parkinson, and Othmer, 1997; Othmer and Schaap, 1998; Cluzel, Surette, and Leibler, 2000; Othmer, Xin, and Xue, 2013). Building on previous efforts (Rico-Martinez, Anderson, and Kevrekidis, 1994; Krischer et al., 1993; González-García, Rico-Martínez, and Kevrekidis, 1998; Bertalan et al., 2019; Lee et al., 2020; Kemeth et al., 2020), we propose a data-based, Machine Learning assisted framework to learn the law of the underlying macroscopic PDE. In particular, based on automatic relevance determination (ARD) (MacKay, 1992; Sandhu et al., 2017) within the Gaussian Process Regression framework (Rasmussen and Williams, 2005; Lee et al., 2020) for feature extraction, and on Gaussian Process Regression and Artificial Neural Networks to learn the collective dynamics (algorithms described in Subchapter 2.2.1), we:

- (a) identify the right-hand-side (RHS) of an effective Keller-Segel-class PDE, thus obtaining a *black-box* PDE model;
- (b) reconstruct the chemotactic term only, assuming that the diffusion term can be estimated by the high-fidelity simulations and/or knowledge of the physics, thus constructing a *gray box* Keller-Segel-class PDE model; and importantly,
- (c) discover data-driven corrections to established approximate closure approximations of the chemotactic term, which have been derived analytically from kinetic theory/statistical mechanics based on series of assumptions (Erban and Othmer, 2004).

### 5.2.1 ABM model

A previously derived bio-mechanical-based Monte-Carlo dynamical model (Othmer, Xin, and Xue, 2013) was used to generate data for the chemotactic motility of *E. Coli* bacteria in response to a fixed chemoattractant substrate profile. The model calculates the probability of rotational directionality of each one of the six flagellae that extend from the surface of the cell based on changes of the concentration of the CheY-P protein, which binds to the protein FliM at the base of the rotor. The changes in the concentration of the CheY-P protein control the direction of the flagellar rotation (Othmer, Xin, and Xue,

2013; Sarkar, Paul, and Blair, 2010), which in turn governs the motility of the cells. When the majority of the flagellar filaments rotate in the counter-clockwise (CCW) direction the cell swims; otherwise it tumbles. The change in time of the concentration of CheY-P protein, say  $C(t)$  is represented by a simple algebraic relation reading (Setayeshgar et al., 2005; Siettos, 2014):  $C(t) = \bar{C} - gu_1(t)$ , where the dynamics of  $u_1(t)$  are modeled by a simple two dimensional excitation/adaptation cartoon model given by:

$$\frac{du_1}{dt} = \frac{f(s) - (u_1 + u_2)}{\tau_e}, \quad \frac{du_2}{dt} = \frac{f(s) - u_2}{\tau_a}, \quad f(s) = k \frac{s}{K_s + s}. \quad (5.13)$$

In the above model,  $\bar{C} = 2.95\mu M$  is the baseline concentration corresponding to the non-excited state (Cluzel, Surette, and Leibler, 2000; Setayeshgar et al., 2005),  $g = 5$  is the amplification response to excitation (Setayeshgar et al., 2005; Othmer, Xin, and Xue, 2013),  $s$  represents the external stimulus (here, a chemoattractant substrate),  $t_e = 0.1$  and  $t_a = 20$  represent the excitation and adaptation time constants, respectively, and  $f$  is the function encoding the signal transduction reflecting the fraction of receptors that are occupied (Othmer, Xin, and Xue, 2013; Erban and Othmer, 2004);  $k = 15$  is a constant that amplifies the input signal and  $K_s = 1\mu M$  is the dissociation constant for the enzyme-substrate complex (Erban and Othmer, 2004; Block, Segall, and Berg, 1983; Setayeshgar et al., 2005). The swimming speed ( $v$ ) also depends on various parameters such as the bacteria strain, the substrate, temperature and density of cells and may vary from  $\sim 10$  to  $\sim 55$  m/s (Berg and Brown, 1972; Maeda et al., 1976). For our simulations, we have set  $v = 30$  m/s. Finally, we note that in the MC model, the movement of the bacteria is not affected by their density (they are “noninteracting”). For an analogous to the above microscopic description of the *E.coli* motility model, Erban and Othmer (Erban and Othmer, 2004) with the aid of statistical mechanics/ kinetic theory, and assuming that the the signal  $s(x)$  is a time independent scalar function, have derived the following parabolic Keller-Segel-class PDE in closed form in 1D:

$$\frac{\partial b}{\partial t} = \frac{\partial}{\partial x} \left( \frac{\bar{v}^2}{2\lambda_0} \frac{\partial b}{\partial x} - \frac{df}{ds} \frac{c\bar{v}^2 t_a}{\lambda_0(1 + 2\lambda_0 t_a)(1 + 2\lambda_0 t_e)} \frac{ds}{dx} b \right). \quad (5.14)$$

Here,  $b = b(x, t)$  is the density of the population at  $x$  and time  $t$ ,  $\bar{v}$  is the *mean* speed of the bacterium’s motion,  $\lambda_0$  represents the basal turning frequency (frequency of tumbles) in the absence of excitation, and  $c$  is a positive constant parameter that amplifies the excitation signal  $u_1$  that governs the switching frequency in the presence of a stimulus (for a detailed description of the derivation of the above PDE and its parameters please refer to (Erban and Othmer, 2004)). Based on the above, we define the *generalized*



Chemotactic term  $CH_g$  as

$$CH_g \equiv \frac{\partial}{\partial x} \left( \frac{df}{ds} \frac{c\bar{v}^2 t_a}{\lambda_0(1+2\lambda_0 t_a)(1+2\lambda_0 t_e)} \frac{ds}{dx} b \right); \quad (5.15)$$

the subscript  $g$  refers to the generalized Keller-Segel PDE. The main assumptions made for the above closed form PDE are the following: (a) all bacteria are running with a constant *velocity*  $\bar{v}$ , without colliding, (b) the tumble phase is neglected, (c) the internal excitation-adaptation dynamics are described by the 2D cartoon model, (d) the distribution of the substrate  $s(x)$  is a time-independent scalar function, (e) the turning frequency (frequency of tumbles in the presence of stimulus), say,  $\lambda(t)$  is a linear function of  $u_1(t)$  given by

$$\lambda(t) = \lambda_0 - cu_1(t), \quad (5.16)$$

and finally that (f) the gradient  $s'(x)$  is shallow, so that for all practical purposes, the second order moment of the microscopic flux is zero, i.e. that (Erban and Othmer, 2004):

$$j_2(x, t) = \int_{\mathbb{R}} \bar{v} \cdot (p_+(x, z_2, t) - p_-(x, z_2, t)) dz_2 = 0. \quad (5.17)$$

$p_{\pm}(x, z_2, t)$  is the density function of the bacteria at  $(x, t)$  with the internal state  $z_2(x, t) = u_2(t) - s(x)$  that run right (+) or left (-). The above assumptions result to the following closed form for the chemotactic coefficient  $\chi$  (Erban and Othmer, 2004):

$$\chi = \frac{df}{ds} \frac{c\bar{v}^2 t_a}{\lambda_0(1+2\lambda_0 t_a)(1+2\lambda_0 t_e)}. \quad (5.18)$$

## 5.2.2 Data driven PDEs

Before discovering effective PDE laws from microscopic simulations, there is a crucial prerequisite: what are the macroscopic observables whose field evolution laws we want to discover? There are cases for which this knowledge is given: in our case we know we want to derive parabolic evolution laws for the bacterial density field. Yet this knowledge is not always *a priori* given, based on domain knowledge: for the chemotaxis problem itself, we know that in different parameter regimes one needs a hyperbolic (higher order) equation for the density field (Erban and Othmer, 2004). Discovering sets of macroscopic observables in terms of which an evolutionary PDE can be closed is a nontrivial task; often this task can be performed using data mining/manifold learning techniques, and has been named “variable-free computation” (in the sense that the relevant variables are identified through, say, PCA or Diffusion Map

processing of the fine scale simulations (Erban et al., 2007; Arbabi and Kevrekidis, 2021).

In our case (with the chemoattractant field  $s(x)$  fixed, and with Keller-Segel-class equations in mind) we know that we want to identify a parabolic evolutionary PDE in terms of the evolution of a normalized bacterial density  $b(x, t)$  in one spatial dimension. That is, we expect  $\frac{\partial b}{\partial t} = F\left(b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, \dots, s, \frac{ds}{dx}, \frac{d^2 s}{dx^2}, \dots\right)$ . The existence of such a relation between the local bacterial density time derivative and the local values and spatial derivatives of this field, as well as of the chemoattractant field, is our working hypothesis. The question then becomes: how many spatial derivatives of the  $b(x, t), s(x, t)$  fields are required in order to infer a useful data-driven closure? Starting with an assumed highest order of possibly influential spatial derivatives, we resolve this here using a feature selection method. More specifically, we use the automatic relevance determination (ARD) in the Gaussian framework (Rasmussen and Williams, 2005), which has been widely used to identify dominant input features for a certain target output using sensitivity analysis (Liu et al., 2019; Lee et al., 2020; Lee et al., 2021), see Subchapter 2.2.1.1. This feature selection method also provides not only computational cost reduction but also a better physical understanding of the underlying PDEs (*explainability*).

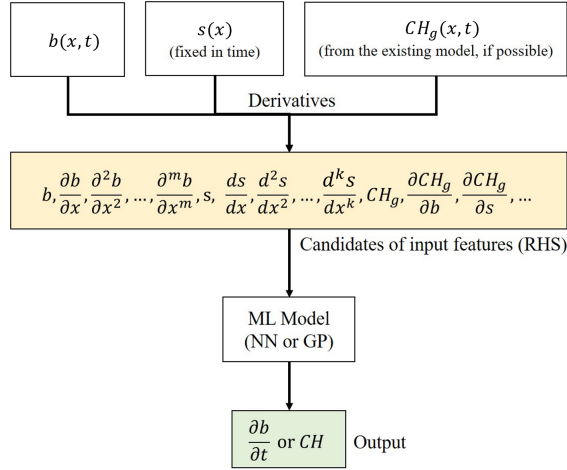
We learn three different types of data-driven PDE models, namely, (1) a black-box, (2) a gray-box, and, (3) PDE RHS's whose closures are learned (based on GPs and ANNs) *as corrections* of the analytically available Keller-Segel PDE closure; how many spatial derivatives are kept in the identified RHS relies on the ARD process above.

In what follows, we describe the basic steps of our numerical scheme.

