Data-driven discovery of the mechanism of the Belousov-Zhabotinsky reaction

Mariana Mota
mariana.mota@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

December 2020

Abstract

In this work, we deduced the mechanism associated with reaction-diffusion waves simulated by the Brusselator, an oscillating chemical reaction system which virtually describes the dynamics of the Belousov-Zhabotinsky reaction. This is done using a recently developed algorithm that aims to derive governing equations from time-series data, the PDE-FIND algorithm. The Brusselator system is analysed, by studying its stability, to better understand its underlying dynamics, as well as simulating typical reaction-diffusion patterns with this model. We reconstructed the phase space using the time evolution data of only one of the two variables of the system. The PDE-FIND algorithm is described and tested, and finally applied to the computed data of the Brusselator system, inferring the reaction-diffusion system that best fits it. The PDE-FIND algorithm showed to be able to correctly identify the system of partial differential equations, which shows that it is possible to obtain valuable information of the local dynamics of a system from data, and thus to infer its underlying kinetic mechanisms.

Keywords: Belousov-Zhabotinsky, reaction-diffusion, Brusselator, algorithm, data mining

1. Introduction

Extracting the governing equations of a system from temporal and spatial data in order to discover its underlying dynamics is a major challenge in diverse areas of science and engineering. In the last decade, data-driven discovery methods have been made possible due to the rapid decrease of the cost of sensors, data storage, and computational resources. Advances in machine learning and data science have made it possible to extract patterns from large sets of data, a breakthrough in the analysis and understanding of complex data. Typically, physical systems' underlying partial differential equations (PDEs) are derived from conservation laws, physical principles, and phenomenological behaviour. However, the development of a new method for deriving underlying PDEs of dynamic processes from big data is essential, since there are still complex systems that escape from quantitative analytic descriptions.

Recently, S. Rudy, S. Brunton, J. Proctor and J. Kutz [1] have developed deep learning techniques to fit observed data, with models based on time series measurements in the spatial domain.

The main goal of this work is to deduce the kinetic mechanism associated with reaction-diffusion waves observed in the Belousov-Zhabotinsky reaction, virtually simulated and described by the Brusselator model. Resorting to the algorithm developed by Rudy and co-authors [1], the collected data will be fitted to a reaction-diffusion model and the corresponding kinetic equations, testing the accuracy and applicability of the model.

1.1. Reaction-Diffusion systems

The shape or pattern of a natural system results from its symmetry or regularity, as well as the frequency with which it is observed in nature. Finding the mechanisms that generate the biological patterns, as well as the reason why some shapes are more abundant than others becomes an important task [2]. In 1952, Turing [3] proposed that these real systems present self-organizing properties which appear in nature as coherent patterns or structures. In the presence of reactive processes, the effect of diffusion could be compensated by local chemical processes, and the reaction between two molecules could amplify local fluctuations to a macroscopic scale, leading to these patterns. Properties of reaction-diffusion systems depend on the balance between chemical and diffusive processes.

A reaction-diffusion system can be defined as a system of partial differential equations of the form:

$$\frac{\partial \varphi}{\partial t} = f(\varphi) + D \cdot \nabla^2 \varphi \tag{1}$$

where $\varphi = (\varphi_1, ..., \varphi_m)$ is a m-dimensional vector representing the concentration of m chemical species, $f(\varphi)$ is a m-dimensional vector field representing the local kinetic mechanism and D is the diffusion matrix. An equation of the type 1 is defined for each one of the dynamical variables of a model. The most impactful and investigated reaction-diffusion model is the Belousov-Zhabotinsky reaction, where oscillating reactions are experimentally observed.

1.2. Belousov-Zhabotinsky Reaction

The Belousov-Zhabotinsky (BZ) reaction is named after B. P. Belousov who discovered the reaction and A. M. Zhabotinsky who continued Belousov's early work [4]. This reaction serves as a classical example of non-equilibrium thermodynamics [5]. It is able to maintain a prolonged state of non-equilibrium that leads to macroscopic temporal oscillations and spatial pattern formation that is very life-like. It makes it possible to observe development of complex patterns in time and space by naked eye [6]. The BZ reaction allows the study of chemical waves and patterns without constant replenishment of reactants, by generating up to several thousand oscillatory cycles in a closed system.

The reaction was first discovered by Belousov, who found that a mixture of chemical species (citric acid, bromate, and cerium catalyst in a sulfuric acid solution) underwent periodic color changes between colourless and yellow [7]. These colour changes indicated the cyclic formation and depletion of differently oxidized cerium species [6]. Later, Zhabotinsky reproduced these results with a different reductant, malonic acid, and showed that oscillations in concentration of ceric ions (Ce^{4+}) lead to the oscillations in the solution's colour [8]. These oscillation are represented in Figure 1. The yellow colour was found to be due to the preponderance of Ce^{4+} ions while the colourless state is due to the cerous ions (Ce^{3+}). He proposed that the BZ reaction consists of two main parts: the autocatalytic oxidation of Ce^{3+} ions by $HBrO_3$ and the reduction of Ce^{4+} ions by malonic acid, which were produced during the overall reaction [8]. The Ce^{4+} reduction is accompanied by the production of Br^- from the bromoderatives of malonic acid.

The main attribute of the BZ reaction in homogeneous media is the induction of periodic fluctuations in the concentrations of the intermediates. In non-homogeneous conditions, local fluctuations in the concentration of reagents are transmitted to the reactor region, giving rise to wave fronts that propagate and interact with each other.

