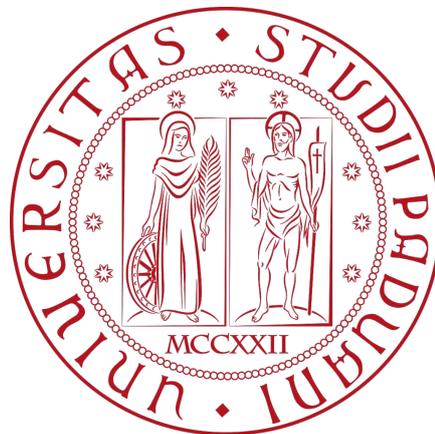

Data-driven Discovery of Mechanism of Systems Described by Partial Differential Equations



Author:
Theivan PASUPATHIPILLAI

Supervisors:
External: Prof. Rui DILÃO*
Internal: Prof. Samir SUWEIS†

Master of Science in Physics of Data
Department of Physics and Astronomy "Galileo Galilei"

December 6, 2023

UNIVERSITY OF PADOVA

Abstract

Department of Physics and Astronomy "Galileo Galilei"

Master of Science in Physics of Data

Data-driven Discovery of Mechanism of Systems Described by Partial Differential Equations

by Theivan PASUPATHIPILLAI

Turing patterns, a phenomenon introduced by mathematician and computer scientist Alan Turing, are intricate spatial patterns that emerge in reaction-diffusion systems, reflecting the dynamic interplay between chemical reactions and diffusion. These patterns, ranging from spots to stripes, offer physicists a captivating playground to explore the fundamental principles governing self-organization in complex systems. Studying Turing patterns not only unveils the underlying mechanisms of pattern formation but also provides valuable insights into the universal principles guiding the spontaneous emergence of order in nature.

This master thesis explores the application of a sparse regression framework to analyze synthetic Turing patterns and experimental data from *Drosophila* embryos. Focusing on the Brussellator model, the algorithm aims to identify coefficients of reaction-diffusion equations based on stationary state and oscillation data. In the analysis of synthetic Turing patterns, we observe that the algorithm's success is intricately linked to parameter selection, particularly the regularization strength and sparsity threshold. The delicate balance between the model's error and complexity, as measured by interaction sparsity, and illustrated by a complexity-error tradeoff. Despite promising results in low to moderate noise scenarios, the algorithm's sensitivity to noise, especially in Laplacian computation, remains a limitation. Extending the study to synthetic oscillation data reveals the algorithm's improved robustness to noise, with successful identification of diffusion in the second-best sparse model. However, challenges persist in accurate diffusion constant estimation and sensitivity to noise (mainly in derivative calculations). Applying the sparse regression framework to *Drosophila* Turing patterns uncovers additional challenges, primarily related to the high noise level in Laplacian computation, limiting the algorithm's accuracy in identifying reaction terms. The conclusions highlight the need for future developments, including the exploration of noise-robust methods, data augmentation strategies, integration with biological models, and advanced parameter optimization techniques to enhance the framework's robustness and applicability to real-world cases. Our work has thus contributed to a deeper understanding of complex biological phenomena governed by reaction-diffusion dynamics.

Contents

Abstract	iii
1 Introduction	1
1.1 Dynamical Systems	1
1.2 Data-driven Methods for System Identification	1
1.3 Sparse Regression for System Identification	2
1.4 Reaction-Diffusion Equation Discovery	3
1.5 Thesis Overview	3
2 Reaction-Diffusion Systems	5
2.1 A Biological Perspective	5
2.2 The Mass Action Law and Turing Instabilities	6
2.3 The Brusselator Model	9
3 Sparse Regression for System Identification	15
3.1 Sparse Identification of Nonlinear Dynamics (SINDy)	15
3.2 Partial Differential Equation Extension	17
4 Results and Discussion	19
4.1 Synthetic Turing Patterns Data	19
4.2 Synthetic Oscillations Data	23
4.3 Drosophila Turing Patterns	26
4.4 Remarks	29
5 Conclusions	31
5.1 Synthetic Turing Patterns Data	31
5.2 Synthetic Oscillations Data	31
5.3 Drosophila Turing Patterns	32
5.4 Further Developments	32
Bibliography	33

Chapter 1

Introduction

1.1 Dynamical Systems

Dynamical systems provide a robust mathematical framework for grasping the intricate interplay between evolving quantities over time. This framework revolves around the investigation, forecasting, and understanding of how a system's state evolves through a series of differential equations or iterative mappings. This versatile approach finds applications in a wide range of phenomena, spanning classical mechanical systems, electrical circuits, turbulent fluids, climate science, finance, ecology, social systems, neuroscience, epidemiology, and nearly every other system that undergoes changes over time.

The world of dynamical systems is a fascinating blend of mathematics, linking various domains such as linear algebra, differential equations, topology, numerical analysis, and geometry. It has become a cornerstone for modeling and analyzing systems across a spectrum of engineering, physical sciences, and life sciences [16].

Modern dynamical systems are currently experiencing a renaissance, transitioning from traditional analytical derivations and first-principles models to data-driven methodologies. The confluence of vast data resources and the power of machine learning is ushering in a transformative era for understanding and dissecting dynamical systems in the fields of science and engineering. Data is abundantly available, even as the fundamental laws or governing equations remain elusive, especially in domains like climate science, finance, epidemiology, and neuroscience. Even in well-established fields like optics and turbulence, where governing equations are firmly established, researchers are increasingly embracing data-driven approaches [9]. Many pressing challenges, reliant on data-driven insights, such as predicting climate change, deciphering neural data for cognitive understanding, curbing disease spread, or optimizing turbulence for energy-efficient power generation and transportation, stand to benefit from advancements in the data-driven discovery of dynamics.

1.2 Data-driven Methods for System Identification

A dynamical system is modeled by an equation of the form

$$\mathbf{u}_t(\mathbf{x}, t) = \mathbf{F}(\mathbf{x}, t, \mathbf{u}, \mathbf{u}_{x_i}, \mathbf{u}_{x_i x_j}, \dots), \quad (1.1)$$

where $\mathbf{u} = [u_1, u_2, \dots]^T$ are the dynamical quantities that describe the state of the systems, e.g. chemical concentrations, the subscript k in u_k is referring to the derivative of u w.r.t. the k variable, \mathbf{x} is a point in the spatial domain, t is time and \mathbf{F} is the functional that describes how the system evolves in time.

The goal of data-driven system identification is to learn the governing equations (1.1) and specifically the functional F . In the context of data-driven discovery, we can broadly categorize the approaches into three main groups [10]: classical sparse methods, classical symbolic methods, and deep modeling methods. It is worth noting that other ways of categorization may exist. The first category involves the use of sparse regression. In this approach, we put forward a library of potential solutions, and then we employ regularization-based techniques to identify the correct solution set, resulting in a succinct and sparse solution. The second category includes symbolic regression methods. This method allows us to learn or generate solutions through the estimation process, which provides a more symbolic representation of the data. The third category concerns the use of deep models to improve the discovery process. It integrates elements of the previous two approaches. For example, deep models can be integrated into symbolic regression for a more integrated and nuanced approach to discovery. A more articulated and nuanced approach to discovery

In this thesis, we use a method from the first category called Sparse Identification of Nonlinear Dynamics (SINDy) [2] and its extension to partial differential equation systems (PDE-FIND)[15].

1.3 Sparse Regression for System Identification

Utilizing sparse regression for system identification, as highlighted in [2], offers a scalable and less prone to over-fitting alternative in contrast to symbolic regression-based methods for equation discovery. Sparse regression algorithms take advantage of the inherent sparsity present in most physical systems' governing equations within a high-dimensional nonlinear function space, where only a few pertinent terms dictate system dynamics.

In the context of partial differential equations (PDEs), as demonstrated in [15], these algorithms have been extended. However, it's important to note that they exhibit heightened sensitivity to noise due to the numerical computation of derivatives. To address this concern, the weak formulation of SINDy, discussed in [7], has proven to be more robust when compared to other techniques.

In practical scenarios, system dynamics might involve latent variables that are challenging to measure. [13] elucidates how prior knowledge of the physical system can effectively tackle this issue while still deriving the dynamical equations via sparse regression.

In cases where the system's parameters are non-constant, identifying the correct set of equations governing the dynamics becomes a formidable task. [14] offers a generalization of the sparse regression method for parametric partial differential equations, addressing this challenge.

Furthermore, the algorithm's versatility is evident in situations where dynamical systems incorporate control parameters, enabling interactions with the system's behavior or accounting for external forcing parameters, as explored in [3]. Additionally, there are instances in the literature demonstrating hybrid approaches that harness the power of deep models and sparse regression, combining the strengths of neural networks as effective function approximators with the interpretability offered by sparse regression, as exemplified in [1].

1.4 Reaction-Diffusion Equation Discovery

Reaction-diffusion models are dynamical systems whose dynamics is driven by partial differential equations (PDE), typically showing a diffusion term and a reaction term as in the following equation

$$\mathbf{u}_t(\mathbf{x}, t) = \mathbf{D} \cdot \nabla^2 \mathbf{u} + \mathbf{R}(\mathbf{u}, \mathbf{x}, t), \quad (1.2)$$

where $\mathbf{u} = [u_1, u_2, \dots]^T$ are the chemical concentrations that describe the state of the systems in every point \mathbf{x} in space and instant t in time, ∇^2 is the Laplacian operator, \mathbf{D} is the vector of diffusion coefficients and \mathbf{R} is the vector of reaction terms that models how the components of \mathbf{u} interact.

Following Alan Turing's seminal work in 1952 concerning the chemical foundations of morphogenesis [18], both theoretical and experimental investigations have revealed that solutions to reaction-diffusion (RD) partial differential equations exhibit self-organizing characteristics, a phenomenon common to extensive biological and chemical systems. These properties manifest as the emergence of coherent spatial patterns or structures within extended media. Given the inherent non-linearity of local kinetic processes in RD systems, local fluctuations can grow and propagate through diffusion, influencing the surrounding environment. Moreover, reaction-diffusion equations, in their partial differential form, serve as fundamental mathematical tools for exploring pattern formation in excitable media. In the context of numerical simulations involving RD equations, one encounters undamped traveling waves, spiral traveling waves, stable strips, and distinctive spotty structures known as Turing patterns, as described in [4]. These patterns shed light on the intriguing dynamics of self-organization in complex systems.

In the realm of reaction-diffusion equation discovery, prior works in the literature offer valuable insights. In [19], the authors focus on defining a set of reaction-diffusion equations as dynamic descriptions of protein network interactions strengths. Their approach involves utilizing gradient-based optimization techniques to determine the equation parameters. However, it's important to note that this method does not extend to systems with entirely unknown reaction terms.

An innovative approach, which combines deep models and sparse regression, is presented in [12]. This approach effectively models reaction-diffusion reaction terms by incorporating prior knowledge of the underlying physics of the system to guide the architecture of the neural network (NN). The recurrent neural network is trained to model the reaction term of the RD equations which is then approximated by a polynomial sparse regression using SINDy.

The sparse regression method that we intend to employ [15] has been tested on a reaction-diffusion system with a domain space that is non-physical for chemical concentrations and populations, which are bound to be non-negative. It's worth highlighting that, to the best of our knowledge, no one has previously applied this particular method to calibrate experiments using real-world experimental data. This uncharted territory holds significant potential for advancing our understanding of real reaction-diffusion systems.

1.5 Thesis Overview

In this thesis, we want to investigate the feasibility of system identification of non-linear dynamics algorithm [2] on reaction-diffusion systems describing chemical interactions. The system we use to test the model is the Brusselator [11, 16], which is a

widely known reaction-diffusion system.