The first model presented is a black-box model for the local dependence of the time derivative of the bacterial density,  $\frac{\partial b}{\partial t}$ , on the (local) density and its spatial derivatives as well as on the local chemoattractant concentration and its derivatives, (i.e. the chemotactic PDE RHS operator) as:

$$\frac{\partial b}{\partial t} = B\left(b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, \dots, \frac{\partial^m b}{\partial x^m}, s, \frac{ds}{dx}, \frac{d^2 s}{dx^2}, \dots, \frac{d^k s}{dx^k}\right), \quad (5.19)$$

The orders  $m$  and  $k$  are identified here by the ARD feature selection algorithm (see Table 5.4). In several cases, we may know some component of the macroscopic dynamic behavior that comes from intuition, previous studies and/or experiments. For example, one can compute, through simulations and/or the aid of statistical mechanics, the diffusion coefficient of the bacterial density. One can then write a gray-box model which contains a Known Term—such as the diffusion term— $KT$ , and Unknown Terms



**Figure 5.19:** A schematic of the proposed data-driven numerical methodology: given the bacterial density profile in space and time, as well as the chemoattractant profile, (and, for the closure correction approach, the analytically available approximate chemotactic term  $CH_g$  from Eq.(5.14) and their spatial derivatives, we extract possible candidates as input features for learning the PDE right-hand-side (RHS) operator. Then, we train our Machine Learning models (ANN or GP) to approximate  $\frac{\partial b}{\partial t}$  or  $CH$  (the “ground truth” chemotactic term).

(the chemotactic terms)  $UT$ :

$$\frac{\partial b}{\partial t} = UT \left( b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, \dots, \frac{\partial^m b}{\partial x^m}, s, \frac{ds}{dx}, \frac{d^2 s}{dx^2}, \dots, \frac{d^k s}{dx^k} \right) + KT. \quad (5.20)$$

Here, we assume knowledge (or ability to infer) the diffusion term, including the corresponding diffusion coefficient,  $D$ , i.e.,  $KT = D \frac{\partial^2 b}{\partial x^2}$ . In fact, here  $D$  is computed from appropriately designed microscopic random-walk/Brownian motion Monte Carlo simulations *in the absence of a chemical gradient*. Therefore, in our case, the Unknown Terms include just the chemotactic term, i.e.  $\frac{\partial b}{\partial t} = D \frac{\partial^2 b}{\partial x^2} + CH$  or  $CH = \frac{\partial b}{\partial t} - D \frac{\partial^2 b}{\partial x^2}$ .

If some partial information is known (e.g. some of the terms in the RHS of the PDE), we can apply the gray-box approach discussed. However, sometimes, we may have a closed form, analytical, qualitatively good (but less accurate quantitatively) PDE model/closure, capable of describing *in a qualitative manner* similar macro-scale dynamics. For example, we may have a closed-form PDE RHS that is capable of predicting qualitative trends, but is not suitable for our particular experimental setup (e.g., one obtained for different types of chemoattractants, different types of bacteria, etc). This can be thought of as a “low-fidelity” model and it can be used in the same spirit it would be used in a *multifidelity* data fusion context (Lee et al., 2019; Perdikaris et al., 2017). Exploiting such existing approximate models, we

propose a different type of gray-box Machine Learning scheme to **calibrate the model to observation data**, so as to match our specific experimental set-up/our detailed numerical simulations.

We propose four types of data-driven closure corrections to enhance the accuracy of an effective “low-fidelity” PDE. In particular, we used the “generalized PDE” model for the chemotactic dynamics introduced in (Erban and Othmer, 2004) (Eq.5.14) as the low-fidelity reference model we want to correct. The four types of closure correction models are detailed in Table 5.4. First, we used a Machine Learning scheme (based on GP or an ANN), to correct *the chemotactic term*  $CH_g$  of the generalized PDE so as to learn the *true* chemotactic term  $CH$ . It may be that the quantitative closure is a simple, smooth function  $F$  of the analytical approximate closure  $CH_g$  in the general form of:

$$CH(b, \frac{\partial b}{\partial x}, \dots, \frac{\partial^m b}{\partial x^m}, s, \frac{ds}{dx}, \dots, \frac{d^k s}{dk^2}) = F(CH_g(b, \frac{\partial b}{\partial x}, \dots, \frac{\partial^m b}{\partial x^m}, s, \frac{ds}{dx}, \dots, \frac{d^k s}{dk^2})). \quad (5.21)$$

More often than not, this does not suffice, and more information/more variables are necessary for quantitative prediction. Our first approach is to exploit data-driven embedding theories (in the spirit of Whitney/Takens embeddings (Whitney, 1936; Nash, 1966; Takens, 1981; Lee et al., 2019)) to discover corrections from the *known*  $CH_g$  to the *unknown*  $CH$  using the first functional derivatives of  $CH_g$  wrt. its variables (Lee et al., 2019):

$$CH = H(CH_g, \frac{\partial CH_g}{\partial b}, \frac{\partial CH_g}{\partial b_x}, \frac{\partial CH_g}{\partial s}, \frac{\partial CH_g}{\partial s_x}, \frac{\partial CH_g}{\partial s_{xx}}). \quad (5.22)$$

Within the GP framework, we also identified a second “version” of this closure correction, using fewer, dominant such derivatives via ARD analysis as:

$$CH = H(\frac{\partial CH_g}{\partial b}, \frac{\partial CH_g}{\partial b_x}, \frac{\partial CH_g}{\partial s}, \frac{\partial CH_g}{\partial s_x}, \frac{\partial CH_g}{\partial s_{xx}}). \quad (5.23)$$

As a second idea, also conceptually based on embedding theories, we also considered another closure correction approach, in which the equation RHS was not just a function of the approximate  $CH_g$ , but also included additional local inputs (in our first attempt we included the local bacterial density  $b$  and the local chemoattractant  $s$ ) as additional information, so that the corrected closure is a learned function of the form

$$CH = h(CH_g, b, s). \quad (5.24)$$

Finally, we also tried a simple “additive” correction; we learned the bias term between the observed

chemotactic term and the approximated chemotactic term of the generalized PDE in the form:

$$CH - CH_g = H(b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, s, \frac{ds}{dx}, \frac{d^2 s}{dx^2}). \quad (5.25)$$

Data-driven Model	Input Features	Output
Black-box (Full)	$b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, s, \frac{ds}{dx}, \frac{d^2 s}{dx^2}$	$\frac{\partial b}{\partial t}$
Black-box (Reduced)	$b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, s, \frac{ds}{dx}$	$\frac{\partial b}{\partial t}$
Gray-box (Full)	$b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, s, \frac{ds}{dx}, \frac{d^2 s}{dx^2}$	$CH$
Gray-box (Reduced)	$b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, s, \frac{ds}{dx}$	$CH$
Functional Correction (Full)	$CH_g, \frac{\partial CH_g}{\partial b}, \frac{\partial CH_g}{\partial b_x}, \frac{\partial CH_g}{\partial s}, \frac{\partial CH_g}{\partial s_x}, \frac{\partial CH_g}{\partial s_{xx}}$	$CH$
Functional Correction (Reduced)	$\frac{\partial CH_g}{\partial b}, \frac{\partial CH_g}{\partial b_x}, \frac{\partial CH_g}{\partial s}, \frac{\partial CH_g}{\partial s_x}, \frac{\partial CH_g}{\partial s_{xx}}$	$CH$
Correction (No derivatives)	$CH_g, b, s$	$CH$
Additive Correction	$b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, s, \frac{ds}{dx}, \frac{d^2 s}{dx^2}$	$CH - CH_g$

**Table 5.4:** Selected groups of input features and the corresponding predicted quantity (output) for different data-driven PDE law correction approaches. “Reduced” represents models constructed using the (fewer) input features selected via ARD within the Gaussian process framework. We only reduce GP models through ARD; the corresponding NN reduction was not attempted.

### 5.2.3 Results

For our computations, we run a Monte Carlo simulation of  $n = 5000$  bacteria initially located at  $x = 5.5$ , from  $t = 0$  to  $t = 5000s$  with  $dt = 2s$  as reporting horizon, and collect the training data. For training, we collected data from four (fixed in time) different chemo-nutrient concentration profiles, all with a Gaussian distribution  $s(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2})$ : (1)  $\mu = 6, \sigma = 1$ ; (2)  $\mu = 6, \sigma = 1.5$ ; (3)  $\mu = 7, \sigma = 1.5$ ; (4)  $\mu = 7, \sigma = 1.25$ . Specifically, from 5000 individual trajectories of the bacterial motion, we estimated the normalized bacterial density  $b(x, t)$  on a uniform grid from  $x = 3$  to  $x = 9$  with  $dx = 0.05$  at every time step using kernel smoothing (Bowman and Azzalini, 1997) as:

$$b(x, t) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i(t)}{h}\right), \quad (5.26)$$

where  $K(\cdot)$  represents the kernel smoothing function (here, a Gaussian function), and  $h$  is the bandwidth (here, set to  $h = 0.3$ ). For the approximation of the first  $\partial b/\partial x$  and  $\partial b/\partial t$  and the second  $\partial^2 b/\partial x^2$  partial derivatives of the density profile, we used central finite differences. Thus, our data-driven models

are constructed based on six input features ( $b$ ,  $\partial b/\partial x$ ,  $\partial^2 b/\partial x^2$ ,  $s$ ,  $ds/dx$  and  $d^2s/dx^2$ ) that are used for learning the corresponding time derivative  $\partial b/\partial t$ .

Gaussian Process learning was performed in Matlab using a RBF kernel with ARD. For feature selection, the cut-off is  $10^5$ . That is, if the optimal hyperparameter value is higher than  $10^5$ , we eliminate the corresponding input features. Neural Network learning was performed in Python using Tensorflow (Abadi et al., 2015) with two hidden layers with [9, 8, 8] neurons each (Black box, Gray box, Correction model respectively), equipped with a hyperbolic tangent activation function. The Neural Network was trained using an Adam optimizer (Kingma and Ba, 2014) and the training hyperparameters were tuned empirically (Epochs [2, 560, 10, 240, 2, 560], Batches [800, 000, 750, 000, 300, 000], Initial Learning Rate 0.02 with a plateau learning rate scheduler with patience 1, 200 epochs and factor 0.5).