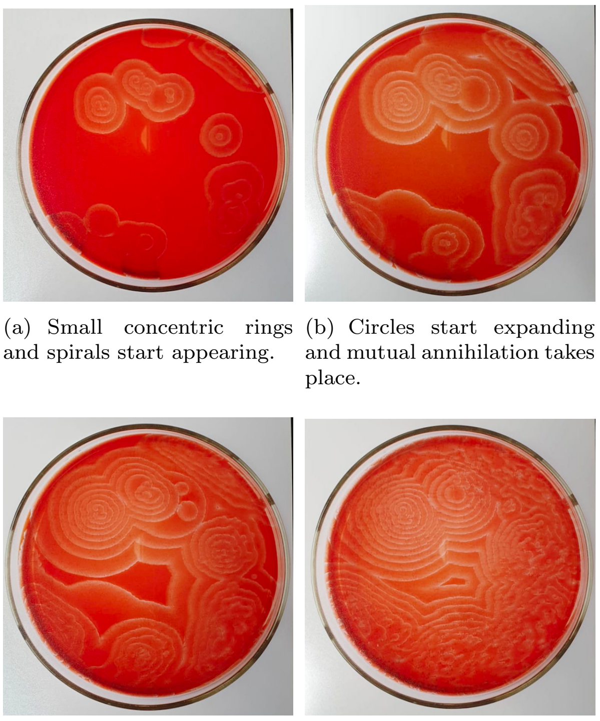

(a) Small concentric rings and spirals start appearing.
(b) Circles start expanding and mutual annihilation takes place.
(c) System keeps expanding.
(d) Strong annihilation results in the loss of circular waves.

Figure 1: Target patterns in the BZ reaction, formed by point pacemakers. The waves emerge from a background of a reduced state (red) in which concentration waves of the autocatalytic species of the reaction are defined, whose production is coupled with the oxidation of ferroin in ferrin (blue).

Zhabotinsky described the propagation of waves of oxidation in thin unstirred layers of BZ reagent, organized as concentric rings that expanded away from a central zone of periodic initiation [4]. These concentric chemical waves generated by point pacemakers formed target patterns, which vary in temporal period, and are composed of pulses of excitation followed by refractory zones. Collisions of the waves lead to mutual annihilation due to the presence of non-excitable refractory zones [6]. If these waves break, the excitation front curls around their refractory tails and form spiral waves. The formed target patterns can be seen in Figure 1. These waves emerge from a background of a reduced state (red) in which concentration waves of $HBrO_2$, the autocatalytic species of the reaction, are defined, whose production is coupled with the oxidation of ferroin in ferrin (blue).

2. Brusselator Model

The Brusselator is a virtual oscillating chemical reaction system that has been used to describe the spatial dynamics of the Belousov-Zhabotinsky reaction. The model was proposed by Prigogine and

Lefever (1968) and was given its name as a reference to its birthplace (Université Libre de Bruxelles). Its theoretical simplicity and the fact that it retains the functional form of more complex reaction networks makes it a widely used model.

The Brusselator reaction consists of four steps:

$$\mathrm{A} \xrightarrow{k_1} \mathrm{X} \tag{2.2a}$$

$$\mathrm{B} + \mathrm{X} \xrightarrow{k_2} \mathrm{Y} + \mathrm{D} \tag{2.2b}$$

$$2\,\mathrm{X} + \mathrm{Y} \xrightarrow{k_3} 3\,\mathrm{X} \tag{2.2c}$$

$$\mathrm{X} \xrightarrow{k_4} \mathrm{E} \tag{2.2d}$$

where X and Y represent the dynamic variables of the system, A and B are control variables (which are kept constant) and $k_i, i = 1,2,3,4$ represent the reaction rates. The third step is autocatalytic since two X molecules make three, and also has an inhibiting factor because Y is used in this process while it is necessary to make the reaction. The reaction scheme is physically unrealistic because of the trimolecular third step, since this reaction is statistically unlikely [9]. However, systems with two dynamical variables can only show limit cycle variations if the kinetic mechanism includes a trimolecular term. Since the reactions are all irreversible, the output variables are irrelevant for the mechanism.

Assuming a two-dimensional media, the following system of differential equations are used to describe the Brusselator model [10]:

$$\frac{\partial X}{\partial t} = k_1 A - k_2 B X + k_3 X^2 Y - k_4 X + D_X \left(\frac{\partial^2 X}{\partial x^2} + \frac{\partial^2 X}{\partial y^2} \right) \tag{2.3}$$

$$\frac{\partial Y}{\partial t} = k_2 B X - k_3 X^2 Y + D_Y \left(\frac{\partial^2 Y}{\partial x^2} + \frac{\partial^2 Y}{\partial y^2} \right) \tag{2.4}$$

The system composed by equations (2.3, 2.4) has an equilibrium unstable solution for $X = A(k_1/k_4)$ and $Y = (B/A)(k_2k_4/k_1k_3)$ and its Hopf bifurcation occurs for $B = (k_4/k_2) + A^2(k_1^2 k3/k_2 k_4^2)$. This system of partial differential equations is reduced to a system of ordinary differential equations when there is suppressed diffusion, and we set $D_X = D_Y = 0$. With these conditions, it is possible to follow the time evolution of X and Y in the concentration space and analyse the temporal evolution of the system.

The reference conditions of this work are: $k_1 = k_2 = k_3 = k_4 = 1$, $A = 1.0$, $B = 2.3$. For an non-diffusive media, the Brusselator's model phase space is shown in Figure 2. Since $B > 2.0$, the fixed point is an unstable focus.

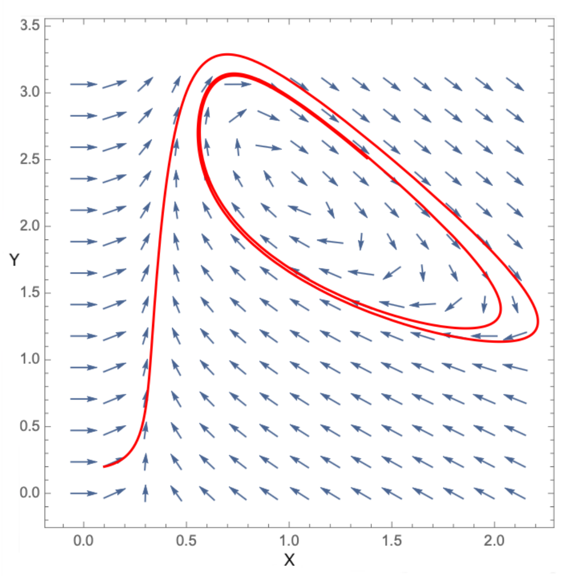


Figure 2: Brusselator's phase space on a non-diffusive media for reference conditions of this work: $A = 1.0$, $B = 2.3$ and $k_i = 1.0$. The phase space orbit converges to a limit cycle, and the fixed point is an unstable focus since $B > 2.0$.