The study addresses the following key questions:

- **Algorithm Performance:**

- * *Can the algorithm accurately identify the equations governing simulated reaction-diffusion systems?*

- **Hyper-parameter Selection:**

- * *How can we optimize the selection of hyper-parameters for the algorithm?*

- **Noise Sensitivity:**

- * *How does the fully data-driven approach to equation discovery behave as noise levels increase?*

- **Applicability to Laboratory Data:**

- * *Can this approach be extended to analyze laboratory data, specifically focusing on protein concentrations in *Drosophila* embryos?*

The thesis is structured into the following main chapters:

1. **Chapter 2: Theoretical Foundation and Framework Introduction**

- * This chapter provides a theoretical introduction to reaction-diffusion systems and a detailed analysis of the reaction-diffusion model employed.

2. **Chapter 3: Sparse Regression Framework for System Identification**

- * In this chapter, we introduce the sparse regression framework as a key component of our system identification method. Theoretical aspects of sparse regression are discussed in detail, laying the groundwork for its application in subsequent chapters.

3. **Chapter 4: Algorithm Application and Laboratory Data Analysis**

- * This chapter presents the outcomes of the algorithm when applied to simulated systems in two distinct stationary states. Additionally, it explores the algorithm's capability to infer the dynamics of two proteins from their stationary states, using laboratory data obtained from *Drosophila* embryo experiments.

4. **Chapter 5: Conclusion**

- * The concluding chapter summarizes the key findings and implications of the research. It highlights the algorithm's performance, optimal hyper-parameter selection, noise sensitivity, and the potential applicability to laboratory data.

Chapter 2

Reaction-Diffusion Systems

2.1 A Biological Perspective

Reaction-diffusion models represent dynamical systems governed by partial differential equations (PDEs). These equations typically comprise a diffusion term and a reaction term, as depicted in Equation (2.1):

$$\mathbf{u}_t(x, t) = \mathbf{D}\nabla^2\mathbf{u} + \mathbf{R}(\mathbf{u}, x, t). \quad (2.1)$$

Here, $\mathbf{u} = [u_1, u_2, \dots]^T$ denotes the chemical concentrations characterizing the system's state at each spatial point x and time instance t . The Laplacian operator, denoted as ∇^2 , plays a pivotal role. Furthermore, \mathbf{D} encompasses the vector of diffusion coefficients, while \mathbf{R} is the vector of reaction terms representing interactions among the components of \mathbf{u} . These systems arise by adding kinetic reaction terms to the diffusion equations of different chemicals.

From a biological perspective, it was Alan Turing [18] who introduced the groundbreaking concept that diffusive coupling between reacting substances in spatially extended media or biological tissues can yield stable patterns known as Turing patterns (Fig. 2.1). These patterns are characterized by non-uniform and consistent spatial distributions of substance concentration. While intrinsic random movement or Brownian motion of molecules in the media tends to homogenize local concentrations and prevent steady gradients in most cases, the situation can be entirely different for certain reacting substances in the media. For instance, when two molecules with binding affinity collide and produce a third molecule, the local concentration of the third molecule increases. Turing's analysis went even further, suggesting the possibility of traveling wave-type phenomena in multi-component systems of reaction-diffusion equations.

The oscillatory behavior foreseen by Turing was experimentally confirmed in the Belousov-Zhabotinsky reaction (Fig. 2.2) in extended media [20]. This auto-catalytic



FIGURE 2.1: Left panel: an example of Turing patterns from simulations of reaction-diffusion equations. Right panel: an example of Turing pattern in nature, i.e. the skin pattern of a giant pufferfish.

reaction, initially conceived as an example of oscillatory chemical reactions, exhibits periodic oscillations in intermediate concentrations in homogeneous media and under an open flow regime. In extended media, fluctuations in the concentrations of reacting substances evolve into wave-like fronts that propagate throughout the media. By manipulating these wave fronts experimentally, it's possible to obtain growing spiral concentration waves, a phenomenon commonly observed in nature.

Moreover, reaction-diffusion systems can give rise to stable Turing patterns, and the experimental observation of Turing patterns in a chemical experiment is a relatively recent development [4]. This underscores the remarkable diversity and relevance of reaction-diffusion dynamics in various scientific domains.

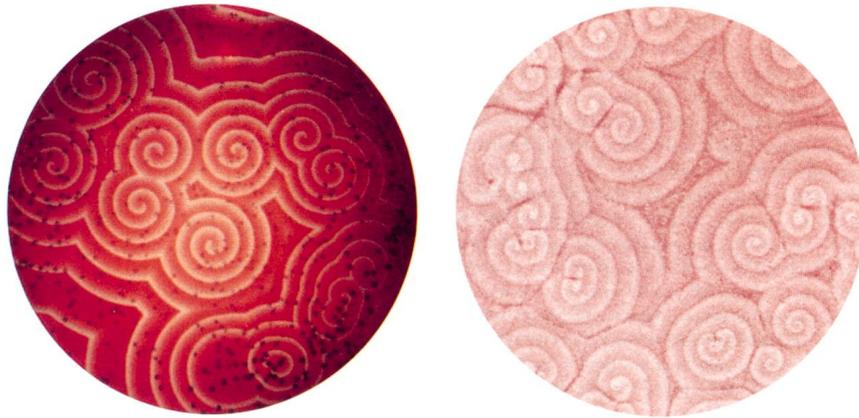
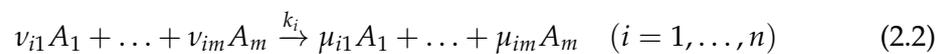


FIGURE 2.2: Patterns of the Belousov-Zhabotinsky reaction.

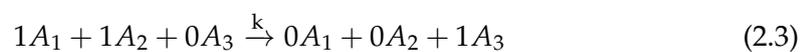
2.2 The Mass Action Law and Turing Instabilities

Before introducing the reaction diffusion model which we use, it is worth to spend some time discussing the framework of chemical reactions modelling and Turing instabilities.

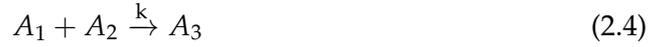
In a solution, atoms and molecules interact, potentially binding together when they possess the necessary chemical affinity, a quality determined by the nature of the chemical bonds in the colliding molecules. Assuming a total of m chemical substances and n chemical reactions, these reactions in the medium are illustrated through n collision diagrams:



Here, A_j represents the atom or molecule of substance number j , and the k_i 's denote rate constants. In the following discussion, we use the same symbols to represent both the atom or molecule of a substance and its concentration in the medium. The integers v_{ij} and μ_{ij} are stoichiometric coefficients (positive integers). The order of each reaction in (2.2) is given by $r_i = \sum v_{ij}$. These diagrams portray interactions between various molecules. For instance, if we have three molecules and molecule A_1 collides with molecule A_2 , resulting in the formation of molecule A_3 , the corresponding reaction diagram can be simplified as follows:



Or, in a simplified form:



Here, k represents the rate of formation of A_3 . In collision diagram number i , if $r_i = 2$, we have a binary collision or a reaction of order 2. If $r_i > 2$, we have a higher-order collision. Collisions involving three or more molecules have a very low probability of occurring and are generally not considered in chemical kinetics.

To derive the equations for the time evolution of the concentrations of the chemical substances in an ensemble of chemical transformations, it is assumed that:

- Substances are homogeneous in the medium and have low density.
- All reactions occur at constant volume and temperature.
- The individual motion of molecules in the medium is independent of other molecules, behaving as Brownian particles. Their collision frequency is proportional to the local concentration.

Under these conditions, the time evolution of the concentrations of all the chemical substances represented by the generic diagram (2.2) is described by a set of m ordinary differential equations:

$$\frac{dA_j}{dt} = \sum_{i=1}^n k_i (\mu_{ij} - \nu_{ij}) A_1^{\nu_{i1}} \dots A_m^{\nu_{im}} \quad (j = 1, \dots, m) \quad (2.5)$$

Here, A_j represents the concentration of chemical substance number j . This system of equations (2.5) is known as the law of mass action [8]. However, it's important to note that equations (2.5) are not necessarily independent. Representing them in the form:

$$\frac{d}{dt} \begin{pmatrix} A_1 \\ \vdots \\ A_m \end{pmatrix} = \begin{pmatrix} \mu_{11} - \nu_{11} & \cdots & \mu_{n1} - \nu_{n1} \\ \vdots & \ddots & \vdots \\ \mu_{1m} - \nu_{1m} & \cdots & \mu_{nm} - \nu_{nm} \end{pmatrix} \begin{pmatrix} k_1 A_1^{\nu_{11}} \cdots A_m^{\nu_{1m}} \\ \vdots \\ k_n A_1^{\nu_{n1}} \cdots A_m^{\nu_{nm}} \end{pmatrix} := \Gamma \begin{pmatrix} \omega_1 \\ \vdots \\ \omega_n \end{pmatrix} \quad (2.6)$$

the $n \times m$ matrix Γ has a rank r , where $r \leq \min(n, m)$. If $r = m$, then the equations in (3) are linearly independent. If $r < m$, the equations in (2.6) are linearly dependent. To analyze this situation, let

$$\mathbf{V}_j^T = (\mu_{1j} - \nu_{1j}, \dots, \mu_{nj} - \nu_{nj}) \quad \text{with } j = 1, \dots, m \quad (2.7)$$

denote the row vectors of Γ . Since $r < m$, there exist constants α_{jk} , with $j = 1, \dots, m$ and $k = 1, \dots, m - r$, such that:

$$\sum \alpha_{jk} \cdot \mathbf{V}_j = 0 \quad (2.8)$$

Given that each equation in (2.6) can be expressed as:

$$\frac{dA_j}{dt} = \mathbf{V}_j \cdot \boldsymbol{\omega} \quad (2.9)$$

We can conclude that:

$$\sum_{j=1}^m \alpha_{jk} \frac{dA_j}{dt} = \sum_{j=1}^m \alpha_{jk} \mathbf{V}_j \cdot \boldsymbol{\omega} = 0 \rightarrow \sum_{j=1}^m \alpha_{jk} A_j = \text{constant}_k \quad \text{for } k = 1, \dots, m - r \quad (2.10)$$

These relations (2.10) define $m - r$ conservation laws.

$$\sum_j \alpha_{jk} A_j(t) = \sum_j \alpha_{jk} A_j(0) \quad \text{for } k = 1, \dots, m - r \quad (2.11)$$

Turing implicitly assumed that all reactions occurring in a medium obey the mass action law. In biological tissues, concentrations are not uniformly distributed, and diffusion plays a vital role in creating local fluctuations in the concentration of certain chemical species. These diffusing substances were termed morphogens, form producers, or evocators.