After learning the time evolution operator, we first tested whether the laws identified could reproduce the trajectories from which the training data were collected. For illustration, we performed numerical integration with the data-driven learned PDEs (using both GP and FNN) from  $t = 20$ s to  $t = 4020$ s with  $dt = 2$ s using a 4-th order Runge-Kutta scheme, as well as a commercial package integrator (explicit Runge-Kutta method of order 5(4) (Dormand and Prince, 1980) as implemented by *solve\_ivp* in Python, resulting in a maximum absolute error of  $4 \cdot 10^{-6}$  and the corresponding integrator (and tolerances) in Matlab with maximum absolute error of  $9 \cdot 10^{-4}$ . High spatial frequency Fourier modes of the bacterial density profile were consistently filtered (a procedure analogous to adding hyperviscosity in hydrodynamic models, (Thiem et al., 2021)) The ground truth spatiotemporal evolution of the bacterial density is shown in Figure 5.20(a) and the corresponding relative errors are shown in Figure 5.20(b) while in Figure 5.20(c), we show the reconstructed profile at  $t = 1000$ s for one of our training chemo-nutrient profiles  $s(x) = \frac{1}{\sqrt{2\pi}1.25^2} \exp(-\frac{1}{2} \frac{(x-7)^2}{1.25^2})$ . The performance of our several different data-driven PDE closure corrections was assessed in terms of the relative approximation error between the ground truth density profiles ( $b^{GT}(x, t)$ ) and the profiles ( $b^{DD}(x, t)$ ) resulting from numerical integration of the learned approximate PDE right-hand-sides. This error was defined as  $E_r = 100 \frac{|b^{GT}(x,t) - b^{DD}(x,t)|}{\max b^{GT}(x,t)}$ ; a comparison to the density profile of the full Monte Carlo simulations at  $t = 1000$ s is also provided.

After that, we tested the performance of the data-driven PDE closure corrections with the test data from a new chemoattractant concentration profile (not included in the training data set):  $s(x) = \frac{1}{\sqrt{2\pi}1.35^2} \exp(-\frac{1}{2} \frac{(x-6.5)^2}{1.35^2})$ . The ground truth of the testing case is plotted in Figure 5.21(a). The predicted profile at  $t = 1000$ s and the corresponding relative errors are shown in Figures 5.21(b) and (c), respectively. Table 5.4, summarizes the different data-driven models with respect to (1) Machine Learning

techniques, (2) selected features, and (3) the corresponding predicted quantity.

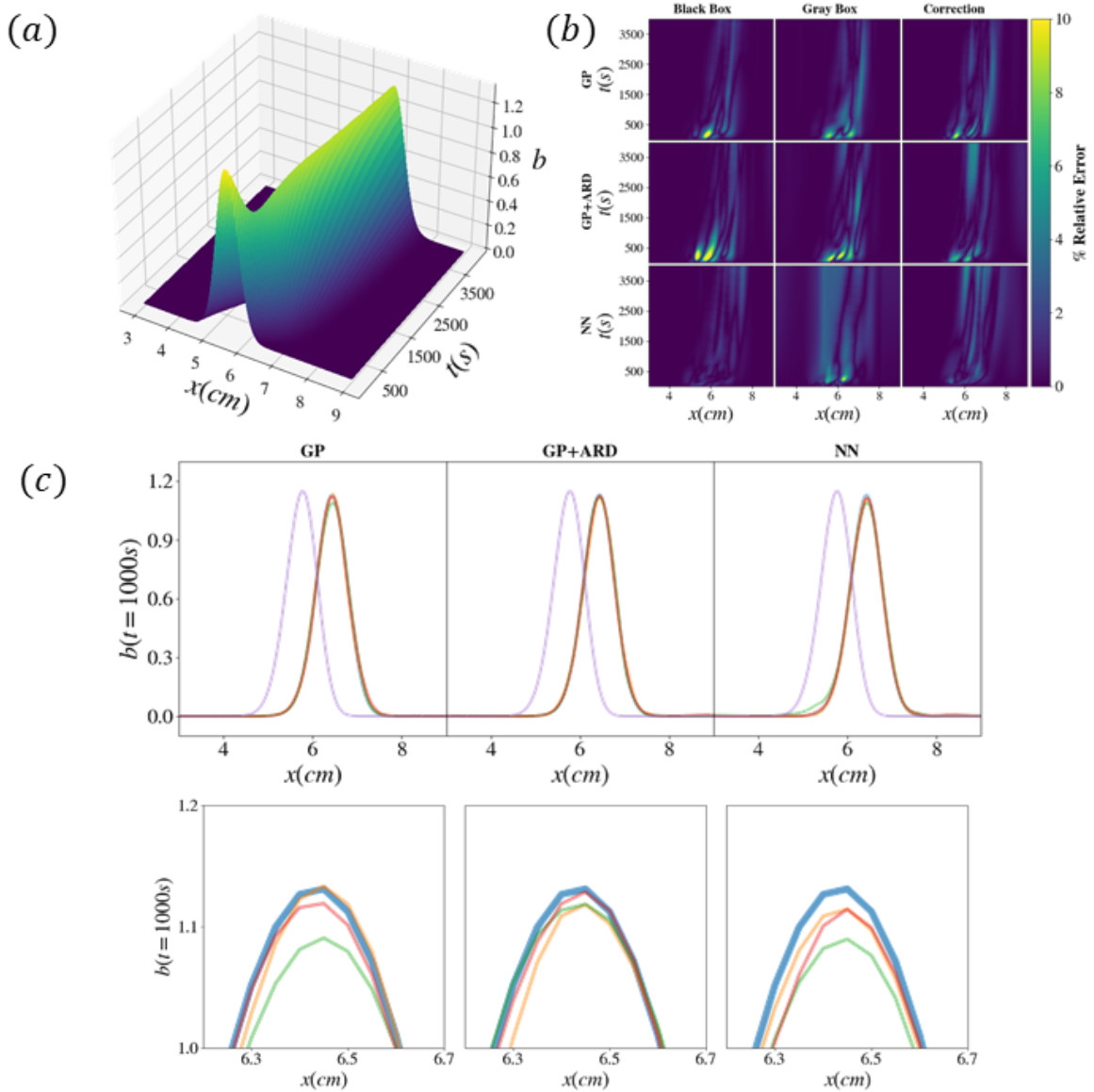
A benefit of reduced feature selection is that the computational cost in the training phase is reduced, while (as shown in Figures 5.20 and 5.21) the predictive accuracy is retained. Regarding the black-box trajectory reconstruction relative error for the training data set, the GP data-driven models never exceed 14% relative error when integrated for a long time (15% for the ARD-reduced GP), while the FNN data-driven models never exceed 4% relative error. For the test data-set, the respective maximum relative errors are 20% and 12% for the GP (full or reduced) and FNN, respectively.

Interestingly, the largest errors observed for the GP are concentrated during the fast, initial transient, while the trajectories become more accurate at later times as they approach steady state. FNNs seem to perform better in capturing this initial transient.

As shown in Figures 5.20 and 5.21, the gray-box models provide comparable, yet slightly improved accuracy, compared to the black-box ones. Specifically, the maximum reconstruction relative errors for the training data set are 9% for the GP (12% for reduced GP) and 10% for the FNN models, while for the testing case these are 12% for the GP (19% for reduced GP) and 10% for the FNN models. These results confirm the capabilities of gray-box models, which combine partial physical knowledge (exact, or even approximate) with data-driven information towards accurate and efficient data-assisted modelling of complex systems.

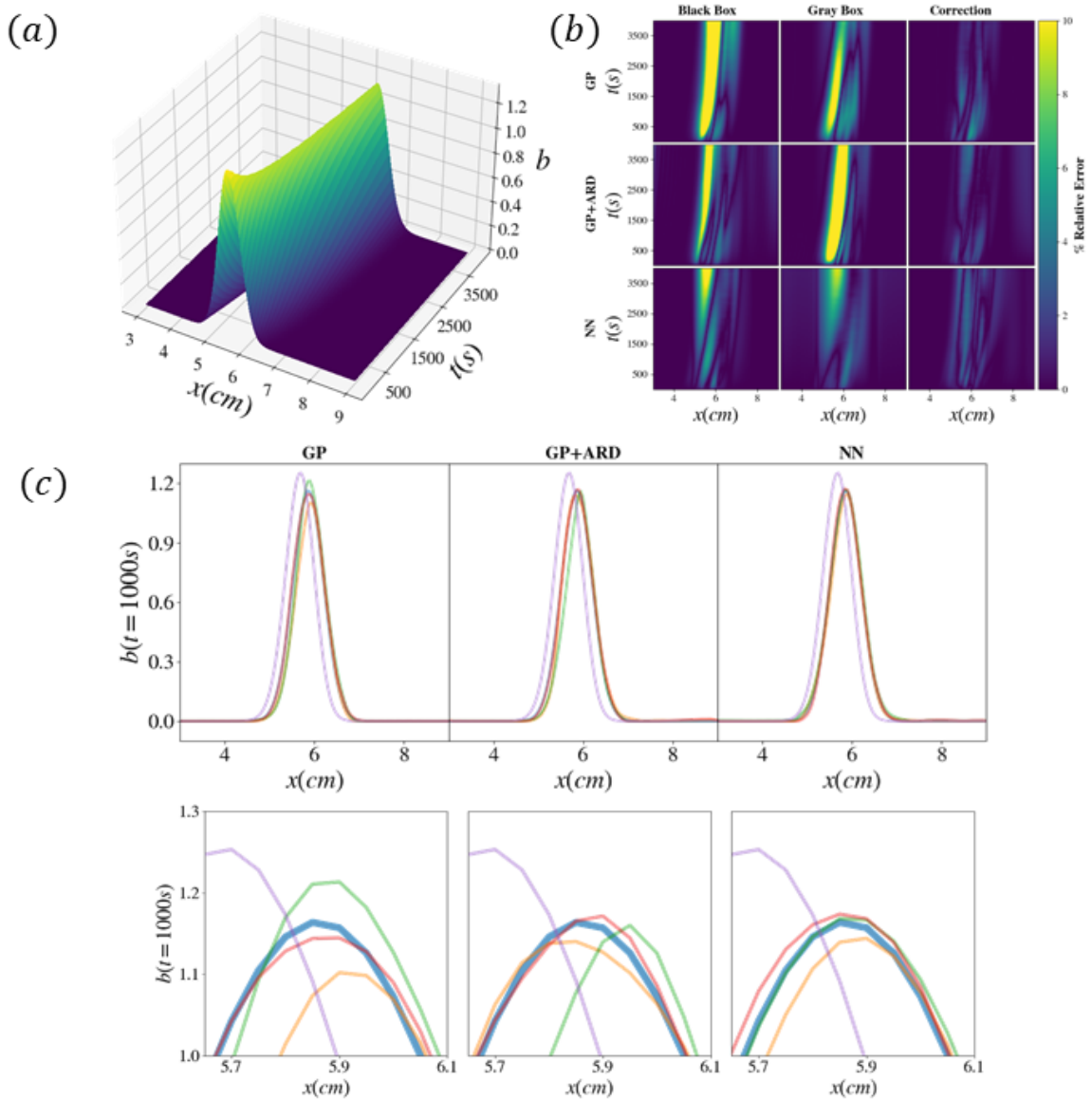
There are, of course no guarantees here for the accurate generalization of the predictions beyond the training data; yet the performance of our models over the test set, and also for chemoattractant profiles not included in the training, appears promising. Our expectation is that the closure correction models, by always making use of the “low-fidelity” (qualitative) information at every time step, may generalize better.