2.1. Phase-Space Reconstruction

For a mathematically modelled system, like the Brusselator, the phase space is known from the equations of motion [11]. However, for experimental and naturally occurring chaotic dynamical systems, the phase space and a mathematical description of the system are often unknown. Usually, the number of dynamical variables available from a given system is restricted [12]. This led Takens to introduce an algorithm in order to reconstruct an attractor with only the information of one of the state variables.

According to Takens delay embedding theorem [13], from a single coordinate of a dynamic system in N dimension, measured at a certain value of the control parameters, the signal can be embedded into a higher-dimensional phase space if appropriate values of time delay (Δt) and embedding dimension (D) are selected [14, 15]. With this, a time series $\{X_i\}_{i=1}^{n}$ can be defined:

$$X_i = (x(i), x(i+\Delta t), x(i+2\Delta t), \ldots, x(i+(D-1)\Delta t)) \tag{2.5}$$

which represents the constructed delay-coordinate vectors, and where $i = 1, 2, 3, \ldots, N - (D-1)\Delta t$. It is possible to obtain an approximate image of the dynamics of the attractor, given a dynamic system where only one time series from a sample of a single state variable is known. The plot between the elements of the vectors (2.5) shows the evolution of the dynamics of the system in D-dimensional phase space.

Using the Takens' technique, we intend to reconstruct the phase space of the original system, with only the information of the X variable. Selecting a dimension of $D = 2$ and a time delay of $\Delta t = 7$, the same reference conditions as the system in Figure 2 are applied to obtain the reconstructed phase space in Figure 3.

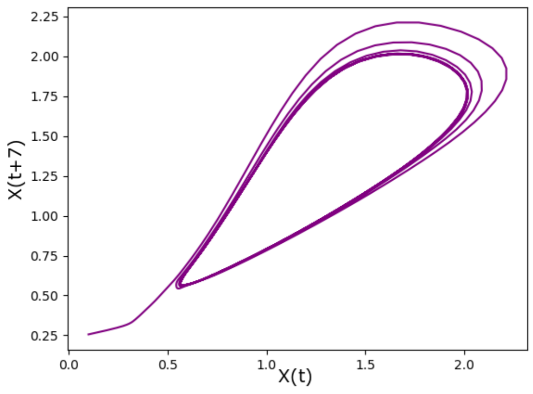


Figure 3: Brusselator's phase space reconstruction with the Takens' technique for reference conditions of this work: $A = 1.0$, $B = 2.3$ and $k_i = 1.0$, and reconstruction conditions of $D = 2$ and $\Delta t = 7$. The phase space of the original system is reconstructed with only the information of the autocatalytic variable X. It shows the same dynamic behaviour and follows the same shape as the original Brusselator phase space portrait.

The reconstructed phase space has the same shape and shows the same dynamic behaviour as the original phase space. In the BZ reaction, only a single diffusive variable is observed (the autocatalytic variable X), while variable Y is, in principle, not diffusive, but a control variable.

2.2. Reaction-Diffusion Pattern Simulation

Patterns that appear experimentally through the BZ reaction are simulated with the 2D Brusselator model, which generates both concentric rings and spiral waves. In this work, only circular waves are simulated.

The choice of parametrization reflected on simplicity of the algebraic equations and computational efficiency. This way, the values for the velocity constants of the model (2.3, 2.4) were set to $k_1 = k_2 = k_3 = k_4 = 1.0$, and the control variables took the numerical values $A = 1.0$ and $B = 2.3$. Only the autocatalytic variable's diffusion coefficient was taken into account ($D_X = 1.0$), and the inhibiting variable is studied with no associated diffusion ($D_Y = 0.0$). The initial distribution of the chemical species concentrations is given by the steady state condition of the model: $X^* = 1.0 \quad \text{and} \quad Y^* = 2.3$

The most frequent pattern in the BZ reaction is the circular wave pattern, which is characterized by a constant propagation speed. To simulate such waves, one can simply apply an infinitesimal perturbation to a point in a still, homogeneous media, defined by steady state conditions. Since the steady state is unstable, the perturbation evolves to an oscillating state and its effects propagate to the revolving space [16]. The waves were initiated with a simple perturbation around the steady state in the central point of the square grid of size $M \times M$:

$$X[M/2][M/2] = X^*_{ss} + 1.0 \qquad (2.6a)$$

$$Y[M/2][M/2] = Y^*_{ss} + 1.0 \qquad (2.6b)$$

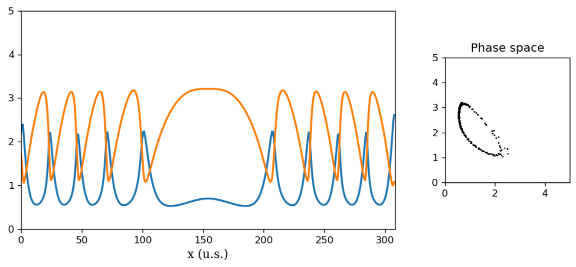


Figure 4: 1D dynamical patterns for a time of evolution of *200 u.t.*, with diffusion only on the autocatalytic variable ($D_X = 1.0$, $D_Y = 0.0$) and for reference conditions of this work. Periodic patterns are formed when the simulation starts with a symmetric perturbation. In the phase space, the limit cycle has a similar trajectory to that of the homogeneous system.