When taking Turing's hypothesis to its full extent, the spatial patterns generated by a specific chemical mechanism can be described using a set of reaction-diffusion equations:

$$\frac{d}{dt} \begin{pmatrix} A_1 \\ \vdots \\ A_m \end{pmatrix} = \begin{pmatrix} \mu_{11} - \nu_{11} & \cdots & \mu_{n1} - \nu_{n1} \\ \vdots & \ddots & \vdots \\ \mu_{1m} - \nu_{1m} & \cdots & \mu_{nm} - \nu_{nm} \end{pmatrix} \begin{pmatrix} k_1 A_1^{\nu_{11}} \cdots A_m^{\nu_{1m}} \\ \vdots \\ k_n A_1^{\nu_{n1}} \cdots A_m^{\nu_{nm}} \end{pmatrix} + \begin{pmatrix} D_1 \nabla^2 A_1 \\ \vdots \\ D_m \nabla^2 A_m \end{pmatrix} \quad (2.12)$$

In this equation, the D_i values represent the diffusion coefficients associated with morphogens or form producers, and ∇^2 is the k -dimensional Laplacian operator. For modeling morphogenesis accurately, it's necessary to identify the morphogenic substances with $D_i > 0$ and the non-diffusing substances ($D_j = 0$). This requires precise knowledge of the developmental process under investigation. However, Turing adopted a more abstract approach. Turing discovered that coupling through diffusion of a system of two non-linear ordinary differential equations with a stable fixed point could lead to instability in the eigenmodes of the associated linearized reaction-diffusion equation. He constructed a one-dimensional spatial system composed of a finite number of interconnected cells arranged in a ring. In each cell, chemical substances evolve over time following the law of mass action (2.6). The flow between adjacent cells is proportional to the difference in local concentrations, and the extended spatial system is described by a reaction-diffusion system of equations. Specifically, Turing considered the following system of reaction-diffusion equations:

$$\begin{aligned} \frac{\partial \phi_1}{\partial t} &= f(\phi_1, \phi_2) + D_1 \nabla^2 \phi_1 \\ \frac{\partial \phi_2}{\partial t} &= g(\phi_1, \phi_2) + D_2 \nabla^2 \phi_2 \end{aligned} \quad (2.13)$$

Here, D_1 and D_2 are the diffusion coefficients for ϕ_1 and ϕ_2 , respectively, and the spatial domain is a k -dimensional cube with side length S . Typically, in the literature of reaction-diffusion systems and in Turing's work, it is implicitly assumed that the local equations in (2.13) are independent, and the phase space variables and parameters are also independent. If the local system associated with (2.13) has a fixed point at $(\phi_1, \phi_2) = (0, 0)$, linearizing around this fixed point yields:

$$\frac{d}{dt} \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} + \begin{pmatrix} D_1 \nabla^2 \phi_1 \\ D_2 \nabla^2 \phi_2 \end{pmatrix} \quad (2.14)$$

For initial data with Neumann boundary conditions (zero flux), the solutions of (2.14) have the general form:

$$\begin{aligned}\phi_1(x_1, \dots, x_k, t) &= \sum_{n_1, \dots, n_k \geq 0} c_{n_1, \dots, n_k}(t) \cos\left(\frac{2\pi n_1 x_1}{S}\right) \cdots \cos\left(\frac{2\pi n_k x_k}{S}\right) \\ \phi_2(x_1, \dots, x_k, t) &= \sum_{n_1, \dots, n_k \geq 0} d_{n_1, \dots, n_k}(t) \cos\left(\frac{2\pi n_1 x_1}{S}\right) \cdots \cos\left(\frac{2\pi n_k x_k}{S}\right)\end{aligned}\quad (2.15)$$

where $c_{n_1, \dots, n_k}(t)$ and $d_{n_1, \dots, n_k}(t)$ are the Fourier coefficients of the solution of (2.14). The terms under the sum in (2.15) are eigenmode solutions of (2.14) and are indexed by a k -tuple of non-negative integers. By introducing (2.15) into (2.14), we obtain the following infinite system of ordinary differential equations:

$$\begin{aligned}\frac{d}{dt} \begin{pmatrix} c_{n_1, \dots, n_k} \\ d_{n_1, \dots, n_k} \end{pmatrix} &= \begin{pmatrix} a_{11} - 4D_1(n_1^2 + \dots + n_k^2) & a_{12} \\ a_{21} & a_{22} - 4D_2(n_1^2 + \dots + n_k^2) \end{pmatrix} \begin{pmatrix} c_{n_1, \dots, n_k} \\ d_{n_1, \dots, n_k} \end{pmatrix} \\ &= J_{n_1, \dots, n_k} \begin{pmatrix} c_{n_1, \dots, n_k} \\ d_{n_1, \dots, n_k} \end{pmatrix}.\end{aligned}\quad (2.16)$$

with $(n_1, \dots, n_k) \geq (0, \dots, 0)$.

For each non-negative k -tuple of integers (n_1, \dots, n_k) , the stability of the eigenmode solutions of (2.16) is determined by the eigenvalues of the matrix J_{n_1, \dots, n_k} . Writing the eigenvalues of J_{n_1, \dots, n_k} as a function of the trace and determinant, we obtain,

$$\lambda_{n_1, \dots, n_k}^{\pm} = \frac{1}{2} \left(\text{Tr } J_{n_1, \dots, n_k} \pm \sqrt{(\text{Tr } J_{n_1, \dots, n_k})^2 - 4 \text{Det } J_{n_1, \dots, n_k}} \right) \quad (2.17)$$

By (2.16), for every $(n_1, \dots, n_k) \geq (0, \dots, 0)$, the real and imaginary parts of (2.17) are bounded from above, and we can define the number,

$$\Lambda = \max \{ \text{Re}(\lambda_{n_1, \dots, n_k}^{\pm}) : (n_1, \dots, n_k) \geq (0, \dots, 0) \} \quad (2.18)$$

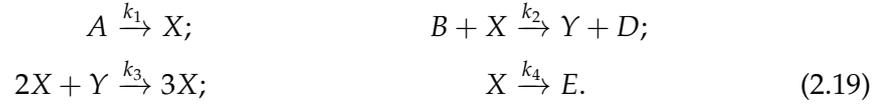
The number Λ is the upper bound of the spectral abscissas of the set of matrices $\{J_{n_1, \dots, n_k} : (n_1, \dots, n_k) \geq (0, \dots, 0)\}$.

This linear analysis led Turing to find that, in the case of two or more diffusive and reacting substances, a stable state of the local system (2.14) could be destabilized by the diffusion terms, inducing a symmetry breaking in the global behavior of the solutions of the non-linear system (2.13). This effect is called Turing or diffusion-driven instability. Depending on the type and magnitude of the unstable eigenmodes of the linearized reaction-diffusion system, different asymptotic states of the non-linear system could eventually be reached. Turing implicitly conjectured that if the eigenvalue with dominant real part of all the unstable eigenmodes of the linearized system is real, an asymptotic time-independent solution of the non-linear equation could eventually be reached. On the other hand, if the eigenvalue with the dominant real part of all the unstable eigenmodes is complex, the solution could evolve into a time periodic spatial function or wave. In the first case, we are in the presence of a Turing instability, and in the second case, we have an oscillatory instability.

2.3 The Brusselator Model

The reaction-diffusion model used to test the sparse regression method is based on the Brusselator model proposed in [11]. According to the authors, this model mimics

an auto-catalytic process and has the following kinetic mechanisms:



Here, X is the auto-catalytic chemical substance. Applying the mass action law to (2.19), we obtain the following system of differential equations:

$$\frac{\partial X}{\partial t} = k_1 A - k_2 B X + k_3 X^2 Y - k_4 X \quad (2.20)$$

$$\frac{\partial Y}{\partial t} = k_2 B X - k_3 X^2 Y \quad (2.21)$$

$$\frac{\partial A}{\partial t} = -k_1 A \quad (2.22)$$

$$\frac{\partial B}{\partial t} = -k_2 B X \quad (2.23)$$

$$\frac{\partial E}{\partial t} = k_4 X \quad (2.24)$$

$$\frac{\partial D}{\partial t} = k_2 B X \quad (2.25)$$

These equations are subject to the conservation laws:

$$B(t) + D(t) = B(0) + D(0); \quad (2.26)$$

$$X(t) + Y(t) + A(t) + E(t) = X(0) + Y(0) + A(0) + E(0); \quad (2.27)$$

Assuming that A and B are constants, which are supplied during the reaction (open flow reactor experiments), the reaction-diffusion Brusselator model becomes:

$$\begin{cases} \frac{\partial X}{\partial t} = k_1 A - k_2 B X + k_3 X^2 Y - k_4 X + D_1 \nabla^2 X \\ \frac{\partial Y}{\partial t} = k_2 B X - k_3 X^2 Y + D_2 \nabla^2 Y \end{cases} \quad (2.28)$$

Here, D_1 and D_2 are diffusion coefficients, and ∇^2 is the Laplace operator. The local component of the vector field associated with equation (2.28) has a fixed point with coordinates:

$$(X_0, Y_0) = \left(\frac{A k_1}{k_4}, \frac{B k_2 k_4}{A k_1 k_3} \right). \quad (2.29)$$

If $B < \frac{k_4}{k_2} + \frac{A^2 k_1^2 k_3}{k_2 k_4^2}$, (X_0, Y_0) is a stable focus. If $B > \frac{k_4}{k_2} + \frac{A^2 k_1^2 k_3}{k_2 k_4^2}$, (X_0, Y_0) is an unstable focus, and the local vector field has a supercritical Hopf bifurcation for $B = \frac{k_4}{k_2} + \frac{A^2 k_1^2 k_3}{k_2 k_4^2}$.

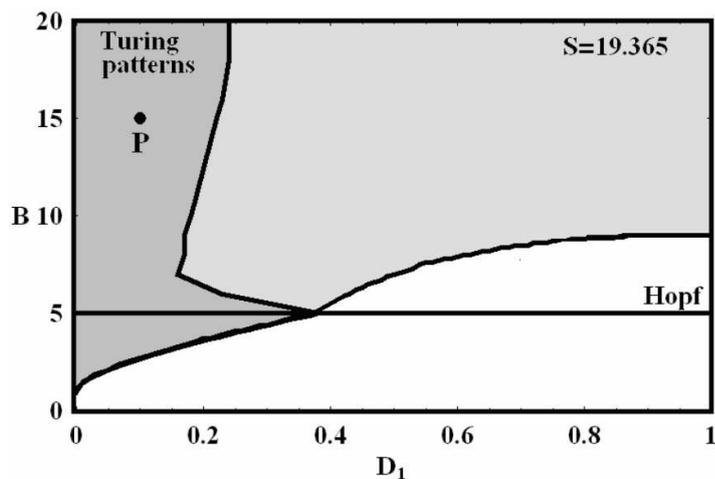


FIGURE 2.3: Bifurcation diagram of the solutions for the Brusselator model (2.28)[6], with parameter values $A = 2$, $k_1 = k_2 = k_3 = k_4 = 1$, and $D_2 = 1$, in a one-dimensional domain of length $S = 19.365$. The line $B = 5$ indicates the supercritical Hopf bifurcation of the fixed point (X_0, Y_0) within the local system. The grey regions correspond to parameter values where the origin exhibits a Turing instability. Numerical integration of equations (2.28) for initial conditions slightly deviating from the steady state (X_0, Y_0) reveals Turing patterns exclusively in the dark-grey region. If $B > 5$, solutions within the light-grey and white regions exhibit spatial constancy and temporal oscillations, with a period defined by the limit cycle. If $B \leq 5$, solutions within the white regions remain time-independent and spatially constant.

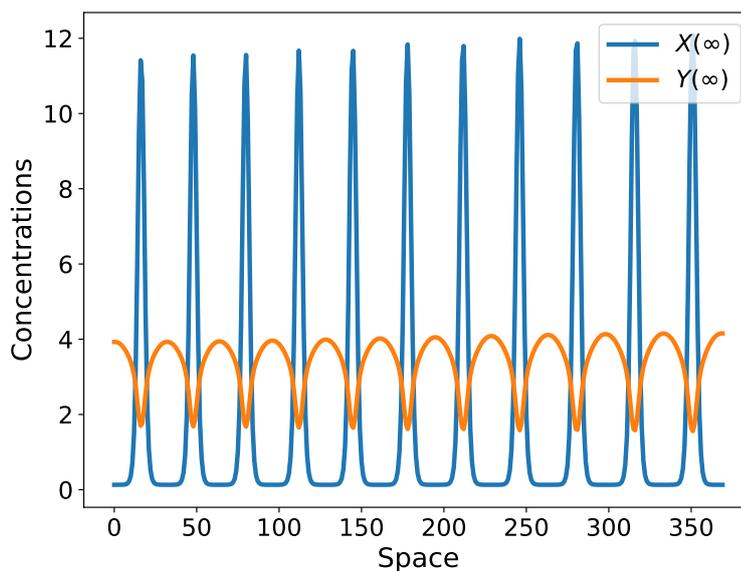


FIGURE 2.4: The image depicts the steady state of the Brusselator model, numerically integrated with a time step of $dt = 0.001$, spatial step $dx^2 = 0.006$, and model parameters set to $k_1 = k_2 = k_3 = k_4 = 1$, $D_1 = 0.1$, $D_2 = 1$, $A = 2$, and $B = 15$.