Here, results are presented for two of the four closure correction approaches, i.e. the ones described in Eq.(5.22, 5.23). In particular, the maximum relative errors for the training case were 10% for the GP (8% for reduced GP) and 6% for the FNN models, while for the testing case they were 5% for the GP (4% for reduced GP) and 4% for the FNN models. Thus, all different closure correction approaches (even though the figures for the “additive” and the “no derivatives” corrections are not included for economy of space), provide reasonable accuracy for validation as well as test profiles qualitatively and even quantitatively.



**Figure 5.20:** (a) **Training/reconstruction** case. Ground Truth (GT) evolution for  $\mu = 7, \sigma = 1.25$ : smoothed profiles of bacterial density derived from post-processing agent-based simulations. (b) Quantitative performance of representative data-driven models (DD) trained by Gaussian Process Regression (GP), reduced Gaussian Process Regression with ARD (GP+ARD) or a Neural Network (NN): relative error (%) based on maximum density for black-box model, gray-box model, and correction model (first, second and third rows respectively). (c) Qualitative comparison of profiles of bacterial density at  $t = 1000$ s: PDEs learned through (left) Gaussian process; (middle) Reduced Gaussian processes (with ARD); and (right) Neural Networks. Profiles are colored as follows: blue – ground truth, orange – black box, green – gray box, purple – PDE Eq.[3]. The bottom row is a blowup of the profile’s peak. Note that for the Monte-Carlo simulations the initial state ( $t = 0$ s) is at  $x = 5.5$ cm for all agents, while all PDE simulations (DD models, PDE in 5.14) begin at  $t = 20$ s. No flux boundary conditions were used, consistent with (Erban and Othmer, 2004).





**Figure 5.21:** (a) **Testing** case. Ground Truth (GT) evolution for  $\mu = 6.5, \sigma = 1.35$ : smoothed profiles of bacterial density derived from post-processing agent-based simulations. (b) Quantitative performance of representative data-driven models (DD) trained by Gaussian Process Regression (GP), reduced Gaussian Process Regression with ARD (GP+ARD) or a Neural Network (NN): relative error (%) based on maximum density for black-box model, gray-box model, and correction model (first, second and third rows respectively). (c) Qualitative comparison of profiles of bacterial density at  $t = 1000$ s: PDEs learned through (left) Gaussian process; (middle) Reduced Gaussian processes (with ARD); and (right) Neural Networks. Profiles are colored as follows: blue – ground truth, orange – black box, green – gray box, purple – PDE Eq.[3]. The bottom row is a blowup of the profile’s peak. Note that for the Monte-Carlo simulations the initial state ( $t = 0$ s) is at  $x = 5.5$ cm for all agents, while all PDE simulations (DD models, PDE in 5.14) begin at  $t = 20$ s. No flux boundary conditions were used, consistent with (Erban and Othmer, 2004).

## 5.2.4 Discussion and Conclusions

Here, a Machine Learning framework is presented for the inverse problem in chemotaxis. In particular, we showed how one can learn black-box and gray-box parabolic PDEs for the emergent dynamics of bacterial density evolution (and, importantly, unknown closures and their corrections) directly from high-fidelity microscopic/stochastic simulations. Specifically, we introduced a computational data-driven framework for nonlinear PDE/closure identification and correction; the framework consisted of three progressively more physics-informed processes: (a) learning a black-box PDE (learning the right-hand-side of an coarse-scale PDE including the diffusion term), (b) learning a gray box PDE (an entire unknown closure, with a known Diffusion term), and (c) obtaining closure corrections (providing a correction of an analytically available closure in a low-fidelity, approximate PDE model). Within this framework, we exploited the Automatic Relevance Determination (ARD) algorithm for feature selection, in order to reduce the number of variables on which the learned quantity depends. The overall approach forms a bridge between analytical/mechanistic/physical understanding, and data-driven “black-box” or “gray-box” learning of physical process dynamics, allowing for a synergy between varying types of physical terms/models and data-driven terms/models.

## 5.3 Experiments

The final data source is experiments and, specifically, bacterial motility tracking through fluorescence measurements. This case appears to be the most involved as it comes with the challenge of working with real-world biological data. The work shown in Subchapter 5.1 will facilitate the data-driven PDE model selection and provide with intuition.

### 5.3.1 Experiment description

As shown in (Bhattacharjee et al., 2021), chemotactic motion can be tracked using confocal fluorescence microscopy of *E. coli* populations; thus, we use the data from these prior experiments here. As detailed further in (Bhattacharjee et al., 2021), we 3D-printed a long, cylindrical inoculum of densely-packed cells within a transparent, biocompatible, 3D porous medium comprised of a packing of hydrogel particles swollen in a defined rich liquid medium. Because the individual particles were highly swollen, their internal mesh size was large enough to permit free diffusion of oxygen and nutrient (primarily *L*-serine, which also acts as a chemoattractant), but small enough to prevent bacteria from penetrating into the individual particles; instead, the cells swam through the interstitial pores between hydrogel particles. The cells constitutively express fluorescent protein in their cytoplasm, enabling us to track their motion as they expanded radially outward from the initial cylindrical inoculum in 3D. The fluorescence measurements were collected with spatial resolution  $dr = 2.48 \mu m$  and temporal resolution  $dt = 10 \text{ min}$ , and were then azimuthally averaged, only considering signal from the transverse, not the vertical, direction. The fluorescence signal thereby determined is directly proportional to the density of metabolically-active bacteria, and we will denote it as  $\bar{b}(r, t)$ ; cells are left behind in the wake of the moving chemotactic front, but become immobilized and lose fluorescence as they run out of oxygen and nutrients.

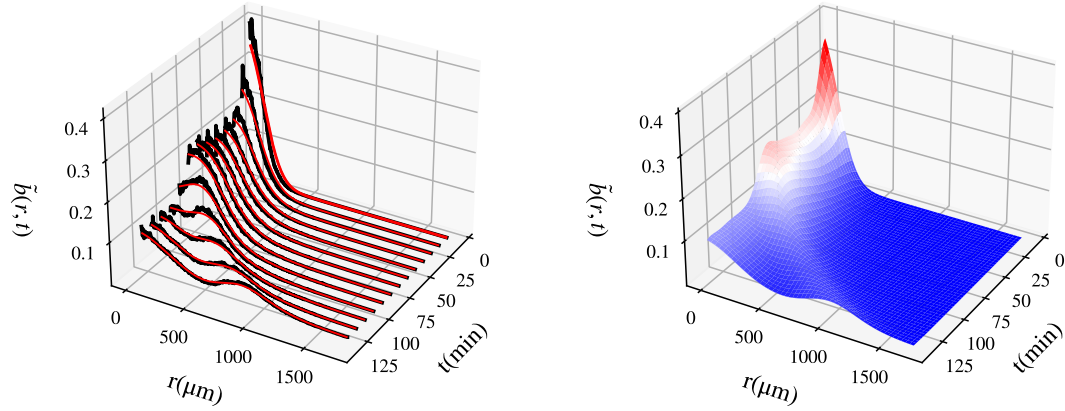
### 5.3.2 Data driven PDE

The training profiles were selected appropriately so that the traveling wave is not too close to the spatial boundaries, and the cylindrical coordinate system remains valid. Profiles were smoothed in space using a local Savitzky-Golay filter and globally using Gaussian Smoothing (Savitzky and Golay, 1964; Getreuer, 2013). The resulting smooth profiles were interpolated in time using Gaussian Radial Basis Functions (Fasshauer, 2007).

We hypothesize that the spatiotemporal behavior of cell density observed in the experiments results from a PDE similar to Eq.5.2. However, we have no measurements of the spatiotemporal evolution of the chemonutrient, therefore, we turn to the methodology described earlier for data-driven models with partial information (for more details, see Subchapter 5.1.2 and specifically, the model with Eq. 5.7).

Interpolation in time allows for well-approximated time derivatives as we can choose data along  $t$  as dense as necessary. In fact, it is possible to estimate second order in time derivatives, which can be used to learn a second order in time continuous-time PDE in lieu of a delay model (Packard et al., 1980), such as that used in 5.7.

$$b_{tt} = f_{NN}^{exp}(b, \nabla \mathbf{b} \cdot \hat{\mathbf{r}}, \Delta b, b_t) \quad (7)$$



**Figure 5.22:** Pre-processing of experimental measurements: (left) Smoothing in space (right) Interpolation in time.

This can be treated as two coupled PDE fields (and integrated as such), by introducing an artificial field  $u(r, t)$ :