When the simulation is started by a symmetric perturbation, periodic patterns are formed, as shown in Figure 4. Although it is important to understand the system's dynamics in one dimension, the reaction-diffusion patterns are better experimentally documented in two dimensional spaces. Figure 5 shows the time frames of 25, 75 and 150 u.t. when simulating the Brusselator in a 2D media. The chosen colour code influences the way the simulations are presented. The evolution of the X substance is documented with a Red-White colour code, and the Y substance is shown in a Blue-White colour code. To simulate the experimental formation of these waves, both canals are merged, forming a Red-Blue output.

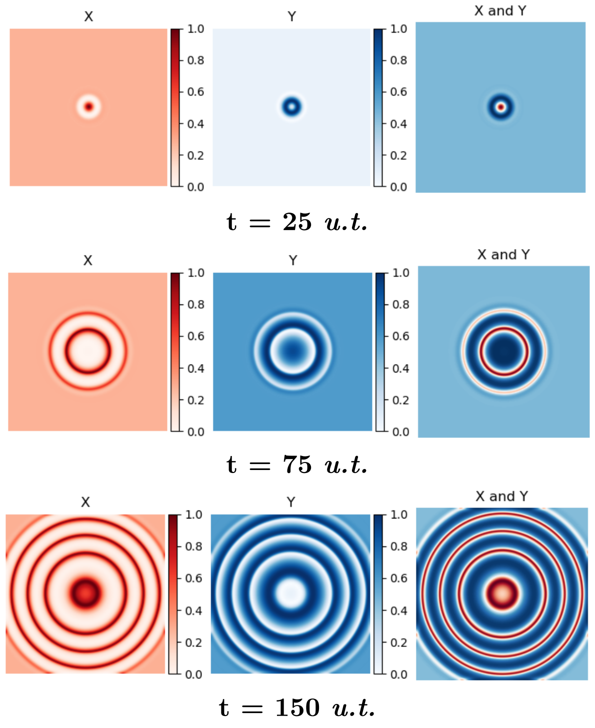


Figure 5: Target patterns in the Brusselator system.

2.3. Characteristic Curves

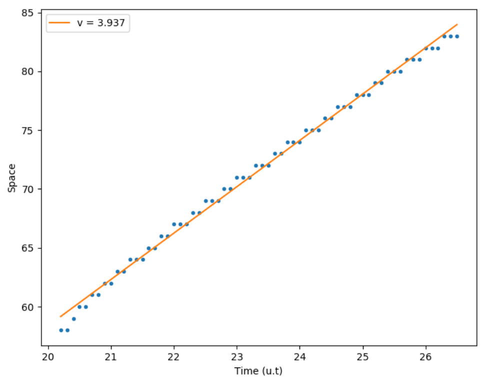
When studying partial differential equations whose solutions are waves that propagate throughout space, the method of characteristics is used to study their propagation speed. The equation of characteristic curves is defined by:

$$x = ct + x_0 \tag{2.7}$$

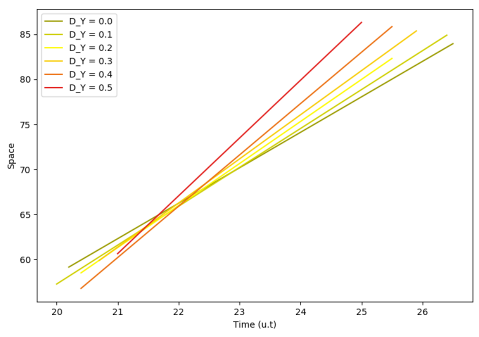
where c is a constant with velocity dimensions. For our study of the speed of the waves formed by the Brusselator, the parameters $A = 1.0$, $B = 2.3$ and $k_i = 1$ were kept fixed, as well as the autocatalytic variable's diffusion coefficient $D_X = 1.0$, while D_Y was varied between 0.0 and 0.5, in 0.1 steps. The simulations were done on a grid of size 100×100.

Initially, a spatial cut is made on the variables' concentrations in the middle of the y-axis, which gives us the reaction-diffusion waves along the x-axis, for $y = 50$. For a single wave-front, the space and time coordinates for which there is a maximum concentration value are taken, and the slope of the line formed by these coordinates gives the propagation speed of the waves (Figure 6).

According to Table 1, there is a non-linear increase of the propagation speed of the waves, according to the increase in D_Y.



(a) The time and space coordinates were linearly fitted to find the propagation speed of the waves, for each value of D_Y.



(b) The propagation speed increases with D_Y. Each line represents a different value of D_Y with a gradient between green and red, where the dark green line shows the propagation speed of a wave simulated with $D_Y = 0.0$ and the red line represents $D_Y = 0.5$.

Figure 6: Study of the propagation speed of concentric waves with parameters: $A = 1.0$, $B = 2.3$, $k_i = 1.0$, $D_X = 1.0$, and varying D_Y between 0.0 and 0.5. The slope of the line formed by the space and time coordinates for which there is a maximum concentration value for a single wave-front gives the propagation speed of the waves.

3. The Data Mining Algorithm

Identifying the structure and parameters of a non-linear system from data has been made possible by advances in sparse regression techniques [17, 18]. These techniques resulted in an algorithm that aims to derive governing equations from time series data collected at a fixed number of spatial locations, the PDE-FIND algorithm. This algorithm applies sparse regression to discover the terms of the governing PDE that represent the data with the greatest accuracy from a large library of possible governing PDEs [19, 20]. It has proved to successfully select the correct linear, non-linear, and spatial derivative terms from a large library, which result

Propagation Speed	D_Y = 0.0	D_Y = 0.1	D_Y = 0.2	D_Y = 0.3	D_Y = 0.4	D_Y = 0.5
v	3.937	4.317	4.667	4.898	5.698	6.418

Table 1: Results for the propagation speeds of the wave fronts for varying values of D_Y.

in the accurate identification of PDEs from data.