Fig. 2.3 explores the parameter space of the Brusselator model to visualize regions of Turing instabilities and Turing patterns. Numerical integration for various values of B and D_1 delineates the boundary separating the dark and light-grey regions in Fig. 2.3. Within the dark-grey region, small perturbations from the steady state evolve into time-independent and spatially non-homogeneous states, known as Turing patterns.

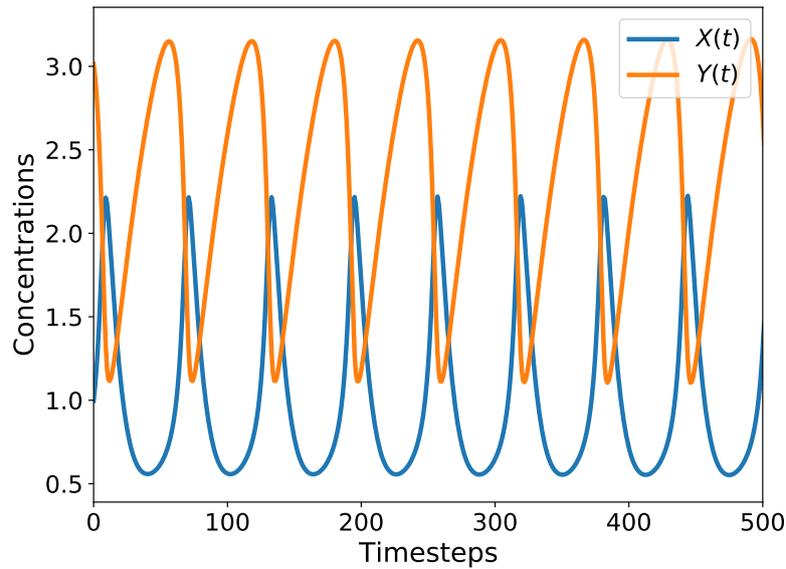


FIGURE 2.5: This image displays the oscillatory behavior of a specific spatial point in our system. The Brusselator model is numerically integrated with a time step of $dt = 0.1$, spatial step $dx^2 = 0.6$, and model parameters set to $k_1 = k_2 = k_3 = k_4 = 1$, $D_1 = 0.1$, $D_2 = 1$, $A = 1$, and $B = 2.3$.

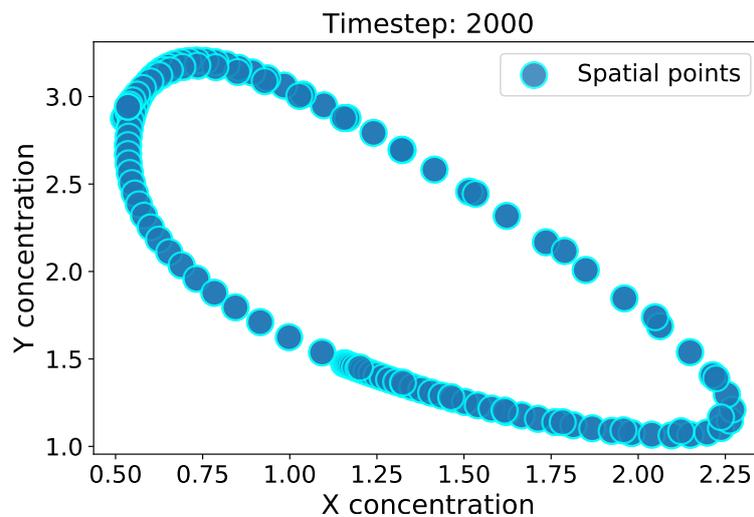


FIGURE 2.6: This figure illustrates the limit cycle towards which the dynamics tend in the oscillatory regime of the Brusselator model. Each point correspond to a point in the one dimensional system at timestep $t = 2000$.

To evaluate the sparse regression model, we simulate the Brusselator model under varying conditions. In Fig. 2.4, we visualize the steady state of the Brusselator model under parameters conducive to Turing patterns. The oscillatory behavior is achieved with parameters $k_1 = k_2 = k_3 = k_4 = 1$, $D_1 = 0.1$, $D_2 = 1$, $A = 2$, and $B = 2.3$. Fig. 2.5 showcases the oscillations at a specific spatial point within the system. In Fig. 2.6, we visualize the system's limit cycle in the oscillatory regime at a significant time step $t \gg 1$.

Chapter 3

Sparse Regression for System Identification

3.1 Sparse Identification of Nonlinear Dynamics (SINDy)

Discovering dynamical system models from data stands as a central challenge in mathematical physics, with a rich historical lineage dating back to the time of Kepler and Newton when the laws governing planetary motion were first revealed. Traditionally, this process relied on a blend of high-precision measurements and expert intuition. However, in today's era of abundant data and increasing computational capabilities, the automated discovery of governing equations and dynamical systems represents a novel and exciting scientific paradigm.

Typically, one of two approaches is followed: either the form of a candidate model is constrained based on prior knowledge of the governing equations, or a few heuristic models are tested and their parameters optimized to align with the available data. Nonetheless, simultaneously identifying both the nonlinear structure and parameters of a model from data poses a considerably more formidable challenge due to the vast number of possible model structures to consider.

The Sparse Identification of Nonlinear Dynamics (SINDy) algorithm [2] bypasses this demanding task of exploring all possible model structures through a clever strategy, exploiting the fact that many dynamical systems can be represented as

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}), \quad (3.1)$$

where the dynamics \mathbf{f} typically involve only a sparse set of active terms within the space of potential right-hand side functions.

The approach is to approximate \mathbf{f} using a generalized linear model:

$$\mathbf{f}(\mathbf{x}) \approx \sum_{k=1}^p \theta_k(\mathbf{x})\xi_k = \mathbf{\Theta}(\mathbf{x})\boldsymbol{\xi}, \quad (3.2)$$

where the objective is to identify the fewest non-zero terms in $\boldsymbol{\xi}$ as possible. This is achieved by solving for the pertinent terms active in the dynamics using sparse regression, which penalizes the inclusion of unnecessary terms in the model and is scalable to handle large and complex problems.

The practical implementation of SINDy involves the collection of time-series data from (3.1), which is structured into a data matrix:

$$\mathbf{X} = [\mathbf{x}(t_1) \quad \mathbf{x}(t_2) \quad \dots \quad \mathbf{x}(t_m)]. \quad (3.3)$$

An analogous matrix is constructed for the derivatives of the data:

$$\dot{X} = [\dot{x}(t_1) \quad \dot{x}(t_2) \quad \dots \quad \dot{x}(t_m)]^T. \quad (3.4)$$

In practice, this matrix derivative can be directly computed from the data within X . For noisy data, a total-variation regularized derivative calculation often provides robust results [5]. When dealing with noisy data, alternative methods include data smoothing through convolution with a kernel (e.g., Gaussian) or employing polynomial interpolation to derive locally smooth derivatives. If the data is noise-free, a standard finite difference method can suffice for derivative estimation.

A library of candidate nonlinear functions $\Theta(X)$ is then assembled based on the data within X :

$$\Theta(X) = [\mathbf{1} \quad X \quad X^2 \quad \dots \quad X^d \quad \dots \quad \sin X \quad \dots] \quad (3.5)$$

Here, X^d represents a matrix where the column vectors comprise all conceivable time series of d -th-degree polynomials based on the state x . Essentially, the library of candidate functions is bound solely by one's creative imagination.

The dynamical system described in (3.1) can be recast in terms of the data matrices presented in (3.4) and (4.9) as follows:

$$\dot{X} = \Theta(X)\Xi. \quad (3.6)$$

Each column ξ_k in Ξ represents a vector of coefficients that determine the active terms in the k -th row of (3.1). A parsimonious model seeks an accurate fit in (3.6) with the fewest terms in Ξ , which can be identified using convex ℓ_1 -regularized sparse regression:

$$\xi_k = \arg \min_{\xi'_k} \|\dot{X}_k - \Theta(X)\xi'_k\|_2 + \lambda \|\xi'_k\|_1. \quad (3.7)$$

In this equation, \dot{X}_k refers to the k -th column of \dot{X} , and λ serves as a regularization parameter that promotes sparsity. Sparse regression methods like the Least Absolute Shrinkage and Selection Operator (LASSO) [17] or the Sequential Thresholded Least-Squares (STLS) algorithm used in SINDy [2] enhance the robustness of this identification process, particularly for noisy over-determined problems.

Various regularization terms can be employed to solve the sparse regression problem, such as ℓ_2 , ℓ_1 , and ℓ_0 norms for sparsity control. Additionally, an alternative option is the Sparse Relaxed Regularized Regression (SR3) [21], which introduces a relaxation of the regularized problem and generally excels in terms of errors, false positives, and conditioning.

The Sequential Thresholded Least Squares algorithm, STLSQ($\Theta, \dot{X}, threshold, n, iterations$), outlines the steps involved in the sparse regression process:

The outcome of the SINDy regression process yields a parsimonious model that includes only the essential terms necessary to explain the observed behavior. In contrast, the brute-force search through the extensive array of potential model structures, which would involve regression onto every possible sparse nonlinear structure, is computationally infeasible. SINDy offers a modern and efficient solution based on convex optimization and machine learning.

It's noteworthy that in the case of discrete-time dynamics, when $\Theta(X)$ primarily consists of linear terms and λ is set to zero to remove sparsity promotion, this

Algorithm 1 STLSQ($\Theta, \dot{X}, threshold, n, iterations$)

```

1:  $\Xi \leftarrow \Theta^\dagger \dot{X}$  ▷ Initial guess: Least-squares
2: for  $k \leftarrow 1$  to  $iterations$  do
3:    $smallinds \leftarrow (\text{abs}(\Xi) < threshold)$  ▷ Find small coefficients
4:    $\Xi[smallinds] \leftarrow 0$  ▷ Threshold small coefficients
5:   for  $ind \leftarrow 1$  to  $n$  do ▷  $n$  is the state dimension
6:      $biginds \leftarrow \neg smallinds[:, ind]$ 
7:      $\Xi[biginds, ind] \leftarrow \Theta[:, biginds]^\dagger \dot{X}[:, ind]$  ▷ Regress dynamics onto
       remaining terms
8:   end for
9: end for
10: return  $\Xi$ 

```

algorithm reduces to Dynamic Mode Decomposition (DMD). However, when least-squares regression, akin to DMD, is employed, any level of measurement error or numerical imprecision will activate all terms in the library, resulting in a non-physical model. SINDy's strength lies in its capacity to identify parsimonious models that exclusively encompass the necessary nonlinear terms, thus yielding interpretable models that circumvent overfitting.

3.2 Partial Differential Equation Extension

A significant extension of the SINDy modeling framework involves the generalization of the library to include partial derivatives, enabling the identification of partial differential equations. This resulting algorithm, named the Partial Differential Equation Functional Identification of Nonlinear Dynamics (PDE-FIND), has proven successful in identifying several canonical PDEs from classical physics purely based on noisy data. These PDEs include Navier–Stokes, Kuramoto–Sivashinsky, Schrödinger, reaction–diffusion, Burgers, Korteweg–de Vries, and the diffusion equation for Brownian motion [15].

PDE-FIND shares similarities with SINDy, as it relies on sparse regression in a library constructed from measurement data. The spatial time-series data is organized into a single column vector $\mathbf{Y} \in \mathbb{C}^{mn}$, representing data collected over m time points and n spatial locations. Additional inputs, such as a known potential for the Schrödinger equation or the magnitude of complex data, are arranged into a column vector $\mathbf{Q} \in \mathbb{C}^{mn}$. Next, a library $\Theta(\mathbf{Y}, \mathbf{Q}) \in \mathbb{C}^{mn \times D}$ of D candidate linear and nonlinear terms, as well as partial derivatives for the PDE, is constructed. Derivatives are computed either using finite differences for clean data or, when noise is present, through polynomial interpolation or by convolving the signal with a smoothing filter and then computing finite differences. The candidate linear and nonlinear terms, along with the partial derivatives, are combined into a matrix $\Theta(\mathbf{Y}, \mathbf{Q})$, which takes the form:

$$\Theta(\mathbf{Y}, \mathbf{Q}) = [1 \quad \mathbf{Y} \quad \mathbf{Y}^2 \quad \dots \quad \mathbf{Q} \quad \dots \quad \mathbf{Y}_x \quad \mathbf{Y}\mathbf{Y}_x \quad \dots]. \quad (3.8)$$

Each column of Θ contains all the values of a particular candidate function across all mn space-time grid points where data is collected. The time derivative \mathbf{Y}_t is computed and reshaped into a column vector. The PDE evolution can be expressed within this library as follows:

$$\mathbf{Y}_t = \Theta(\mathbf{Y}, \mathbf{Q})\xi \quad (3.9)$$

Each entry in ξ corresponds to a coefficient for a term in the PDE. For canonical PDEs, the vector ξ is sparse, indicating that only a few terms are active.