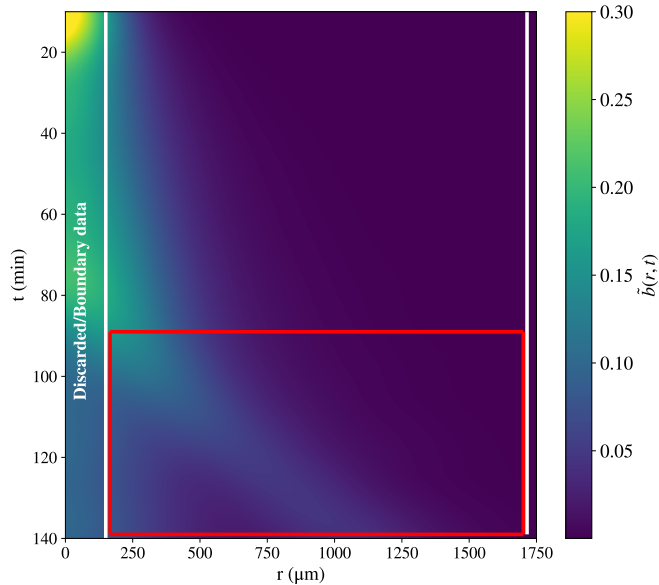
$$u_t = f_{NN}^{exp}(b, \nabla \mathbf{b} \cdot \hat{\mathbf{r}}, \Delta b, u)$$

$$b_t = u$$

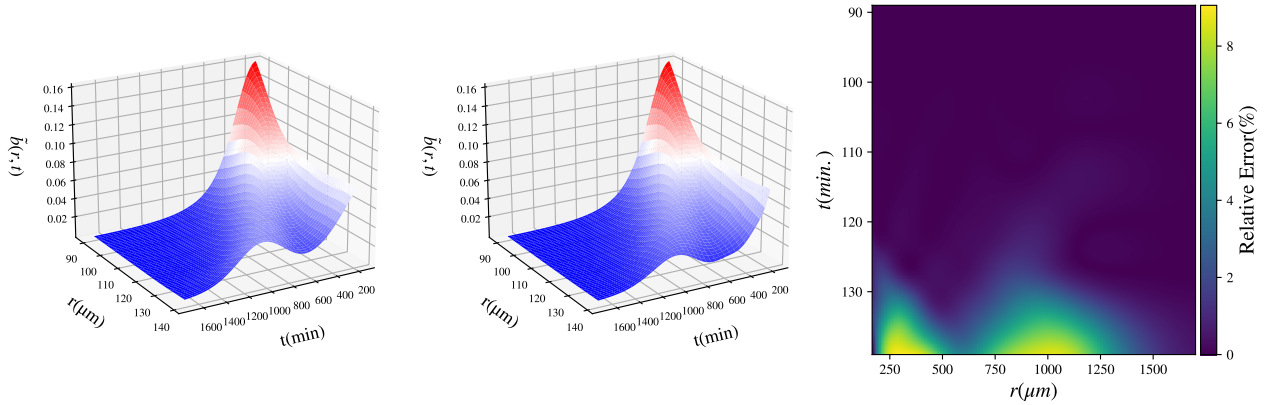
### 5.3.3 Results

Given the nutrient-starved/hypoxic conditions at  $r \approx 0$ , our training data were selected away from the origin. In addition, we assign bilateral boundary corridors which are not used for training or validation of our data-driven model, but only to provide data-informed boundary conditions when integrating the learned PDE.

The learning algorithm consists of a Deep feed-forward Neural Network with 3 hidden layers and 90 neurons per layer. When integrating the data-driven model, SVD filtering was used:  $u_t$  is projected to a lower-dimensional space, defined by the dominant singular vectors of the the  $u_t(r, t)$  data used in training (Halko, Martinsson, and Tropp, 2010). This is a procedure analogous to adding hyperviscosity in hydrodynamic models (Thiem et al., 2021). Here, the eight first singular vectors were used, containing  $> 99\%$  of the variance. The model was validated by integration in the spatiotemporal domain of the formation and propagation of the traveling wave (shown in red in Fig. 5.22).



**Figure 5.23:** Segmentation of the pre-processed data into: boundary corridors/ discarded data, training subset (the complement), subset chosen for reproduction (red rectangle).



**Figure 5.24:** Reproduction of experimental observations using a Data-driven Neural Network for the traveling wave regime: (left) Ground Truth, (middle) Neural-Network PDE integration results, (right) % Relative Error.

### 5.3.4 Discussion and Conclusions

The methodology initially presented in Subchapter 5.1 was applied to data from chemotactic bacterial motility experiments. This demonstrates that the proposed data-driven PDE methodologies can be implemented even when the data at hand are partial, noisy, and/or sparse.

The conclusions from Subchapter 5.3 are aligned with those from Subchapter 5.1: A second order model can indeed capture the dynamics of a data set with partial information. In this case, a deeper Neural Network is required to capture the real-world dynamics from experimental observations. Indeed, the data-driven PDE manages to capture important characteristics of the traveling wave, such as its speed and amplitude. It also manages to capture the dynamics *on the left* of the traveling wave: the bacteria density remains stationary, as in that region the chemonutrient gradient is negligible. Note that, as discussed in (Bhattacharjee et al., 2021), analytical models fail to capture this behavior without non-autonomous correction terms .

# References

- Whitney, Hassler (1936). "Differentiable Manifolds". In: *Annals of Mathematics* 37.3, pp. 645–680. ISSN: 0003486X. URL: <http://www.jstor.org/stable/1968482>.
- Patlak, Clifford S (1953). "A mathematical contribution to the study of orientation of organisms". In: *The bulletin of mathematical biophysics* 15.4, pp. 431–476.
- Savitzky, Abraham. and M. J. E. Golay (1964). "Smoothing and Differentiation of Data by Simplified Least Squares Procedures." In: *Analytical Chemistry* 36.8, pp. 1627–1639. DOI: 10.1021/ac60214a047. eprint: <https://doi.org/10.1021/ac60214a047>. URL: <https://doi.org/10.1021/ac60214a047>.
- Nash, J. (1966). "Analyticity of the Solutions of Implicit Function Problems With Analytic Data". In: *Annals of Mathematics* 84.3, pp. 345–355. ISSN: 0003486X.
- Adler, Julius (1969). "Chemoreceptors in bacteria". In: *Science* 166.3913, pp. 1588–1597.
- Keller, Evelyn F and Lee A Segel (1971). "Model for chemotaxis". In: *Journal of theoretical biology* 30.2, pp. 225–234.
- Berg, Howard C and Douglas A Brown (1972). "Chemotaxis in Escherichia coli analysed by three-dimensional tracking". In: *Nature* 239.5374, pp. 500–504.
- Larsen, Steven H, Robert W Reader, Edward N Kort, Wung-Wai Tso, and Julius Adler (1974). "Change in direction of flagellar rotation is the basis of the chemotactic response in Escherichia coli". In: *Nature* 249.5452, pp. 74–77.
- Maeda, Kayo, Yasuo Imae, Jun-Ichi Shioi, and Fumio Oosawa (1976). "Effect of temperature on motility and chemotaxis of Escherichia coli." In: *Journal of bacteriology* 127.3, pp. 1039–1046.
- Parkinson, John S (1976). "cheA, cheB, and cheC genes of Escherichia coli and their role in chemotaxis." In: *Journal of bacteriology* 126.2, pp. 758–770.
- Dormand, J.R. and P.J. Prince (1980). "A family of embedded Runge-Kutta formulae". In: *Journal of Computational and Applied Mathematics* 6.1, pp. 19–26. ISSN: 0377-0427. DOI: [https://doi.org/10.1016/0771-050X\(80\)90013-3](https://doi.org/10.1016/0771-050X(80)90013-3). URL: <https://www.sciencedirect.com/science/article/pii/0771050X80900133>.
- Packard, N. H., J. P. Crutchfield, J. D. Farmer, and R. S. Shaw (1980). "Geometry from a Time Series". In: *Phys. Rev. Lett.* 45 (9), pp. 712–716. DOI: 10.1103/PhysRevLett.45.712. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.45.712>.
- Parkinson, John S (1980). "Novel mutations affecting a signaling component for chemotaxis of Escherichia coli". In: *Journal of bacteriology* 142.3, pp. 953–961.
- Aeyels, Dirk (1981). "Generic Observability of Differentiable Systems". In: *SIAM Journal on Control and Optimization* 19.5, pp. 595–603. DOI: 10.1137/0319037. eprint: <https://doi.org/10.1137/0319037>. URL: <https://doi.org/10.1137/0319037>.
- Boyd, Alan, Alexandra Krikos, and Melvin Simon (1981). "Sensory transducers of E. coli are encoded by homologous genes". In: *Cell* 26.3, pp. 333–343.

- Gurtin, M. E. (1981). *An Introduction to Continuum Mechanics*. Vol. 158. Mathematics in science and engineering. Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], New York-London. ISBN: 0-12-309750-9.
- Takens, Floris (1981). "Detecting strange attractors in turbulence". In: ed. by David Rand and Lai-Sang Young, pp. 366–381.
- Block, Steven M., Jeffrey E. Segall, and Howard C. Berg (1982). "Impulse responses in bacterial chemotaxis". In: *Cell* 31.1, pp. 215–226. ISSN: 0092-8674. DOI: [https://doi.org/10.1016/0092-8674\(82\)90421-4](https://doi.org/10.1016/0092-8674(82)90421-4). URL: <https://www.sciencedirect.com/science/article/pii/0092867482904214>.
- Block, Steven M., Jeffery E Segall, and Howard C Berg (1983). "Adaptation kinetics in bacterial chemotaxis." In: *Journal of bacteriology* 154.1, pp. 312–323.
- Ishihara, Akira, Jeffrey E Segall, Steven M Block, and Howard C Berg (1983). "Coordination of flagella on filamentous cells of Escherichia coli." In: *Journal of Bacteriology* 155.1, pp. 228–237.
- Segel, Lee A, Albert Goldbeter, Peter N Devreotes, and Barry E Knox (1986). "A mechanism for exact sensory adaptation based on receptor modification". In: *Journal of theoretical biology* 120.2, pp. 151–179.
- Flanders, H. (1989). *Differential forms with applications to the physical sciences*. Second. Dover Books on Advanced Mathematics. Dover Publications, Inc., New York. ISBN: 0-486-66169-5.
- Liu, JD and John S Parkinson (1989). "Role of CheW protein in coupling membrane receptors to the intracellular signaling system of bacterial chemotaxis". In: *Proceedings of the National Academy of Sciences* 86.22, pp. 8703–8707.
- Bryant, R. L., S. S. Chern, R. B. Gardner, H. L. Goldschmidt, and P. A. Griffiths (1991). *Exterior Differential Systems*. Vol. 18. Mathematical Sciences Research Institute Publications. New York, NY: Springer New York. ISBN: 978-1-4613-9716-8 978-1-4613-9714-4. DOI: [10.1007/978-1-4613-9714-4](https://doi.org/10.1007/978-1-4613-9714-4). (Visited on 11/06/2021).
- Sauer, Tim, James A. Yorke, and Martin Casdagli (1991). "Embedology". In: *Journal of Statistical Physics* 65.3, pp. 579–616. ISSN: 1572-9613. DOI: [10.1007/BF01053745](https://doi.org/10.1007/BF01053745). URL: <https://doi.org/10.1007/BF01053745>.
- MacKay, David JC (1992). "Bayesian interpolation". In: *Neural computation* 4.3, pp. 415–447.
- Krischer, K, R Rico-Martinez, IG Kevrekidis, HH Rotermund, G Ertl, and JL Hudson (1993). "Model identification of a spatiotemporally varying catalytic reaction". In: *AIChE Journal* 39.1, pp. 89–98.
- Rico-Martinez, R., J.S. Anderson, and I.G. Kevrekidis (1994). "Continuous-time nonlinear signal processing: A neural network based approach for gray box identification". In: pp. 596–605. URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-0028723099&partnerID=40&md5=88de0ca8a967a54274c853e11a84d03f>.
- Bowman, Adrian W and Adelchi Azzalini (1997). *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*. Vol. 18. OUP Oxford.
- Shampine, Lawrence F. and Mark W. Reichelt (1997). "The MATLAB ODE Suite". In: *SIAM Journal on Scientific Computing* 18.1, pp. 1–22. DOI: [10.1137/S1064827594276424](https://doi.org/10.1137/S1064827594276424). eprint: <https://doi.org/10.1137/S1064827594276424>. URL: <https://doi.org/10.1137/S1064827594276424>.
- Spiro, Peter A, John S Parkinson, and Hans G Othmer (1997). "A model of excitation and adaptation in bacterial chemotaxis". In: *Proceedings of the National Academy of Sciences* 94.14, pp. 7263–7268.
- Stark, J., D.S. Broomhead, M.E. Davies, and J. Huke (1997). "Takens embedding theorems for forced and stochastic systems". In: *Nonlinear Analysis: Theory, Methods & Applications* 30.8,