3.1. PDE-FIND algorithm

We start by assuming a parameterized and non-linear PDE of the general form:

$$u_t = N(u, u_x, u_{xx}, ..., x, t, \mu) \tag{3.1}$$

where the subscripts represent partial differentiation in time or space, μ denotes parameters in the system, and N is an unknown right-hand side that is usually a non-linear function of $u(x,t)$, its derivatives, and μ parameters. The main objective of the algorithm is to construct N from time series measurements of the system at a fixed number of spatial locations in x. It is assumed that the function N may be expressed as a sum of a small number of terms, which makes the space of possible contributing terms very large compared to the sparse functional form.

Upon discretization, $\mathbf{U}$ is denoted to be a matrix containing the values of u, and hence the right hand side of equation (3.1) can be expressed as a function of $\mathbf{U}$. There is also a matrix $\mathbf{Q}$ which contains additional information about the system that may be relevant.

Initially, all the spatial time series data is collected and combined into a single column vector $\mathbf{U} \in \mathbb{C}^{n\times m}$, which represents data collected over m time points and n spatial locations. The additional input is also considered in a column vector $\mathbf{Q} \in \mathbb{C}^{n\times m}$. Then, the algorithm creates a large library $\mathbf{\Theta}(\mathbf{U}, \mathbf{Q}) \in \mathbb{C}^{nm\times D}$ of D candidate terms that may appear in N, including linear and non-linear terms, and partial derivatives, and then a sparse subset of active terms is selected from this list. The candidate terms are then combined into a matrix $\mathbf{\Theta}(\mathbf{U}, \mathbf{Q})$:

$$\mathbf{\Theta}(\mathbf{U}, \mathbf{Q}) = \begin{bmatrix} 1 & \mathbf{U} & \mathbf{U}^2 & ... & \mathbf{Q} & ... & \mathbf{U}_x & \mathbf{U}\mathbf{U}_x \end{bmatrix} \tag{3.2}$$

Each column of the matrix $\mathbf{\Theta}$ contains all the values for a candidate function across all the grid points on which data was collected, as shown in Figure 7 (1b). The time derivative is also taken to compute $\mathbf{U}_t$ and it is then reshaped into a column vector, just like the columns of $\mathbf{\Theta}$. The PDE evolution can be represented by the linear equation:

$$\mathbf{U}_t = \mathbf{\Theta}(\mathbf{U}, \mathbf{Q})\xi \tag{3.3}$$

For large data sets, PDE-FIND can be effectively used on subsampled data. A set of spatial points is randomly selected and uniformly subsampled in time, resulting in the use of only a fraction of the dataset. In the linear system in 3.3, a fraction of the rows is ignored. The subsampling method is illustrated in Figure 7.

Usually, the number of rows in a linear system is the same as the total number of data points, which calls for a very large system. If Θ is assumed to be an over complete library, there should be the possibility to represent the PDE with a sparse vector of coefficients ξ. The algorithm will pick enough candidate functions that the full PDE can be written as a weighted sum of library terms. In this linear system, each row represents an observation of the dynamics at some point in time and space:

$$u_t(x, y) = \sum_j \Theta_j(u(x,t), q(x,t))\xi_j \tag{3.4}$$

3.2. Sparse Regression

Usually, the sparsest vector ξ that satisfies Equation 3.3 is required. One could think to simply solve the least squares problem for ξ in order to get a representation of the dynamics. However, this would lead to a PDE with every functional form contained in the library. A better alternative is to use penalized sparse regression, which allows to create a linear regression model that is penalized for having too many variables in the model. Sparse regression is used to approximate a solution of

$$\xi = argmin_{\hat{\xi}} \left\|\mathbf{\Theta}\hat{\xi} - \mathbf{U}_t\right\|_2^2 + \lambda \left\|\hat{\xi}\right\|_0 \tag{3.5}$$

where ξ represents the true (unknown) parameter value that generated the data and $\hat{\xi}$ is its estimate, and $\lambda \left\|\hat{\xi}\right\|_0$ is a penalty term ($\lambda > 0$ represents how much is penalized).

The first tested regression method was to relax the problem to a convex ℓ_1 regularized least squares [21]; however, this technique was found to have difficulty finding a sparse basis when the data matrix Θ has high correlations between columns, which is the case for many dynamical systems. Then, a second alternative method for sparse regression was tested, which is called sequentially thresholded least squares (STLS) [19]. Although this method showed better results, it still did not perform outstandingly when applied to the PDE-FIND algorithm. Finally, the problem was approximated using candidate solutions to a ridge regression problem with