If the library has a rich enough column space to represent the dynamics, the PDE can be well approximated by (3.9) with a sparse vector of coefficients ξ . To identify the active terms in the dynamics, sparsity-promoting regression, similar to SINDy, is employed. It's crucial to note that the regression problem in (3.9) may suffer from poor conditioning. Errors in computing the derivatives may be amplified when inverting Θ . Therefore, a least-squares regression significantly alters the qualitative nature of the inferred dynamics.

In practice, the goal is to find the sparsest vector ξ that satisfies (3.9) with a small residual. Instead of attempting a computationally intractable combinatorial search for all possible sparse vector structures, a common technique is to relax the problem to a convex ℓ_1 -regularized least-squares [17]. However, this may perform poorly with highly correlated data. Instead, ridge regression with hard thresholding is employed, known as sequential threshold ridge regression (STRidge), as presented in Algorithm 2 (reproduced from Rudy et al. [15]). By iteratively refining the tolerance of Algorithm 2, the best predictor is determined based on selection criteria.

In [15] the authors extend the STRidge algorithm to an ℓ_0 -regularized optimization problem as follows:

$$\xi = \arg \min_{\xi} \|\Theta(\mathbf{Y}, \mathbf{Q})\xi - \mathbf{Y}_t\|_2^2 + \epsilon \kappa(\Theta(\mathbf{Y}, \mathbf{Q})) \|\xi\|_0 \quad (3.10)$$

where $\kappa(\Theta)$ is the condition number of the matrix Θ , offering more robust regularization for ill-posed problems. The penalty term $\|\xi\|_0$ discourages over-fitting by selecting from an optimal position in a Pareto front.

Algorithm 2 STRidge($\Theta, \mathbf{Y}_t, \alpha, \text{threshold}, \text{iters}$)

```

 $\hat{\xi} \leftarrow \arg \min_{\xi} \|\Theta\xi - \mathbf{Y}_t\|_2^2 + \alpha \|\xi\|_2^2$ 
bigcoeffs  $\leftarrow \{j : |\hat{\xi}_j| \geq \text{threshold}\}$ 
 $\hat{\xi}[\sim \text{bigcoeffs}] \leftarrow 0$ 
 $\hat{\xi}[\text{bigcoeffs}] \leftarrow \text{STRidge}(\Theta[:, \text{bigcoeffs}], \mathbf{Y}_t, \text{threshold}, \text{iters} - 1)$ 
return  $\hat{\xi}$ 

```

After having implemented a rudimentary version of the algorithm for educational purposes, for the analysis undertaken in this thesis we leveraged the optimized Python package PySindy [pysindy] to discover equations within the analyzed datasets.

Chapter 4

Results and Discussion

4.1 Synthetic Turing Patterns Data

We start our analysis by employing the sparse regression framework on the stationary state of the Brussellator within the Turing patterns regime. In this state, the system satisfies the following set of equations.

$$\begin{cases} \frac{\partial X}{\partial t} = k_1 A - k_2 B X + k_3 X^2 Y - k_4 X + D_1 \nabla^2 X = 0 \\ \frac{\partial Y}{\partial t} = k_2 B X - k_3 X^2 Y + D_2 \nabla^2 Y = 0 \end{cases} \quad (4.1)$$

This means that we can express the diffusion terms as the reaction terms changed of sign and rescaled by the diffusion constants.

$$\begin{cases} \nabla^2 X = -20 + 160X - 10X^2 Y \\ \nabla^2 Y = -15X + X^2 Y \end{cases} \quad (4.2)$$

Where we have substituted the parameters with the actual coefficient of the simulated system. To apply the sparse regression algorithm we have to compute the Laplacian (Fig. 4.1) starting from the data. This procedure is done by finite differences and it is highly sensible to noise.

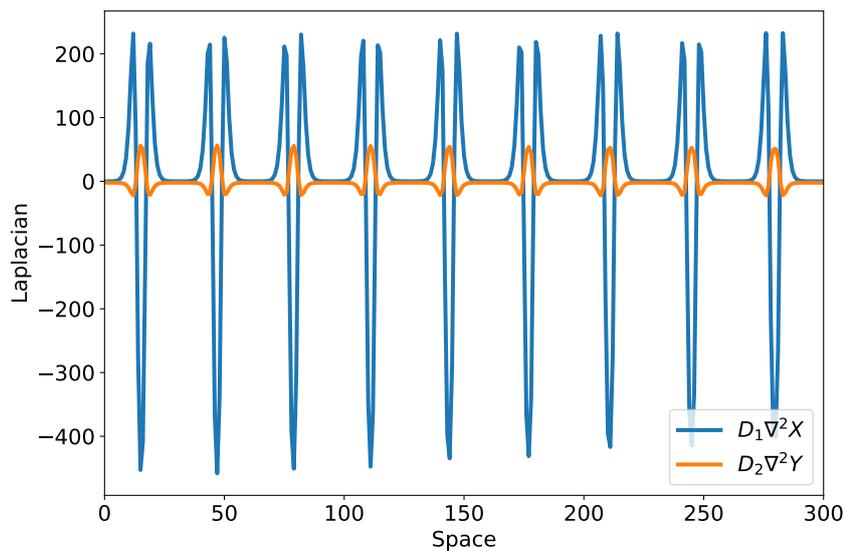


FIGURE 4.1: This image displays the computed discrete Laplacian of the simulated Turing patterns.

After having computed the Laplacian we want to apply the sparse identification of nonlinear dynamics (SINDy), and write the Laplacian as a matrix product as follows

$$\begin{cases} \nabla^2 X &= \Theta(X, Y) \xi_x \\ \nabla^2 Y &= \Theta(X, Y) \xi_y, \end{cases} \quad (4.3)$$

where X and Y are the concentrations vectors of the system with dimensions $M = 370$, ξ_x and ξ_y are the sparse vectors with most of the components equal to 0, $\Theta(X, Y)$ is the library of functions, which in this case is composed by all the polynomial in X and Y up to degree 4.

$$\Theta(X, Y) = [\mathbf{1} \quad X \quad Y \quad X^2 \quad XY \quad Y^2 \quad \dots \quad X^4 \quad Y^4]. \quad (4.4)$$

After choosing the threshold parameter and the regularization strength α we can apply the algorithm. At each iteration of the sequential thresholded algorithm, we choose the sparse vector coming from the following optimization.

$$\begin{cases} \xi_x = \arg \min_{\xi_x'} \{ \|\nabla^2 X - \Theta(X, Y) \xi_x'\|_2^2 + \alpha \|\xi_x'\|_2^2 \} \\ \xi_y = \arg \min_{\xi_y'} \{ \|\nabla^2 Y - \Theta(X, Y) \xi_y'\|_2^2 + \alpha \|\xi_y'\|_2^2 \} \end{cases} \quad (4.5)$$

The process of identifying the optimal sparse model is significantly shaped by the selection of the regularization parameter, α , and the choice of the threshold. The threshold operates iteratively, selectively discarding less relevant columns in the regression matrix. A high threshold encourages greater sparsity, potentially leading to a scenario where no columns are selected, and every term in the sparse vector is set to 0, a condition indicative of minimal model complexity. Conversely, a low threshold results in every component of the sparse vector being non-zero, signaling potential over-fitting of the model.

The pivotal role of the threshold becomes evident as it governs the delicate trade-off between model error and complexity. The regularization strength, denoted by α , further fine-tunes this balance by influencing the magnitude of terms in the sparse vector. A higher α imposes penalties on larger-normed vectors, promoting a "simpler" solution. Conversely, lower α values may result in larger terms within the sparse vector, potentially leading to over-fitting. To summarize, if we hold α constant, a higher threshold fosters sparsity in the vector. On the other hand, fixing the threshold while increasing α encourages a sparser vector. In the presence of noise, a higher α becomes indispensable to mitigate the risk of over-fitting. This trade-off is visually evident in Fig. 4.2, which illustrates the relationship between model complexity, quantified by the number of nonzero coefficients in the sparse vectors, and error as a function of threshold.

The selection of the optimal parameters, which include both regularization strength and threshold, is achieved by grid search. The ideal parameter set achieves an equilibrium by ensuring that the sparse vectors contain a minimum number of nonzero terms. Choosing a more complex solution than the optimal one does not produce a significant reduction in error, while choosing a simpler solution leads to a significant increase in error. Fig. 4.3 shows how the choice is made. After conducting the grid search we can plot the error as a function of the number of non-zero coefficients

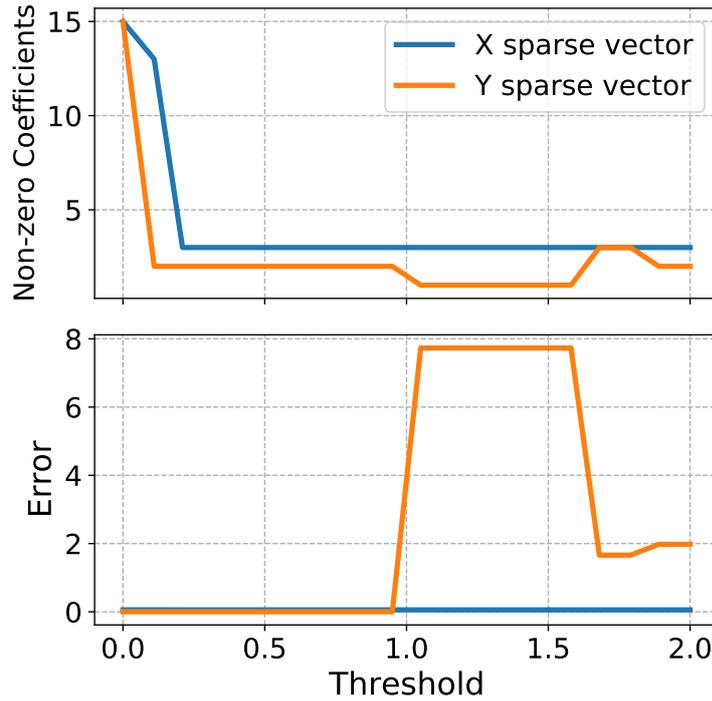


FIGURE 4.2: This image shows the existing trade-off between the complexity of the model and the error which is controlled by the magnitude of the threshold. For this plot, the regularization strength is fixed to $\alpha = 10^{-5}$. The error is measured as Mean Absolute Error (MAE).

of the sparse vectors. We plot this for every parameters choice and choose the best option according to the error and the complexity of the model (Fig. 4.3).

Our goal is to employ this method for the accurate identification of dynamic system equations based on real data. To pursue this goal, we evaluated the robustness of the algorithm in the presence of different levels of noise to ensure its reliability and performance under different conditions.

As demonstrated in Table 4.1, the algorithm shows a good ability to accurately identify the coefficients that constitute the differential equations (4.12) up to a specific noise threshold.