- pp. 5303–5314. ISSN: 0362-546X. DOI: [https://doi.org/10.1016/S0362-546X\(96\)00149-6](https://doi.org/10.1016/S0362-546X(96)00149-6). URL: <https://www.sciencedirect.com/science/article/pii/S0362546X96001496>.
- Zhu, Ciyou, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal (1997). “Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-Scale Bound-Constrained Optimization”. In: *ACM Trans. Math. Softw.* 23.4, 550–560. ISSN: 0098-3500. DOI: [10.1145/279232.279236](https://doi.org/10.1145/279232.279236). URL: <https://doi.org/10.1145/279232.279236>.
- González-García, R., R. Rico-Martínez, and I.G. Kevrekidis (1998). “Identification of distributed parameter systems: A neural net based approach”. In: *Computers & Chemical Engineering* 22, S965–S968. ISSN: 0098-1354. DOI: [https://doi.org/10.1016/S0098-1354\(98\)00191-4](https://doi.org/10.1016/S0098-1354(98)00191-4). URL: <https://www.sciencedirect.com/science/article/pii/S0098135498001914>.
- Multiphysics, COMSOL (1998). “Introduction to COMSOL multiphysics®”. In: *COMSOL Multiphysics, Burlington, MA, accessed Feb 9*, p. 2018.
- Othmer, Hans G and Pauline Schaap (1998). “Oscillatory cAMP signaling in the development of *Dictyostelium discoideum*”. In: *Comments on Theoretical Biology* 5, pp. 175–282.
- Stark, J. (1999). “Delay Embeddings for Forced Systems. I. Deterministic Forcing”. In: *Journal of Nonlinear Science* 9.3, pp. 255–332. ISSN: 1432-1467. DOI: [10.1007/s003329900072](https://doi.org/10.1007/s003329900072). URL: <https://doi.org/10.1007/s003329900072>.
- Cluzel, Philippe, Michael Surette, and Stanislas Leibler (2000). “An ultrasensitive bacterial motor revealed by monitoring signaling proteins in single cells”. In: *Science* 287.5458, pp. 1652–1655.
- Alexandridis, AP, CI Siettos, HK Sarimveis, AG Boudouvis, and GV Bafas (2002). “Modelling of nonlinear process dynamics using Kohonen’s neural networks, fuzzy systems and Chebyshev series”. In: *Computers & Chemical Engineering* 26.4-5, pp. 479–486.
- Heit, Bryan, Samantha Tavener, Eko Raharjo, and Paul Kubes (2002). “An intracellular signaling hierarchy determines direction of migration in opposing chemotactic gradients”. In: *The Journal of cell biology* 159.1, pp. 91–102.
- Siettos, Constantinos I and George V Bafas (2002). “Semiglobal stabilization of nonlinear systems using fuzzy control and singular perturbation methods”. In: *Fuzzy Sets and Systems* 129.3, pp. 275–294.
- Siettos, Constantinos I, George V Bafas, and Andreas G Boudouvis (2002). “Truncated Chebyshev series approximation of fuzzy systems for control and nonlinear system identification”. In: *Fuzzy sets and systems* 126.1, pp. 89–104.
- Sourjik, Victor and Howard C. Berg (2002). “Receptor sensitivity in bacterial chemotaxis”. In: *Proceedings of the National Academy of Sciences* 99.1, pp. 123–127. ISSN: 0027-8424. DOI: [10.1073/pnas.011589998](https://doi.org/10.1073/pnas.011589998). eprint: <https://www.pnas.org/content/99/1/123.full.pdf>. URL: <https://www.pnas.org/content/99/1/123>.
- Li, Ju, Panayotis G. Kevrekidis, C. William Gear, and Ioannis G. Kevrekidis (2003). “Deciding the Nature of the Coarse Equation through Microscopic Simulations: The Baby-Bathwater Scheme”. In: *Multiscale Modeling & Simulation* 1.3, pp. 391–407. DOI: [10.1137/S1540345902419161](https://doi.org/10.1137/S1540345902419161). eprint: <https://doi.org/10.1137/S1540345902419161>. URL: <https://doi.org/10.1137/S1540345902419161>.
- Stark, J., D. S. Broomhead, M. E. Davies, and J. Huke (2003). “Delay Embeddings for Forced Systems.II. Stochastic Forcing”. In: *Journal of Nonlinear Science* 13.6, pp. 519–577. ISSN: 1432-1467. DOI: [10.1007/s00332-003-0534-4](https://doi.org/10.1007/s00332-003-0534-4). URL: <https://doi.org/10.1007/s00332-003-0534-4>.
- Erban, Radek and Hans G Othmer (2004). “From individual to collective behavior in bacterial chemotaxis”. In: *SIAM Journal on Applied Mathematics* 65.2, pp. 361–391.

- Emonet, Thierry, Charles M Macal, Michael J North, Charles E Wickersham, and Philippe Cluzel (2005). "AgentCell: a digital single-cell assay for bacterial chemotaxis". In: *Bioinformatics* 21.11, pp. 2714–2721.
- Rasmussen, Carl Edward and Christopher K. I. Williams (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press. ISBN: 026218253X.
- Setayeshgar, Simas, C William Gear, Hans G Othmer, and Ioannis G Kevrekidis (2005). "Application of coarse integration to bacterial chemotaxis". In: *Multiscale Modeling & Simulation* 4.1, pp. 307–327.
- Wu, Mingming, John W Roberts, Sue Kim, Donald L Koch, and Matthew P DeLisa (2006). "Collective bacterial dynamics revealed using a three-dimensional population-scale defocused particle tracking technique". In: *Applied and environmental microbiology* 72.7, pp. 4987–4994.
- Erban, Radek, Thomas A Frewen, Xiao Wang, Timothy C Elston, Ronald Coifman, Boaz Nadler, and Ioannis G Kevrekidis (2007). "Variable-free exploration of stochastic models: a gene regulatory network example". In: *The Journal of chemical physics* 126.15, 04B618.
- Fasshauer, Gregory E (2007). *Meshfree Approximation Methods with Matlab*. WORLD SCIENTIFIC. DOI: 10.1142/6437. eprint: <https://www.worldscientific.com/doi/pdf/10.1142/6437>. URL: <https://www.worldscientific.com/doi/abs/10.1142/6437>.
- Chavanis, Pierre-Henri (2008). "Nonlinear mean field Fokker-Planck equations. Application to the chemotaxis of biological populations". In: *The European Physical Journal B* 62.2, pp. 179–208.
- Tindall, Marcus J, Philip K Maini, Steven L Porter, and Judith P Armitage (2008a). "Overview of mathematical approaches used to model bacterial chemotaxis II: bacterial populations". In: *Bulletin of mathematical biology* 70.6, p. 1570.
- Tindall, Marcus J, SL Porter, PK Maini, G Gaglia, and Judith P Armitage (2008b). "Overview of mathematical approaches used to model bacterial chemotaxis I: the single cell". In: *Bulletin of mathematical biology* 70.6, pp. 1525–1569.
- Lee, J. (2009). *Manifolds and Differential Geometry*. en. Vol. 107. Graduate Studies in Mathematics. Providence, Rhode Island: American Mathematical Society. ISBN: 978-0-8218-4815-9 978-1-4704-1170-1. DOI: 10.1090/gsm/107. URL: <http://www.ams.org/gsm/107> (visited on 04/19/2022).
- Bellomo, Nicola, Abdelghani Bellouquid, Juan Nieto, and Juan Soler (2010). "Multiscale biological tissue models and flux-limited chemotaxis for multicellular growing systems". In: *Mathematical Models and Methods in Applied Sciences* 20.07, pp. 1179–1207.
- Halko, Nathan, Per-Gunnar Martinsson, and Joel A. Tropp (2010). *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*. arXiv: 0909.4061 [math.NA].
- Sarkar, Mayukh K, Koushik Paul, and David Blair (2010). "Chemotaxis signaling protein CheY binds to the rotor protein FliN to control the direction of flagellar rotation in Escherichia coli". In: *Proceedings of the National Academy of Sciences* 107.20, pp. 9370–9375.
- Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.
- Wilson, S. O. (2011). "Differential Forms, Fluids, and Finite Models". en. In: *Proceedings of the American Mathematical Society* 139.7, pp. 2597–2604. ISSN: 0002-9939, 1088-6826. DOI: 10.1090/S0002-9939-2011-11003-7. URL: <http://www.ams.org/jourcgi/jour-getitem?pii=S0002-9939-2011-11003-7> (visited on 11/19/2021).