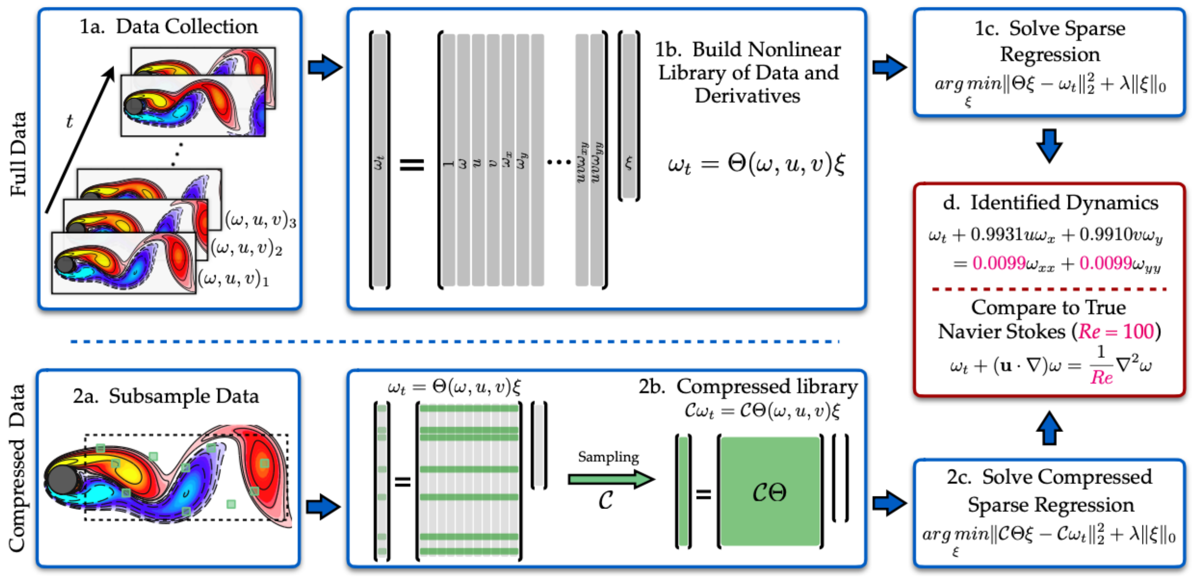

Full Data
1a. Data Collection
t
$(\omega,u,v)_3$
$(\omega,u,v)_2$
$(\omega,u,v)_1$
1b. Build Nonlinear Library of Data and Derivatives
$\omega_t = \Theta(\omega,u,v)\xi$
1c. Solve Sparse Regression
$\arg\min_{\xi} \|\Theta\xi - \omega_t\|_2^2 + \lambda\|\xi\|_0$
d. Identified Dynamics
$\omega_t + 0.9931u\omega_x + 0.9910v\omega_y = 0.0099\omega_{xx} + 0.0099\omega_{yy}$
Compare to True Navier Stokes ($Re = 100$)
$\omega_t + (\mathbf{u}\cdot\nabla)\omega = \frac{1}{Re}\nabla^2\omega$
Compressed Data
2a. Subsample Data
$\omega_t = \Theta(\omega,u,v)\xi$
2b. Compressed library
$\mathcal{C}\omega_t = \mathcal{C}\Theta(\omega,u,v)\xi$
Sampling
$\mathcal{C}$
$\mathcal{C}\Theta$
2c. Solve Compressed Sparse Regression
$\arg\min_{\xi} \|\mathcal{C}\Theta\xi - \mathcal{C}\omega_t\|_2^2 + \lambda\|\xi\|_0$

Figure 7: Steps in the PDE-FIND algorithm, applied to infer the Navier-Stokes equation from data, for a full dataset and for a compressed dataset. Starting from data (**1a.**), a large linear system is formed to represent the PDE, by taking the numerical derivatives and compiling the data into a large matrix $\boldsymbol{\Theta}$, where every column is a possible non-linear function of the data (**1b.**). Sparse regression is then used (**1c.**) to identify active terms in the PDE (**d.**) . For large datasets, we can identify the same dynamics by using a small subset of the data points, so sparse sampling may be used to reduce the size of the problem (**2a.**), which is equivalent to taking a subset of rows from the linear system (**2b.**). An identical sparse regression problem is formed but with fewer rows (**2c.**), and finally the active terms in ξ are synthesized into a PDE (**d.**).

hard thresholding: in this algorithm, least squares from STLS is substituted by ridge regression, and so this regression method was given the name Sequential Threshold Ridge regression (STRidge). This is an ℓ_2 regularized variation of the least squares problem, with an ℓ_2 norm that corresponds to the sum of the squared coefficients, which corresponds to shrinking the regression coefficients so that variables with minor contribution to the outcome have their coefficients close to zero. It is defined by:

$$\begin{aligned}\hat{\xi} &= argmin_{\xi} \left\|\boldsymbol{\Theta}\hat{\xi} - \mathbf{U}_t\right\|_2^2 + \lambda \|\xi\|_2^2 \\ &= (\boldsymbol{\Theta}^T\boldsymbol{\Theta} + \lambda I)^{-1}\boldsymbol{\Theta}^T\mathbf{U}_t\end{aligned} \tag{3.6}$$

3.3. Numerical Evaluation of Derivatives

The most important task for the success of the PDE-FIND algorithm is the numerical evaluation of derivatives, which also shows to be the biggest challenge in its implementation [1]. The best derivative computation method was found to be polynomial interpolation. For each point where there is a derivative being computed, a polynomial of degree P is fit to greater than P points, and derivatives of the polynomial are taken to be approximate to those of the numerical data. It is difficult to fit a polynomial near the boundaries, so the data points close to them are not used in the regression. This data is difficult to differentiate, and it was found that this strongly influences the results and accuracy of PDE-FIND.

3.4. Limitations

There are a few situations which may lead to the underperformance of the PDE-FIND algorithm. The first one is when there is an incomplete library of terms. When the algorithm is applied to a dataset where the dynamics are unknown, it might happen that the column space of Θ is insufficient. When this is the case, the PDE-FIND algorithm will usually not converge to the real dynamics of a system. Another challenging case is when a dataset has a high level of noise. The problem of numerically differentiating noisy data makes identifying governing equations difficult. It is expected that, by increasing the level of noise, the PDE-FIND algorithm is able to identify the correct terms with increasing error until the noise level reaches a value for which more terms than necessary are added in the equations. Solutions computed on courser grids, this is, grids with less spatial and temporal points, show a decline in accuracy when compared to solutions computed from finer grids. Using a large number of points not only helps supply sufficient data for the regression, but most importantly it makes numerical evaluation of derivatives possible.

4. Results

After carefully analysing the PDE-FIND algorithm and deeply learning how it works, it was finally tested on a dataset of the 2D Brusselator system. If the model is successfully able to identify the correct system of partial differential equations of the Brusselator system, we show that it is possible to obtain valuable information of the local dynamics of a system from data, and therefore to infer its underlying kinetic mechanisms.