The impact of noise is closely related to the process of calculating the derivatives by finite differences. Moreover, the calculation involves obtaining a second-order derivative from a signal affected by noise, with the noise level of the derivative increasing proportionally to the order of differentiation. During this research we also explore other possible approaches that are more noise-robust to calculate derivatives, such as:

- Polynomial Interpolation: we define a rolling window and a degree k . In each window we interpolate the point with a polynomial of degree k and then we take the derivative of the polynomial. This method introduces two more hyper-parameter which are the size of the window and the degree of the polynomial.
- Total Variation Denoising: We get a smoothed signal by minimizing its total variation and then we compute finite differences derivatives. The optimization

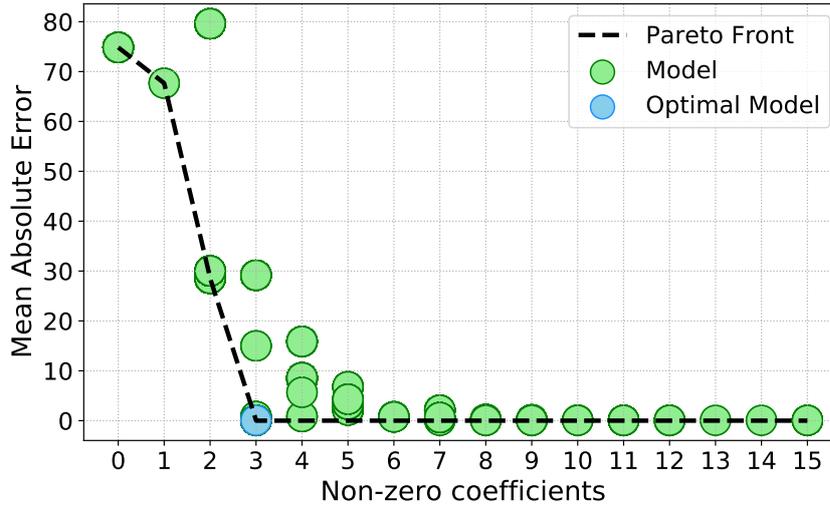


FIGURE 4.3: Each point in the graph is an output model for specific hyper-parameters, i.e. the threshold and the regularization strength. For each model we plot its error and its complexity measured in number of non-zero coefficient of the sparse vector. It is clear that there is a Pareto optimal solution that represent the best choice for the hyper-parameters. The plot is referred to concentration X .

is the following one

$$\arg \min_u \left\{ \lambda \sum_{i=1}^{N-1} |u_{i+1} - u_i| + \frac{1}{2} \sum_{i=1}^N (u_i - u_{0_i})^2 \right\}$$

where u_0 is the one dimensional noisy signal and u is the denoised one.

- Smoothed Finite Differences: we convolve the data with a smoothing filter, e.g. a Gaussian or a moving average filter. After this, we evaluate derivatives through finite differences.

All the methods for computing derivatives mentioned above do not exhibit comparable performance to finite differentiation, whether in the presence or absence of noise. The attempts to denoise or smooth the signal appear to result in a substantial loss of information regarding the system dynamics. Consequently, the algorithm outputs a model that is less accurate than the one obtained through finite differentiation.

To address the issue of noise, we explored an alternative sparse regression optimization algorithm known as Sparse Relaxed Regularized Regression (SR3) [21]. This optimization approach is supposedly more robust to noise, as suggested by the literature.

$$\begin{cases} \xi_x = \arg \min_{\xi'_x} \frac{1}{2} \|\nabla^2 X - \Theta(X, Y) \xi'_x\|_2^2 + \alpha \|u\|_0 + \frac{1}{2\nu} \|\xi'_x - u\|_2^2 \\ \xi_y = \arg \min_{\xi'_y} \frac{1}{2} \|\nabla^2 Y - \Theta(X, Y) \xi'_y\|_2^2 + \alpha \|u\|_0 + \frac{1}{2\nu} \|\xi'_y - u\|_2^2 \end{cases} \quad (4.6)$$

Despite the implementation of this strategy, the performance of the best model showed marginal improvement over the Sequential Thresholded Ridge Regression (STRidge) used initially. The results, incorporating more than six polynomial terms,

Noise	Chemical	Alpha	Threshold	Reaction Term
0%	X	10^{-5}	0.2	$-20.000 + 159.999x - 10.000x^2y$
	Y	10^{-5}	0.1	$-15.000x + 1.000x^2y$
0.001%	X	10^{-5}	0.2	$-20.000 + 159.999x - 10.000x^2$
	Y	10^{-5}	0.1	$-15.000x + 1.000x^2y$
0.01%	X	10^{-5}	0.2	$-20.001 + 159.999x - 10.000x^2y$
	Y	10^{-5}	0.1	$-15.000x + 1.000x^2y$
0.1%	X	10^{-5}	4.72	$-20.003 + 159.973x - 9.998x^2y$
	Y	10^{-4}	0.2	$-14.998x + 1.000x^2y$
0.5%	X	5×10^{-3}	5.226	$-19.857 + 159.490x - 9.972x^2y$
	Y	10^{-2}	0.6	$-14.970x + 0.998x^2y$
1%	X	0.5	6.7	$12.680x^2 + 25.685xy - 15.990x^2y +$ $57.876xy^2 - 16.450xy^3$
	Y	0.5	0.5	$-3.791xy + 0.554x^2y$

TABLE 4.1: In the table are listed the results of the best model for different noise levels. If the noise is small enough the algorithm correctly identifies the reaction terms of the the reaction-diffusion system. The reported noise levels are expressed as a percentage of the standard deviation of the noise-free dataset.

clearly indicated signs of over-fitting. As a result, we chose to rely exclusively on the more efficient STRidge algorithm.

As highlighted earlier, a notable limitation stems from the poor noise tolerance associated with the way we compute derivatives. Addressing this issue is critical for further refinement of the algorithm. Furthermore, when dealing with Turing models, unlike the oscillation regime, we face the challenge of limited data availability. In this regime, the model is fit with only 370 data points, in sharp contrast to the few thousand we can use for the time-dependent oscillation regime. This discrepancy emphasizes the need for nuanced considerations and adjustments in the modeling approach for the different regimes.

4.2 Synthetic Oscillations Data

We now aim to explore the algorithm's performance using the same set of reaction-diffusion equations but with different parameters. Specifically, the selected coefficients for the equations (4.7) induce a steady state marked by sustained oscillations.

$$\begin{cases} \frac{\partial X}{\partial t} = 1 - 3.3X + X^2Y + \nabla^2 X \\ \frac{\partial Y}{\partial t} = 2.3X - X^2Y + 0.01\nabla^2 Y \end{cases} \quad (4.7)$$

In this case, we are looking for the sparse vectors that best satisfy the following equation.

$$\begin{cases} \frac{\partial X}{\partial t} = \Theta(X, Y)\xi_x \\ \frac{\partial Y}{\partial t} = \Theta(X, Y)\xi_y \end{cases} \quad (4.8)$$

In this case the regression matrix Θ is composed of all polynomial terms in X and Y up to degree 4 and spatial derivative terms up to second order.

$$\Theta(X, Y) = \left[\mathbf{1} \quad X \quad Y \quad X^2 \quad XY \quad Y^2 \quad \dots \quad \frac{\partial X}{\partial z} \quad \dots \quad \frac{\partial^2 X}{\partial z^2} \right] \quad (4.9)$$

where z is the spatial dimension of the one-dimensional system. In this case, the optimization algorithm run at each step of the iterative thresholded method is the solution of the following equations.

$$\begin{cases} \xi_x = \arg \min_{\xi'_x} \left\| \frac{\partial X}{\partial t} - \Theta(X, Y) \xi'_x \right\|_2 + \alpha \|\xi'_x\|_2 \\ \xi_y = \arg \min_{\xi'_y} \left\| \frac{\partial Y}{\partial t} - \Theta(X, Y) \xi'_y \right\|_2 + \alpha \|\xi'_y\|_2 \end{cases} \quad (4.10)$$

As in the previous case there is a high dependency on noise given by the numerical computation of finite derivatives. And accordingly with the case of simulated Turing patterns even if we try to compute the derivative in more-noise robust method the performance of the algorithm does not improve compared to the finite differences scheme. Even though with this dataset the algorithm performs more poorly in terms of correct identification of the coefficients in the reaction terms, we have to appreciate the noise robustness of the approximate solutions and the correct choice of the polynomial terms in the library of functions Θ .

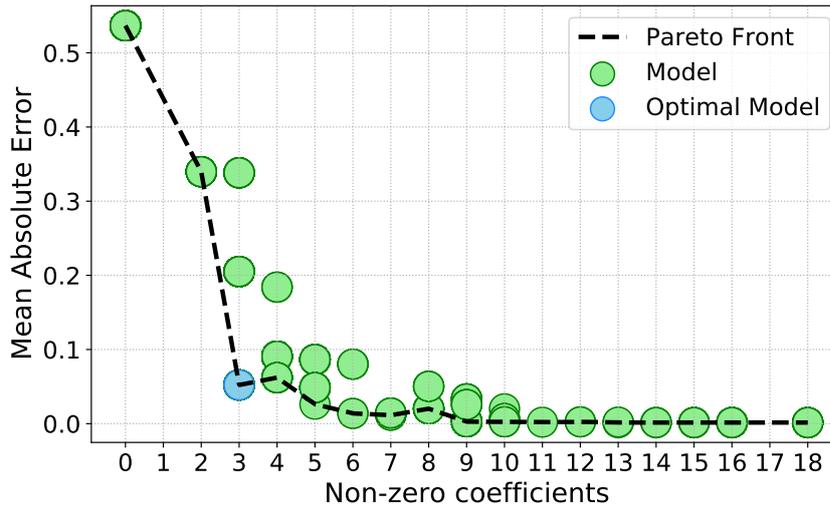


FIGURE 4.4: Each point in the graph is an output model for specific hyper-parameters, i.e. the threshold and the regularization strength. For each model, we plot its error and its complexity measured in the number of non-zero coefficients of the sparse vector. It is clear that there is a Pareto optimal solution that represents the best choice for the hyper-parameters. The plot is referred to a concentration X .

Given the interpretation of under- and over-fitting as respectively high error and low number of terms in the equation and low error and high number of terms, we can pick the best model hyper-parameters of the model as done previously (Fig. 4.4). The optimal model, which is the one with the lowest sparsity and lowest error (Fig. 4.4) does identify the correct library terms but does not manage to detect the diffusion terms. The following table reports the results of the algorithm as noise increases.

Noise	Chemical	α	Threshold	Identified Equation
0%	X	10^{-5}	0.2	$0.899 - 3.176x + 0.990x^2y$
	Y	10^{-5}	0.1	$2.232x - 0.981x^2y$
0.001%	X	10^{-5}	0.2	$0.899 - 3.176x + 0.990x^2y$
	Y	10^{-5}	0.1	$2.232x - 0.981x^2y$
0.01%	X	10^{-5}	0.2	$0.899 - 3.176x + 0.990x^2y$
	Y	10^{-5}	0.1	$2.232x - 0.981x^2y$
0.1%	X	10^{-5}	4.72	$0.899 - 3.176x + 0.990x^2y$
	Y	10^{-4}	0.2	$2.232x - 0.981x^2y$
0.5%	X	5×10^{-3}	0.1	$0.898 - 3.175x + 0.990x^2y$
	Y	10^{-2}	0.6	$2.232x - 0.981x^2y$
1%	X	0.5	6.7	$0.898 - 3.174x + 0.990x^2y$
	Y	0.5	0.5	$2.231x - 0.981x^2y$
5%	X	0.6464	10^{-5}	$0.887 - 3.136x + 0.977x^2y$
	Y	0.608	0.1	$2.207x - 0.970x^2y$
10%	X	0.9696	10^{-5}	$3.363 - 7.695x - 1.111y + 1.857x^2 + 2.198xy$
	Y	0.1	0.8	$4.545x - 1.723x^2 - 1.143xy$

TABLE 4.2: In the table are listed the results of the best model for different noise levels. If the noise is small enough, the algorithm correctly identifies the reaction terms of the reaction-diffusion system. The noise is the reported percentage of the standard deviation of the noise-free simulations.