- Coburn, Luke, Luca Cerone, Colin Torney, Iain D Couzin, and Zoltan Neufeld (2013). “Tactile interactions lead to coherent motion and enhanced chemotaxis of migrating cells”. In: *Physical biology* 10.4, p. 046002.
- Franz, Benjamin and Radek Erban (2013). “Hybrid modelling of individual movement and collective behaviour”. In: *Dispersal, individual movement and spatial ecology*. Springer, pp. 129–157.
- Getreuer, Pascal (2013). “A Survey of Gaussian Convolution Algorithms”. In: *Image Processing On Line* 3, pp. 286–310.
- Othmer, Hans G, Xiangrong Xin, and Chuan Xue (2013). “Excitation and adaptation in bacteria—a model signal transduction system that controls taxis and spatial pattern formation”. In: *International journal of molecular sciences* 14.5, pp. 9205–9248.
- Rousset, Mathias and Giovanni Samaey (2013). “Simulating individual-based models of bacterial chemotaxis with asymptotic variance reduction”. In: *Mathematical Models and Methods in Applied Sciences* 23.12, pp. 2155–2191.
- Kingma, Diederik P. and Jimmy Ba (2014). *Adam: A Method for Stochastic Optimization*. DOI: 10.48550/ARXIV.1412.6980. URL: <https://arxiv.org/abs/1412.6980>.
- Siettos, Constantinos (2014). “Coarse-grained computational stability analysis and acceleration of the collective dynamics of a Monte Carlo simulation of bacterial locomotion”. In: *Applied Mathematics and Computation* 232, pp. 836–847.
- Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. URL: <https://www.tensorflow.org/>.
- Wang, Bo and Tao Chen (2015). “Gaussian process regression with multiple response variables”. In: *Chemometrics and Intelligent Laboratory Systems* 142, pp. 159–165. ISSN: 0169-7439. DOI: <https://doi.org/10.1016/j.chemolab.2015.01.016>. URL: <https://www.sciencedirect.com/science/article/pii/S0169743915000180>.
- Brunton, Steven L., Joshua L. Proctor, and J. Nathan Kutz (2016). “Discovering governing equations from data by sparse identification of nonlinear dynamical systems”. In: *Proceedings of the National Academy of Sciences* 113.15, pp. 3932–3937. DOI: 10.1073/pnas.1517384113. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.1517384113>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.1517384113>.
- Altaf, M. U., E. S. Titi, T. Gebrael, O. M. Knio, L. Zhao, M. F. McCabe, and I. Hoteit (2017). “Downscaling the 2D Bénard convection equations using continuous data assimilation”. In: *Computational Geosciences* 21.3, pp. 393–410. ISSN: 1573-1499. DOI: 10.1007/s10596-017-9619-2. URL: <https://doi.org/10.1007/s10596-017-9619-2>.
- Farhat, Aseel, Evelyn Lunasin, and Edriss S. Titi (2017). “Continuous Data Assimilation for a 2D Bénard Convection System Through Horizontal Velocity Measurements Alone”. In: *Journal of Nonlinear Science* 27.3, pp. 1065–1087. ISSN: 1432-1467. DOI: 10.1007/s00332-017-9360-y. URL: <https://doi.org/10.1007/s00332-017-9360-y>.
- Perdikaris, P., M. Raissi, A. Damianou, N. D. Lawrence, and G. E. Karniadakis (2017). “Non-linear information fusion algorithms for data-efficient multi-fidelity modelling”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473.2198, p. 20160751. DOI: 10.1098/rspa.2016.0751. eprint: <https://royalsocietypublishing>.

- [org/doi/pdf/10.1098/rspa.2016.0751](https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.2016.0751). URL: <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2016.0751>.
- Sandhu, Rimple, Chris Pettit, Mohammad Khalil, Dominique Poirel, and Abhijit Sarkar (2017). “Bayesian model selection using automatic relevance determination for nonlinear dynamical systems”. In: *Computer Methods in Applied Mechanics and Engineering* 320, pp. 237–260.
- Yasuda, Shugo (2017). “Monte Carlo simulation for kinetic chemotaxis model: An application to the traveling population wave”. In: *Journal of Computational Physics* 330, pp. 1022–1042.
- Vlachas, Pantelis R., Wonmin Byeon, Zhong Y. Wan, Themistoklis P. Sapsis, and Petros Koumoutsakos (2018). “Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 474.2213, p. 20170844. DOI: 10.1098/rspa.2017.0844. eprint: <https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.2017.0844>. URL: <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2017.0844>.
- Bar-Sinai, Yohai, Stephan Hoyer, Jason Hickey, and Michael P. Brenner (2019). “Learning data-driven discretizations for partial differential equations”. In: *Proceedings of the National Academy of Sciences* 116.31, pp. 15344–15349. DOI: 10.1073/pnas.1814058116. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.1814058116>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.1814058116>.
- Bertalan, Tom, Felix Dietrich, Igor Mezić, and Ioannis G Kevrekidis (2019). “On learning Hamiltonian systems from data”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.12, p. 121107.
- Chen, Ricky T. Q., Yulia Rubanova, Jesse Bettencourt, and David Duvenaud (2019). *Neural Ordinary Differential Equations*. arXiv: 1806.07366 [cs.LG].
- Ghorbani, Amirata, Abubakar Abid, and James Zou (2019). “Interpretation of Neural Networks Is Fragile”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01, pp. 3681–3688. DOI: 10.1609/aaai.v33i01.33013681. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/4252>.
- Lee, Seungjoon, Felix Dietrich, George E Karniadakis, and Ioannis G Kevrekidis (2019). “Linking Gaussian process regression with data-driven manifold embeddings for nonlinear data fusion”. In: *Interface Focus* 9.3, p. 20180083.
- Liu, Kailong, Yi Li, Xiaosong Hu, Mattin Lucu, and Widanalage Dhammika Widanage (2019). “Gaussian process regression with automatic relevance determination kernel for calendar aging prediction of lithium-ion batteries”. In: *IEEE Transactions on Industrial Informatics* 16.6, pp. 3767–3777.
- Painter, Kevin J (2019). “Mathematical models for chemotaxis and their applications in self-organisation phenomena”. In: *Journal of theoretical biology* 481, pp. 162–182.
- Raissi, Maziar, Paris Perdikaris, and George E Karniadakis (2019). “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational Physics* 378, pp. 686–707.
- Kemeth, Felix P, Tom Bertalan, Thomas Thiem, Felix Dietrich, Sung Joon Moon, Carlo R Laing, and Ioannis G Kevrekidis (2020). “Learning emergent PDEs in a learned emergent space”. In: *arXiv preprint arXiv:2012.12738*.
- Lee, Seungjoon, Mahdi Kooshkbaghi, Konstantinos Spiliotis, Constantinos I Siettos, and Ioannis G Kevrekidis (2020). “Coarse-scale PDEs from fine-scale observations via machine learning”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 30.1, p. 013141.
- Luk, Kevin and Roger Grosse (2020). *A Coordinate-Free Construction of Scalable Natural Gradient*. URL: <https://openreview.net/forum?id=H1lBYCEFD8>.

- Sitzmann, V., J. Martel, A. Bergman, D. Lindell, and G. Wetzstein (2020). “Implicit Neural Representations with Periodic Activation Functions”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 7462–7473. URL: <https://proceedings.neurips.cc/paper/2020/hash/53c04118df112c13a8c34b38343b9c10-Abstract.html> (visited on 04/19/2022).
- Vlachas, Pantelis R, Jaideep Pathak, Brian R Hunt, Themistoklis P Sapsis, Michelle Girvan, Edward Ott, and Petros Koumoutsakos (2020). “Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics”. In: *Neural Networks* 126, pp. 191–217.
- Arbabi, Hassan and Ioannis G Kevrekidis (2021). “Particles to partial differential equations parsimoniously?” In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.3, p. 033137.
- Bhattacharjee, Tapomoy, Daniel B. Amchin, Jenna A. Ott, Felix Kratz, and Sujit S. Datta (2021). “Chemotactic migration of bacteria in porous media”. In: *Biophysical Journal* 120.16, pp. 3483–3497. ISSN: 0006-3495. DOI: 10.1016/j.bpj.2021.05.012. URL: <https://doi.org/10.1016/j.bpj.2021.05.012>.
- Bronstein, M. M., J. Bruna, T. Cohen, and P. Veličković (2021). “Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges”. In: *arXiv:2104.13478 [cs, stat]*. URL: <http://arxiv.org/abs/2104.13478> (visited on 07/23/2021).
- Chen, Yifan, Bamdad Hosseini, Houman Owhadi, and Andrew M Stuart (2021). “Solving and learning nonlinear pdes with gaussian processes”. In: *Journal of Computational Physics* 447, p. 110668.
- Jenner, E. and M. Weiler (2021). “Steerable Partial Differential Operators for Equivariant Neural Networks”. en. In: URL: <https://openreview.net/forum?id=N9W24a4zU> (visited on 04/20/2022).
- Karniadakis, George Em, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang (2021). “Physics-informed machine learning”. In: *Nature Reviews Physics* 3.6, pp. 422–440.
- Lee, Kibaek, Alberto M Hernández, D Scott Stewart, and Seungjoon Lee (2021). “Data-driven blended equations of state for condensed-phase explosives”. In: *Combustion Theory and Modelling*, pp. 1–23.
- Thiem, Thomas N., Felix P. Kemeth, Tom Bertalan, Carlo R. Laing, and Ioannis G. Kevrekidis (2021). “Global and local reduced models for interacting, heterogeneous agents”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.7, p. 073139. DOI: 10.1063/5.0055840. eprint: <https://doi.org/10.1063/5.0055840>. URL: <https://doi.org/10.1063/5.0055840>.
- Weiler, Maurice, Patrick Forré, Erik Verlinde, and Max Welling (2021). *Coordinate Independent Convolutional Networks – Isometry and Gauge Equivariant Convolutions on Riemannian Manifolds*. arXiv: 2106.06020 [cs.LG].
- Bellomo, N, N Outada, J Soler, Y Tao, and M Winkler (2022). “Chemotaxis and cross-diffusion models in complex environments: Models and analytic problems toward a multiscale vision”. In: *Mathematical Models and Methods in Applied Sciences*, pp. 1–80.
- Galaris, Evangelos, Gianluca Fabiani, Ioannis Gallos, Ioannis Kevrekidis, and Constantinos Siettos (2022). “Numerical Bifurcation Analysis of PDEs From Lattice Boltzmann Model Simulations: a Parsimonious Machine Learning Approach”. In: *Journal of Scientific Computing* 92.2, p. 34. ISSN: 1573-7691. DOI: 10.1007/s10915-022-01883-y. URL: <https://doi.org/10.1007/s10915-022-01883-y>.

- Lee, Seungjoon, Yorgos M. Psarellis, Constantinos I. Siettos, and Ioannis G. Kevrekidis (2022). *Learning black- and gray-box chemotactic PDEs/closures from agent based Monte Carlo simulation data*. DOI: 10.48550/ARXIV.2205.13545. URL: <https://arxiv.org/abs/2205.13545>.
- Rao, Chengping, Pu Ren, Yang Liu, and Hao Sun (2022). *Discovering Nonlinear PDEs from Scarce Data with Physics-encoded Learning*. arXiv: 2201.12354 [cs.LG].
- Vlachas, Pantelis R., Georgios Arampatzis, Caroline Uhler, and Petros Koumoutsakos (2022). "Multiscale simulations of complex systems by learning their effective dynamics". In: *Nature Machine Intelligence* 4.4, pp. 359–366. ISSN: 2522-5839. DOI: 10.1038/s42256-022-00464-w. URL: <https://doi.org/10.1038/s42256-022-00464-w>.
- LeCun, Yann, Yoshua Bengio, et al. (n.d.). "Convolutional networks for images, speech, and time series". In: ().