4.1. PDE-FIND algorithm on the Brusselator system

First, the Brusselator dataset was compute in Matlab, setting the initial conditions and the equations for the evolution of the system, its parameters, grid size and data acquisition time. The Brusselator parameters were set to: $A = 1.0$, $B = 2.3$, $k_i = 1.0$, $D_X = 1.0$ and $D_Y = 0.0$. The variables were then saved on a Matlab file format to be read by the algorithm. Figure 8 shows a snapshot of the evolution of the Brusselator system at a certain time point, to ensure that the data acquisition was well performed.

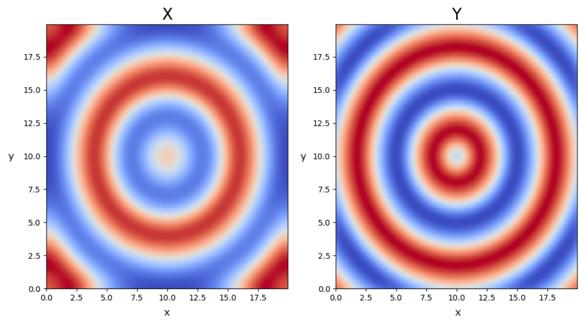


Figure 8: Numerical solution to the Brusselator system of equations, plotted in space-time for $t = 10$ u.t.. Red colour represents the autocatalytic X substance and blue represents the Y substance.

The conditions that can be varied according to how many sampling points we want in our simulation, as well as how fine or coarse we choose our grid of data points to be, are: the number of spatial points in the grid $m \times m$, the number of temporal points in the simulation n as well as the time step dt, the number of spatial (num_{xy}) and temporal (num_t) points to be actually used in the simulation, and the step taken by the algorithm when it is searching for the optimal tolerance d_{tol} . As such, the chosen testing conditions were:

$$n = 512, m = 201, dt = 0.05 \tag{4.1}$$

$$num_{xy} = 50 \tag{4.2}$$

$$num_t = 85 \tag{4.3}$$

$$d_{tol} = 1 \tag{4.4}$$

Results obtained with a lower number of sampling points resulted in a less accurate convergence. Applying the algorithm to the Brusselator dataset with the above mentioned testing conditions, the algorithm converged to equations for u and v very satisfactorily close to the real equations, and the results are presented in Table 4.1.

This result shows that the algorithm was able to accurately identify the PDE with an error of $(0.04 \pm 0.01)\%$ when compared to the original set of equations.

In order to see if the method would be equally successful if we added diffusion on the inhibiting variable, two other values of D_Y were tested. The results for $D_Y = 0.5$ and $D_Y = 1.0$ are shown in Table 4.1. Although the algorithm was able to converge, a larger number of sampling points ($num_{xy} = 10000$) had to be used for both cases, otherwise the method would be unsuccessful. The remaining testing conditions 4.1 were maintained. The algorithm was able to identify the PDE's for both cases with errors of $(0.78 \pm 1.41)\%$ for $D_Y = 0.5$ and $(0.17 \pm 0.16)\%$ for $D_Y = 1.0$.

Finally, the algorithm was also tested on spiral waves, shown in Figure 9. Maintaining the same parameters as the original system ($A = 1.0$, $B = 2.3$, $k_i = 1.0$, $D_X = 1.0$ and $D_Y = 0.0$) and testing conditions 4.1, the spiral waves are obtained by changing the initial conditions.

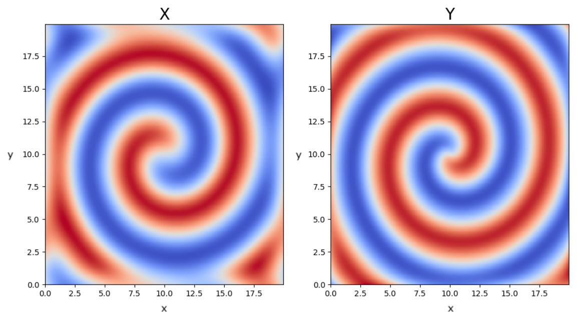


Figure 9: Numerical solution to the Brusselator system of equations, exhibiting spiral waves, for $t = 10$ u.t..

The method was once again able to accurately converge to the real system of PDE's with an error of $(0.04 \pm 0.01)\%$. Results for every case are present in Table 4.1.

Correct PDE ($D_Y = 0.0$)	$u_t = 1.0 + u_{xx} + u_{yy} - 3.3u + u^2v$ $v_t = 2.3u - u^2v$
Identified PDE	$u_t = 0.999538 + 0.999808u_{xx} + 0.999375u_{yy} - 3.298330u - 0.999497u^2v$ $v_t = 2.299332u - 0.999729u^2v$
Correct PDE ($D_Y = 0.5$)	$u_t = 1.0 + u_{xx} + u_{yy} - 3.3u + u^2v$ $v_t = 0.5v_{xx} + 0.5v_{yy} + 2.3u - u^2v$
Identified PDE	$u_t = 0.998089 + 1.047309u_{xx} + 0.996114u_{yy} - 3.290348u - 0.995294u^2v$ $v_t = 0.500109v_{xx} + 0.500111v_{yy} + 2.294337u - 0.994988u^2v$
Correct PDE ($D_Y = 1.0$)	$u_t = 1.0 + u_{xx} + u_{yy} - 3.3u + u^2v$ $v_t = v_{xx} + v_{yy} + 2.3u - u^2v$
Identified PDE	$u_t = 0.996333 + 1.001942u_{xx} + 1.001798u_{yy} - 3.283741u - 0.997793u^2v$ $v_t = 1.000130v_{xx} + 1.000112v_{yy} + 2.299436u - 0.999773u^2v$
Correct PDE (spirals)	$u_t = 1.0 + u_{xx} + u_{yy} - 3.3u + u^2v$ $v_t = 2.3u - u^2v$
Identified PDE	$u_t = 0.999620 + 0.999541u_{xx} + 0.999577u_{yy} - 3.298610u - 0.999578u^2v$ $v_t = 2.299442u - 0.999773u^2v$

Table 2: Identification of the Brusselator reaction-diffusion equation system with PDE-FIND.