Noise	Chemical	α	Threshold	Second Best Sparse Model
0%	X	10^{-5}	0.2	$-3.117x^2 + 0.867x^3 + 0.988x^2y + 0.806\nabla^2x$
	Y	10^{-5}	0.1	$2.234x - 0.982x^2y - 1.208\nabla^2y$
0.001%	X	10^{-5}	0.2	$-3.117x^2 + 0.867x^3 + 0.988x^2y + 0.805\nabla^2x$
	Y	10^{-5}	0.1	$2.234x - 0.982x^2y - 1.207\nabla^2y$
0.01%	X	10^{-5}	0.2	$-3.117x^2 + 0.867x^3 + 0.988x^2y + 0.801\nabla^2x$
	Y	10^{-5}	0.1	$2.234x - 0.982x^2y - 1.195\nabla^2y$
0.1%	X	10^{-5}	4.72	$-3.117x^2 + 0.867x^3 + 0.988x^2y + 0.787\nabla^2x$
	Y	10^{-4}	0.2	$2.234x - 0.982x^2y - 1.159\nabla^2y$
0.5%	X	5×10^{-3}	0.1	$-3.115x^2 + 0.866x^3 + 0.988x^2y + 0.542\nabla^2x$
	Y	10^{-2}	0.6	$2.233x - 0.981x^2y - 0.657\nabla^2y$

TABLE 4.3: In the table are listed the results of the second-best model for different noise levels. As we can notice the algorithm correctly selects the Laplacian to include in the equation.

Selecting the second-best sparse model enables the algorithm to successfully identify the presence of diffusion. It's noteworthy that even in the presence of minimal noise, the algorithm can detect the existence of diffusion. In the case of chemical species X , the identified diffusion constant aligns closely with the actual value (Eq. (4.7)). Similarly, for chemical species Y , the algorithm identifies diffusion, albeit with an inaccurate diffusion constant. Importantly, in the case of chemical species Y , the second-best optimal solution accurately selects all the correct function library terms, providing a robust approximation of the coefficients for the reaction terms. With the current data, the algorithm is more robust to noise compared to the Turing patterns data. The cause can be identified in the relative abundance of data points compared with the stationary case. Moreover, the inability of the algorithm to compute the correct diffusion term is once again linked to noisy derivatives. The higher the order of the derivative of a noise data, the higher will be also the noise of the derivative.

4.3 Drosophila Turing Patterns

Turing patterns are self-organizing order structures that we can see regularly in the natural world. In this section we comment on the trial to calibrate experimental data coming from Turing patterns arising in an embryo of the drosophila. In this precocious state of development, it is reasonable to think that pattern formation is driven by protein production and interaction. We apply the sparse regression framework of laboratory data of two proteins concentrations (Fig. 4.5).

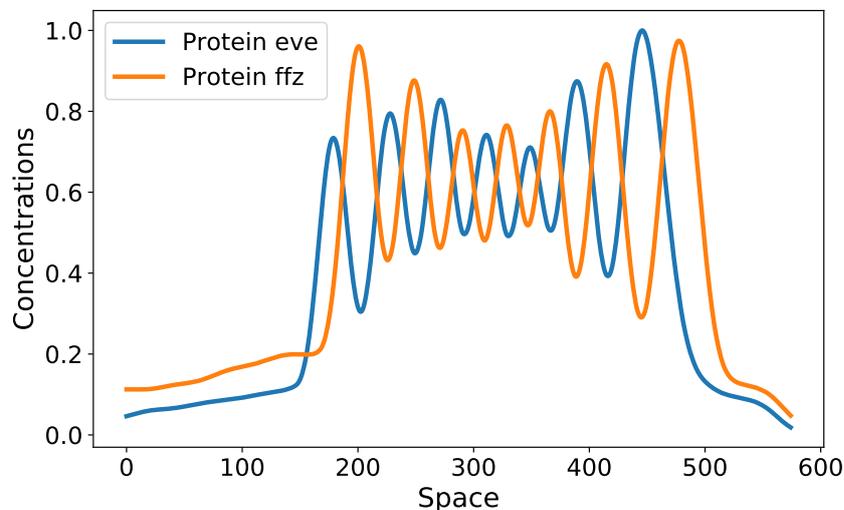


FIGURE 4.5: The picture shows the distribution of two proteins in the embryo of a fruit fly.

As done in the first section of this chapter we compute the discrete Laplacian through finite differences avoiding the boundary regions where the pattern is not present. It is already clear from the form of the Laplacian in Fig. 4.6 that there is a high presence of noise. Nonetheless, we want to investigate if the magnitude of the noise is too high for our model to correctly identify the dynamical equations. In particular, as previously done on synthetic data of the Turing pattern coming from the brusselator model, we want to apply the sparse regression framework to identify the reaction term for the two proteins.

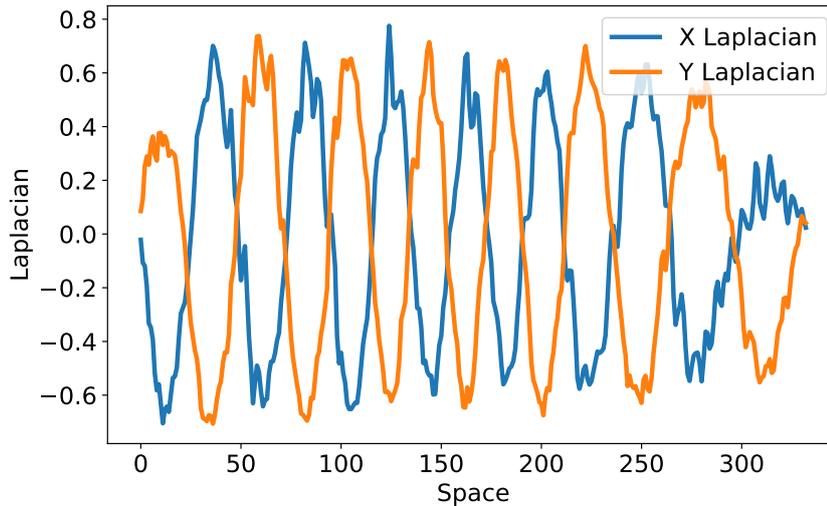


FIGURE 4.6: The picture shows the Laplacian of the distribution of two proteins in the embryo of a fruit fly.

Given the interpretation of under- and over-fitting as respectively high error and low number of terms in the equation and low error and high number of terms, we can pick the best model hyper-parameters of the model as done previously (Fig. 4.7).

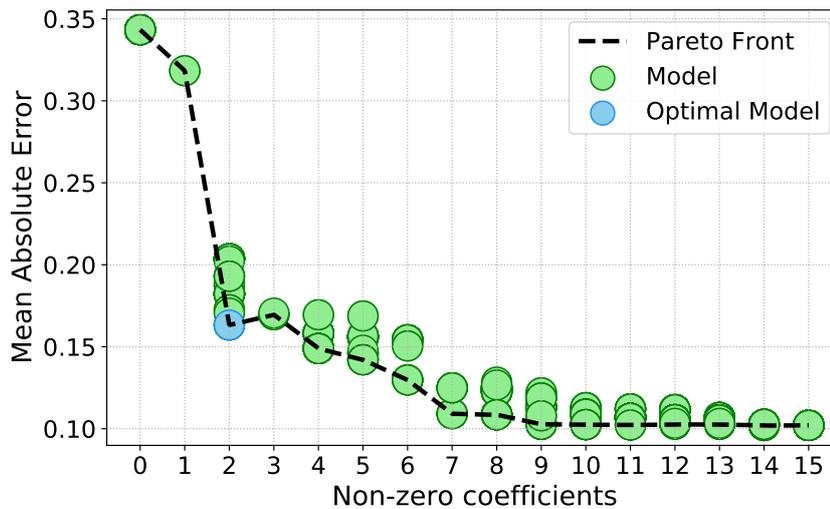


FIGURE 4.7: The plot shows the optimal and most sparse model for the protein dataset of the drosophila.

We apply the algorithm to this data finding that noise is indeed too high to have a precise or reasonable approximation of the reaction terms. This fact can be noticed by looking at the prediction of the fit in Fig. 4.8

The optimal model we found for the real system is presented in the following equation, which is sparse.

$$\begin{cases} \nabla^2 X &= -\frac{R_x(X,Y)}{D_1} = -3.686XY + 5.606XY^2; \\ \nabla^2 Y &= -\frac{R_y(X,Y)}{D_2} = 2.353X^2Y - 3.408XY^3. \end{cases} \quad (4.11)$$

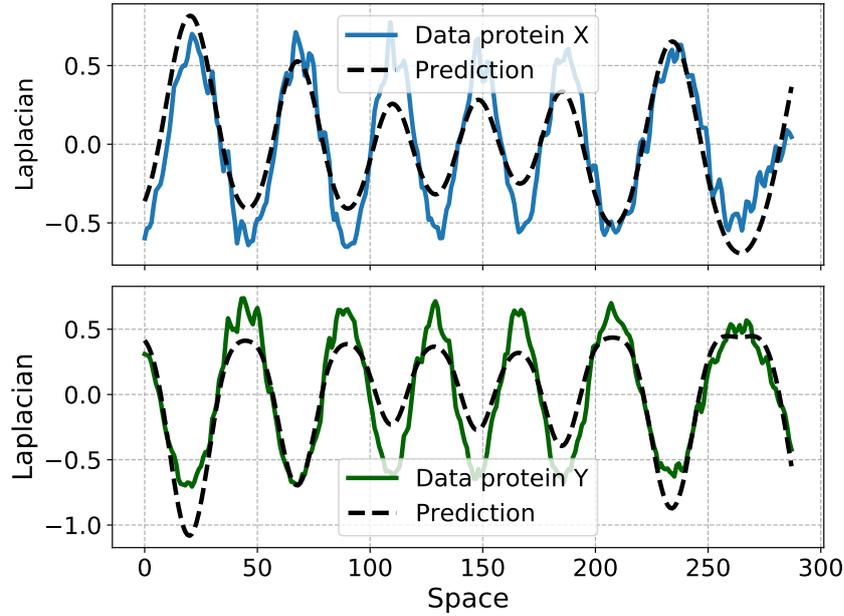


FIGURE 4.8: The plot shows the best model fit (dotted lines) and the Laplacian of the protein concentrations of the drosophila.

The system of equations considering just the reaction term is the following is the following

$$\begin{cases} \frac{\partial X}{\partial t} = AX + BXY^2 \\ \frac{\partial Y}{\partial t} = CX^2Y + DXY^3 \end{cases} \quad (4.12)$$

where $A = 3.686$, $B = -5.606$, $C = -2.353$, $D = 3.408$. This system has a fixed point in $(X_0, Y_0) = (-\frac{A^2D}{B^2C}, -\frac{A}{B})$. The Jacobian evaluated in (X_0, Y_0) has real eigenvalues $\lambda_1 = -0.834$ and $\lambda_2 = 2.675$. So it is an unstable fixed point but the behaviour of the system with diffusion might vary.

When fitting the Laplacian, the diffusion constant emerges as a scaling factor accompanying the coefficients of the reaction term, as outlined in the equation (4.12). When we standardize the diffusion constant to 1 and use the scalar coefficients, derived from the sparse regression model, we essentially re-scale the time. Although this is not a problem for a single system, where the actual dynamics remain consistent, it becomes a crucial consideration for two coupled differential equations. In this context, uniform scaling of the system is essential. The maximum diffusion constant between the two chemicals sets the spatial and temporal scales. Accordingly, we set one diffusion constant to 1, assuming it is the maximum, and conduct a grid search for the other within the range $(0, 1)$. Next, after determining the diffusion constants, we numerically integrate the reaction-diffusion equations derived from (4.12) including the diffusion terms, looking for the steady state of the system.