## Chapter 6

# Discussion and Conclusion

This dissertation begins with a broad premise: data-driven algorithms can facilitate traditional modeling of complex systems in Chemistry and Biology, when used *synergistically* with it. Three systems are employed in support of this statement, presented in the order of increasing involvement of Machine Learning:

- (i) **Discovering Limits of Entrainment for Circadian Neuronal Networks** (Chapter 3): Here, bifurcation studies of a high-dimensional computational neuroscience model are performed, by simply probing integration trajectories. The matrix free approach presented, can be applied to any black-box integrator model. Diffusion Maps allow the discovery of a latent heterogeneity space, answering the nontrivial question “How *really* different are all neurons in a network?”.
- (ii) **Optimizing Reactors under Dynamic Operation** (Chapter 4): Here, a detailed analysis of the dynamics of periodically forced reactors is performed. Employing suitable algorithms from numerical methods and nonlinear dynamics we accelerate the location of steady states and efficiently explore the parameter space thus understanding the added degrees of freedom. Using Bayesian methods we enable an active learning modality, constructing surrogate models for the solution and objective function manifolds.
- (iii) **Learning Chemotactic PDEs with Machine Learning** (Chapter 5): Here, macroscale chemotactic PDEs are identified from a variety of data sets: high-fidelity macroscale data from deterministic PDE simulations, high-fidelity microscale data from stochastic agent-based simulations and real-world experimental data. It is demonstrated how we can incorporate *a priori* Biology/Chemistry

knowledge to Machine Learning algorithms. This can be done at various levels: knowledge of relevant operators, knowledge of some PDE terms or knowledge of entire approximate PDEs.

Hopefully, this thesis stands as evidence that this synergy is a “two way street”: It is not only Machine Learning that assists traditional modeling in Biology/Chemistry, e.g. through universal approximation, computational efficiency and model versatility. It is also the fundamental knowledge of Biology/Chemistry that helps construct generalizable, explainable and parsimonious models (acting in a sense like Occam’s razor (Blumer et al., 1987)).

Future directions are abundant, especially in Biology. Multiscale system identification (see, for example Fig.1.4) can lead to a better understanding of pathological phenotypes and connect them with early spatiotemporal predictors (observers) in the single-cell scale (Alber et al., 2019; Ji et al., 2021; Chabiniok et al., 2016; Peng et al., 2021). Exploring the (possibly reduced (Pozharskiy et al., 2020)) parameter space of complex systems is also of major importance: Discovering the effective degrees of freedom (Angerer et al., 2015), predicting the system’s response to different inputs (Shipp et al., 2002), “foracasting” qualitative changes of behavior (bifurcations (Ferrell, 2012)) or rare events (Wan et al., 2018), and driving it to optimal responses by tuning its degrees of freedom (Villaverde and Banga, 2014; Radivojević et al., 2020; Mason et al., 2021; Padmanabhan, Meskin, and Haddad, 2017).

In conclusion, there is great value in the synergistic use of Machine Learning with traditional modeling. Given the assumptions, limitations and benefits of each, this is a promising research direction for both scientists/engineers with field-specific expertise and Machine Learning practitioners.



# References

- Blumer, Anselm, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth (1987). "Occam's Razor". In: *Information Processing Letters* 24.6, pp. 377–380. ISSN: 0020-0190. DOI: [https://doi.org/10.1016/0020-0190\(87\)90114-1](https://doi.org/10.1016/0020-0190(87)90114-1). URL: <https://www.sciencedirect.com/science/article/pii/0020019087901141>.
- Shipp, Margaret A., Ken N. Ross, Pablo Tamayo, Andrew P. Weng, Jeffery L. Kutok, Ricardo C.T. Aguiar, Michelle Gaasenbeek, Michael Angelo, Michael Reich, Geraldine S. Pinkus, Tane S. Ray, Margaret A. Koval, Kim W. Last, Andrew Norton, T. Andrew Lister, Jill Mesirov, Donna S. Neuberg, Eric S. Lander, Jon C. Aster, and Todd R. Golub (2002). "Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning". In: *Nature Medicine* 8.1, pp. 68–74. ISSN: 1546-170X. DOI: [10.1038/nm0102-68](https://doi.org/10.1038/nm0102-68). URL: <https://doi.org/10.1038/nm0102-68>.
- Ferrell, James E. (2012). "Bistability, Bifurcations, and Waddington's Epigenetic Landscape". In: *Current Biology* 22.11, R458–R466. ISSN: 0960-9822. DOI: <https://doi.org/10.1016/j.cub.2012.03.045>. URL: <https://www.sciencedirect.com/science/article/pii/S0960982212003326>.
- Villaverde, Alejandro F. and Julio R. Banga (2014). "Reverse engineering and identification in systems biology: strategies, perspectives and challenges". In: *Journal of The Royal Society Interface* 11.91, p. 20130505. DOI: [10.1098/rsif.2013.0505](https://royalsocietypublishing.org/doi/pdf/10.1098/rsif.2013.0505). eprint: <https://royalsocietypublishing.org/doi/abs/10.1098/rsif.2013.0505>. URL: <https://royalsocietypublishing.org/doi/abs/10.1098/rsif.2013.0505>.
- Angerer, Philipp, Laleh Haghverdi, Maren Büttner, Fabian J. Theis, Carsten Marr, and Florian Büttner (2015). "destiny: diffusion maps for large-scale single-cell data in R". In: *Bioinformatics* 32.8, pp. 1241–1243. ISSN: 1367-4803. DOI: [10.1093/bioinformatics/btv715](https://academic.oup.com/bioinformatics/article-pdf/32/8/1241/16920896/btv715.pdf). eprint: <https://academic.oup.com/bioinformatics/article-pdf/32/8/1241/16920896/btv715.pdf>. URL: <https://doi.org/10.1093/bioinformatics/btv715>.
- Chabiniok, Radomir, Vicky Y. Wang, Myriantithi Hadjicharalambous, Liya Asner, Jack Lee, Maxime Sermesant, Ellen Kuhl, Alistair A. Young, Philippe Moireau, Martyn P. Nash, Dominique Chapelle, and David A. Nordsletten (2016). "Multiphysics and multiscale modelling, data&#x2013;model fusion and integration of organ physiology in the clinic: ventricular cardiac mechanics". In: *Interface Focus* 6.2, p. 20150083. DOI: [10.1098/rsfs.2015.0083](https://royalsocietypublishing.org/doi/pdf/10.1098/rsfs.2015.0083). eprint: <https://royalsocietypublishing.org/doi/pdf/10.1098/rsfs.2015.0083>. URL: <https://royalsocietypublishing.org/doi/abs/10.1098/rsfs.2015.0083>.
- Padmanabhan, Regina, Nader Meskin, and Wassim M. Haddad (2017). "Reinforcement learning-based control of drug dosing for cancer chemotherapy treatment". In: *Mathematical Biosciences* 293, pp. 11–20. ISSN: 0025-5564. DOI: <https://doi.org/10.1016/j.mbs.2017.08.004>. URL: <https://www.sciencedirect.com/science/article/pii/S0025556417304327>.

- Wan, Zhong Yi, Pantelis Vlachas, Petros Koumoutsakos, and Themistoklis Sapsis (2018). “Data-assisted reduced-order modeling of extreme events in complex dynamical systems”. In: *PLOS ONE* 13.5, pp. 1–22. DOI: [10.1371/journal.pone.0197704](https://doi.org/10.1371/journal.pone.0197704). URL: <https://doi.org/10.1371/journal.pone.0197704>.
- Alber, Mark, Adrian Buganza Tepole, William R. Cannon, Suvaranu De, Salvador Dura-Bernal, Krishna Garikipati, George Karniadakis, William W. Lytton, Paris Perdikaris, Linda Petzold, and Ellen Kuhl (2019). “Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences”. In: *npj Digital Medicine* 2.1, p. 115. ISSN: 2398-6352. DOI: [10.1038/s41746-019-0193-y](https://doi.org/10.1038/s41746-019-0193-y). URL: <https://doi.org/10.1038/s41746-019-0193-y>.
- Pozharskiy, Dmitry, Noah J. Wichrowski, Andrew B. Duncan, Grigorios A. Pavliotis, and Ioannis G. Kevrekidis (2020). *Manifold Learning for Accelerating Coarse-Grained Optimization*. DOI: [10.48550/ARXIV.2001.03518](https://arxiv.org/abs/2001.03518). URL: <https://arxiv.org/abs/2001.03518>.
- Radiojević, Tijana, Zak Costello, Kenneth Workman, and Hector Garcia Martin (2020). “A machine learning Automated Recommendation Tool for synthetic biology”. In: *Nature Communications* 11.1, p. 4879. ISSN: 2041-1723. DOI: [10.1038/s41467-020-18008-4](https://doi.org/10.1038/s41467-020-18008-4). URL: <https://doi.org/10.1038/s41467-020-18008-4>.
- Ji, Yuge, Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis (2021). “Machine learning for perturbational single-cell omics”. In: *Cell Systems* 12.6, pp. 522–537. ISSN: 2405-4712. DOI: <https://doi.org/10.1016/j.cels.2021.05.016>. URL: <https://www.sciencedirect.com/science/article/pii/S2405471221002027>.
- Mason, Derek M., Simon Friedensohn, Cédric R. Weber, Christian Jordi, Bastian Wagner, Simon M. Meng, Roy A. Ehling, Lucia Bonati, Jan Dahinden, Pablo Gainza, Bruno E. Correia, and Sai T. Reddy (2021). “Optimization of therapeutic antibodies by predicting antigen specificity from antibody sequence via deep learning”. In: *Nature Biomedical Engineering* 5.6, pp. 600–612. ISSN: 2157-846X. DOI: [10.1038/s41551-021-00699-9](https://doi.org/10.1038/s41551-021-00699-9). URL: <https://doi.org/10.1038/s41551-021-00699-9>.
- Peng, Grace C. Y., Mark Alber, Adrian Buganza Tepole, William R. Cannon, Suvaranu De, Salvador Dura-Bernal, Krishna Garikipati, George Karniadakis, William W. Lytton, Paris Perdikaris, Linda Petzold, and Ellen Kuhl (2021). “Multiscale Modeling Meets Machine Learning: What Can We Learn?” In: *Archives of Computational Methods in Engineering* 28.3, pp. 1017–1037. ISSN: 1886-1784. DOI: [10.1007/s11831-020-09405-5](https://doi.org/10.1007/s11831-020-09405-5). URL: <https://doi.org/10.1007/s11831-020-09405-5>.