4.2. Limited Data

In Section 3.4, it was referred that one of the cases for which the PDE-FIND algorithm may not converge to the desired solution is when there is not a rich enough dataset with a sufficient number of points to successfully numerically evaluate the derivatives. To test this, the algorithm is applied on a number of discretizations of the Brusselator equations, and the results are shown in Table 3. Initially, the system was computed on a finer grid over 201 temporal points, and successively computed on to coarser grids over shorter sampling times, in order to evaluate the method with courser sampling.

		Spatial Points (n)				
		512	256	128	64	32
Temporal Points (m)	201	0.041	0.041	0.049	0.28	🟥
	101	0.77	0.71	🟥	🟥	🟥
	51	🟥	🟥	🟥	🟥	🟥

Table 3: Accuracy of the PDE-FIND algorithm with various grids sizes on the Brusselator system. Red table entries denote a misidentification of the sparsity pattern either due to the inclusion of extra terms which are not present in the real PDE, or missing any term of either the PDE for u_t or v_t. Numbers shown represent the average parameter error in percentage.

Finally, another analysis was done by varying the number of spatial sampling points num_{xy} on a grid of size 512, over an interval of 201 u.t. , where only 85 sampling temporal points are chosen. This was done to test the lower limit from which the algorithm fails to correctly converge, and the results are shown in Table 4. We can see that there is a lower threshold of 25 sampling spatial points from which the PDE stops being correctly identified.

Number of sampling points	5000	1000	100	50	25	15	10
Error in percentage	0.041	0.039	0.036	0.041	🟥	🟥	🟥

Table 4: Accuracy of the PDE-FIND algorithm with different numbers of spatial sampling points num_{xy} on the Brusselator system. The numbers shown represent the average parameter error in percentage. Red table entries denote a misidentification of the sparsity pattern due to missing any term of either the PDE for u_t or v_t.

5. Conclusions

In this work, we have analysed the possibility of calibrating a reaction-diffusion partial differential equation exhibiting travelling waves transient solutions with the PDE-FIND algorithm. With this, we have shown the possibility of introducing the BZ data to obtain information about the local dynamics and therefore for underlying kinetic mechanisms of the BZ reaction. Our ultimate goal would be to find the optimal R-D system that best fits the BZ reaction data.

After having successfully calibrated the model, the next step would be to capture and analyse images of reaction-diffusion travelling waves of the Belousov-Zhabotinsky reaction in order to create a dataset to be read by the algorithm. To do this, we would start by creating the reaction in the lab, by mixing sulphuric acid, ferroin solution, sodium bromate, potassium bromide and sodium malonate. Then, resorting to suitable material and software, video footage of the evolution of the reaction would be taken and analysed. For this work, only circular waves would be studied. The different colour channels (red and blue) would be separated to represent the X and Y variables, respectively. Then, a dataset containing these variables would be created by scanning images of the travelling waves during successive time instants, which would transform the visual data into numerical data points. Finally, the algorithm would read this dataset and deduce the kinetic mechanism associated with reaction-diffusion waves observed in the Belousov-Zhabotinsky reaction.

References

[1] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 2017. 3:e1602614.

[2] J. Sainhas and R. Dilão. Morfogénese em sistemas de reacção-difusão. *Ciência*, 9.VI(12), 1998.

[3] A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72, 1952.

[4] J. Tyson. What Everyone Should Know About the Belousov-Zhabotinsky Reaction. *Frontiers in Mathematical Biology*, 100:569–587, 1994.

[5] I. R. Epstein and J. A. Pojman. *An Introduction to Nonlinear Chemical Dynamics.* Oxford University Press, 1998.

[6] A. M. Zhabotinsky. Belousov-Zhabotinsky reaction. *Scholarpedia*, 2(9):1435, 2007. doi:10.4249/scholarpedia.1435.

[7] S. Ault and E. Holmgreen. Dynamics of the brusselator. 2003.

[8] A. M. Zhabotinsky. A history of chemical oscillations and waves. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 1(4):379–386, 1991. doi:10.1063/1.165848.

[9] I. Prigogine and R. Lefever. Symmetry breaking instabilities in dissipative systems. ii. *Chemical Physics - CHEM PHYS.*, 48(4):1695–1700, 1968. doi:10.1063/1.1668896.

[10] R. H. Enns and G. McGuire. *Nonlinear Physics with Mathematica for Scientists and Engineers.* Birkhauser, 2001.

[11] T. D. Sauer. Attractor reconstruction. *Scholarpedia*, 1(10):1727, 2006. doi:10.4249/scholarpedia.1727.

[12] R. Sujith. Tutorial 2: Tools from nonlinear dynamics. 2019.

[13] F. Takens. *Detecting Strange Attractors in Turbulence.*, volume 898. 11 2006.

[14] R. Dilão. *Uma Introdução à Teoria dos Sistemas Dinâmicos e do Caos.* 2018.

[15] M. B. Kennel, R. Brown, and H. D. Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical review A*, 45(6):3403, 1992.

[16] J. Sainhas. *Morfogénese em Sistemas de Reacção-Difusão.* PhD thesis, Instituto Superior Técnico, 1999.

[17] J. Bongard and H. Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. U.S.A.*, 104:9943–9948, 2007.

[18] M. Schimidt and H. Lipson. Distilling free-form natural laws from experimental data. *Science*, 324:81—-85, 2009.

[19] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. U.S.A.*, 113:3932–3937, 2016.

[20] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton. Data-driven discovery of coordinates and governing equations. *Proc. Natl. Acad. Sci. U.S.A.*, 116:22445–22451, 2019.

[21] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. B*, 58:267—-288, 1996.