After this analysis we could not see any Turing pattern structure in the stationary state of the systems with various diffusion constants. Already from the approximation of the computed discrete Laplacian (Fig. 4.6), and even more in the prediction of the Laplacian (Fig.4.8) we can see that the noise level is high. The prediction of the Laplacian manages to find the correct period of the periodic signal but performs poorly in identifying peaks and wells. So even though the spatial structure of the

protein distribution looks pretty similar to the one of Turing pattern originated from reaction-diffusion we could not calibrate the experiment and re-propose the same results in a simulated system.

4.4 Remarks

The analysis of the algorithm reveals its potential in identifying terms associated with local interactions between reactants, even in the presence of moderate noise. However, to enhance its performance, as we stated in the preceding paragraphs, further development of denoising or derivative computation methods is necessary, distinct from the ones attempted in this thesis.

A key observation is the potential relationship between the algorithm's resilience to noise and the number of points considered for the fit. As we saw with the two different dataset, the noise robustness of the algorithm varies. To investigate this we could vary the sample abundance to discern if the accuracy of the model is significantly affected by the number of points. We expect the model to depend highly on data points number in low sample abundance and we hypothesis the presence of a number of sampling point after which the performance does not rise and remains constant. This analysis could be enriched by assessing the algorithm's efficiency across different scenarios, manipulating both the number of points and the intensity of noise. The expectation is that as noise increases, more points are required to prevent model over-fitting.

Concerning optimization, the ridge regression method has been consistently employed. This method allows for the exploration of parameter space, providing a straightforward means to select the optimal model by plotting model error against the number of terms present in the models. Exploratory attempts with a relaxed version of regression, incorporating l1 and l0 norm regularization, were made to enhance performance in noisy conditions. However, the outcome was unfavorable, leading to the continuation of the standard ridge regression method. The algorithm's effectiveness depends on various factors, including the ad hoc choices of regularization and derivative calculation methods, which may be system-dependent.

Investigating the hypotheses we are proposing is beyond the scope of the thesis. We place emphasis on the potential insights that could be gained from a deeper comprehension of the algorithm's behavior, in particular in our case where we are interested in a specific kind of models and equations, namely reaction-diffusion.

Chapter 5

Conclusions

5.1 Synthetic Turing Patterns Data

In this study, we applied a sparse regression framework to analyze synthetic Turing patterns generated by the Brussellator within the Turing patterns regime. The algorithm aimed to identify the coefficients of the reaction-diffusion equations based on stationary state data. We observed that the success of the algorithm is highly sensitive to the choice of parameters, such as the regularization strength (α) and the threshold for sparsity. Exploring the delicate balance between model error and complexity was crucial to find the optimal model, as illustrated in the complexity-error tradeoff (Fig. 4.2).

The grid search for optimal parameters demonstrated the importance of fine-tuning to achieve an equilibrium, ensuring that the sparse vectors contain a minimal number of nonzero terms. However, the algorithm's performance was affected by noise in the Laplacian computation, particularly in the calculation of derivatives using finite differences.

To address noise sensitivity, we explored alternative methods for calculating derivatives, such as polynomial interpolation, total variation denoising, and smoothed finite differences. However, these approaches did not outperform the finite differences scheme.

Additionally, we investigated the Sparse Relaxed Regularized Regression (SR3) algorithm as an alternative to mitigate noise impact. Despite implementation, the performance improvement was marginal, and the initial Sequential Thresholded Ridge Regression remained the more efficient choice.

The algorithm showed promising results in identifying coefficients in the presence of low to moderate noise levels. However, the sensitivity to noise, especially in derivative calculations, remains a limitation that requires further consideration and refinement.

5.2 Synthetic Oscillations Data

Extending the analysis to synthetic oscillation data, we explored the algorithm's performance in identifying coefficients for reaction-diffusion equations inducing sustained oscillations. Similar challenges with noise sensitivity were encountered, and the impact of noise on derivative calculations persisted. Nevertheless, the algorithm demonstrated better robustness to noise compared to the Turing patterns data, likely due to the relative abundance of data points.

The algorithm successfully identified the presence of diffusion in the second-best sparse model, even with minimal noise, although the accuracy of diffusion constant

estimation varied. The optimal model proved crucial in accurately selecting function library terms, providing a robust approximation of coefficients for reaction terms.

The limitations associated with noise in derivative calculations remained, emphasizing the need for further improvements and considerations when dealing with different regimes and noise levels.

5.3 *Drosophila* Turing Patterns

Applying the sparse regression framework to experimental data from Turing patterns in *Drosophila* embryos posed additional challenges. The Laplacian computation revealed a high presence of noise, impacting the algorithm's ability to accurately identify reaction terms. Despite attempts to explore alternative methods and algorithms, the noise level proved too high for precise approximation.

The analysis also revealed challenges related to limited data availability, emphasizing the need for nuanced modeling approaches for different regimes.

In conclusion, while the sparse regression framework shows promise in identifying reaction-diffusion coefficients, addressing noise sensitivity and adapting to different regimes are essential for robust and reliable results.

5.4 Further Developments

The presented research opens avenues for future exploration and refinement of the sparse regression framework for reaction-diffusion systems:

- **Noise Robustness:** Develop and test additional noise-robust methods for derivative calculations to enhance the algorithm's performance in the presence of high noise levels.
- **Data Augmentation:** Investigate techniques for data augmentation or collection strategies to mitigate challenges associated with limited data availability, especially in experimental settings.
- **Parameter Optimization:** Explore advanced optimization techniques to automate the process of selecting optimal hyper-parameters, reducing the need for manual grid search.
- **Real-world Applications:** Apply the refined framework to real-world biological systems beyond synthetic data, considering the intricacies of experimental noise and complexities of biological processes. In the context of the reaction-diffusion system, a challenging problem would be to calibrate the Belousov-Zhabotinsky reaction from laboratory data which up until today remains an open problem.

By addressing these future directions, we aim to enhance the robustness, accuracy, and applicability of the sparse regression framework, contributing to a deeper understanding of complex biological phenomena governed by reaction-diffusion dynamics.

Bibliography

- [1] Joseph Bakarji et al. “Discovering governing equations from partial measurements with deep delay autoencoders”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 479.2276 (2023), p. 20230422. DOI: [10.1098/rspa.2023.0422](https://doi.org/10.1098/rspa.2023.0422).
- [2] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. “Discovering governing equations from data by sparse identification of nonlinear dynamical systems”. In: *Proceedings of the National Academy of Sciences* 113.15 (2016), pp. 3932–3937. DOI: [10.1073/pnas.1517384113](https://doi.org/10.1073/pnas.1517384113).
- [3] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. “Sparse Identification of Nonlinear Dynamics with Control (SINDYc)”. In: *IFAC-PapersOnLine* 49.18 (2016). 10th IFAC Symposium on Nonlinear Control Systems NOLCOS 2016, pp. 710–715. ISSN: 2405-8963. DOI: <https://doi.org/10.1016/j.ifacol.2016.10.249>.
- [4] V. Castets et al. “Experimental evidence of a sustained standing Turing-type nonequilibrium chemical pattern”. In: *Phys. Rev. Lett.* 64 (24 June 1990), pp. 2953–2956. DOI: [10.1103/PhysRevLett.64.2953](https://doi.org/10.1103/PhysRevLett.64.2953).
- [5] Rick Chartrand. “Numerical Differentiation of Noisy, Nonsmooth Data”. In: *ISRN Appl. Math.* 2011 (Jan. 2011). DOI: [10.5402/2011/164564](https://doi.org/10.5402/2011/164564).
- [6] Rui Dilão. “Turing instabilities and patterns near a Hopf bifurcation”. In: *Applied Mathematics and Computation* 164.2 (2005). 12th International Workshop on Dynamics and Control, pp. 391–414. ISSN: 0096-3003. DOI: <https://doi.org/10.1016/j.amc.2004.06.036>. URL: <https://www.sciencedirect.com/science/article/pii/S0096300304003261>.
- [7] Daniel R. Gurevich, Patrick A. K. Reinbold, and Roman O. Grigoriev. “Robust and optimal sparse regression for nonlinear PDE models”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.10 (Oct. 2019), p. 103113. ISSN: 1054-1500. DOI: [10.1063/1.5120861](https://doi.org/10.1063/1.5120861).
- [8] Hermann Haken. *Synergetics. Introduction and Advanced Topics*. 1st ed. Originally published in 2 volumes as volume 1 in the series “Springer Series in Synergetics”. Springer Berlin, Heidelberg, 2004, pp. XV, 764. DOI: [10.1007/978-3-662-10184-1](https://doi.org/10.1007/978-3-662-10184-1).
- [9] Joshua S North, Christopher K Wikle, and Erin M Schliep. “A Review of Data-Driven Discovery for Dynamic Systems”. In: *International Statistical Review* (2023).
- [10] Joshua S. North, Christopher K. Wikle, and Erin M. Schliep. “A Review of Data-Driven Discovery for Dynamic Systems”. In: *International Statistical Review* n/a.n/a (). DOI: <https://doi.org/10.1111/insr.12554>.
- [11] I. Prigogine and R. Lefever. “Symmetry Breaking Instabilities in Dissipative Systems. II”. In: *The Journal of Chemical Physics* 48.4 (Sept. 2003), pp. 1695–1700. ISSN: 0021-9606. DOI: [10.1063/1.1668896](https://doi.org/10.1063/1.1668896).

- [12] Chengping Rao et al. "Encoding physics to learn reaction–diffusion processes". In: *Nature Machine Intelligence* 5.7 (July 2023), pp. 765–779. DOI: [10.1038/s42256-023-00685-7](https://doi.org/10.1038/s42256-023-00685-7).
- [13] Patrick A. K. Reinbold and Roman O. Grigoriev. "Data-driven discovery of partial differential equation models with latent variables". In: *Phys. Rev. E* 100 (2 Aug. 2019), p. 022219. DOI: [10.1103/PhysRevE.100.022219](https://doi.org/10.1103/PhysRevE.100.022219).
- [14] Samuel Rudy et al. "Data-Driven Identification of Parametric Partial Differential Equations". In: *SIAM Journal on Applied Dynamical Systems* 18.2 (2019), pp. 643–660. DOI: [10.1137/18M1191944](https://doi.org/10.1137/18M1191944).
- [15] Samuel H. Rudy et al. "Data-driven discovery of partial differential equations". In: *Science Advances* 3.4 (2017), e1602614. DOI: [10.1126/sciadv.1602614](https://doi.org/10.1126/sciadv.1602614). URL: <https://www.science.org/doi/abs/10.1126/sciadv.1602614>.
- [16] Steven H Strogatz. *Nonlinear dynamics and chaos with student solutions manual: With applications to physics, biology, chemistry, and engineering*. CRC press, 2018.
- [17] Robert Tibshirani. "Regression Shrinkage and Selection Via the Lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288. DOI: <https://doi.org/10.1111/j.2517-6161.1996.tb02080.x>.
- [18] Alan Mathison Turing. "The chemical basis of morphogenesis". In: *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 237.641 (1952), pp. 37–72. DOI: [10.1098/rstb.1952.0012](https://doi.org/10.1098/rstb.1952.0012).
- [19] Insoon Yang and Claire J. Tomlin. "Reaction–diffusion systems in protein networks: Global existence and identification". In: *Systems Control Letters* 74 (2014), pp. 50–57. ISSN: 0167-6911. DOI: <https://doi.org/10.1016/j.sysconle.2014.09.013>. URL: <https://www.sciencedirect.com/science/article/pii/S0167691114002102>.
- [20] A. N. Zaikin and A. M. Zhabotinsky. "Concentration Wave Propagation in Two-dimensional Liquid-phase Self-oscillating System". In: *Nature* 225.5232 (Feb. 1970), pp. 535–537. DOI: [10.1038/225535b0](https://doi.org/10.1038/225535b0).
- [21] Peng Zheng et al. "A Unified Framework for Sparse Relaxed Regularized Regression: SR3". In: *IEEE Access* 7 (2019), pp. 1404–1423. DOI: [10.1109/ACCESS.2018.2886528](https://doi.org/10.1109/ACCESS.2018.2886528).