Maria Chiara D'Autilia

Parameter Identification Problems in Differential Models: numerical analysis and applications

Dottorato in Matematica ed Informatica, Salento (2019).

<http://www.bdim.eu/item?id=tesi_2019_DAutiliaMariaChiara_1>

Università del Salento

Dipartimento di Matematica e Fisica "E. De Giorgi"

Doctoral thesis in Mathematics and Computer Science

Parameter Identification Problems in Differential Models: numerical analysis and applications

Tutor:

Chiar.ma Prof.ssa

Ivonne SGURA

Dottoranda:

Maria Chiara D'AUTILIA

Dottorato di ricerca in Matematica ed Informatica XXXI ciclo

Mathematics subject classification: 65L09 - 65M32 - 65M06 - 35K57 - 65K10

Abstract

In many scientific fields the experimental data are usually given by time series or images, describing some chemico-physical phenomena of interest. One of the main problem in science and engineering is to define a mathematical model describing these phenomena and depending on a set of parameters physically traceable in the experimental processes. In this thesis we are interested in the Parameter Identification Problem (PIP) for differential models that can be formulated as a constrained minimization problem, where the cost function measures a certain distance between the data and the solution of the model. The constraints are represented by a systems of Ordinary Differential Equations (ODEs) or Partial Differential Equations (PDEs) and the unknowns are the parameters of the differential model. The thesis is divided in two parts: at first we focus on the ODE-PIP and then we examine the PDE-PIP. In particular, in the first case we deal with time series with oscillatory behavior; in the second case we study images that represent peculiar spatial structures.

For the ODE-PIP, since it can be seen as an optimal control problem, we discuss the well known Direct and Indirect approaches for the approximation of the optimal solution and we analyze the numerical issues involved at each step of the discrete formulation. Then, we present an ODE-PIP based on an ODE system with oscillatory dynamics. We show that the classical Direct approach, based on the least-square norm as cost function, fails in the minimization due to the presence of multiple minima. For this reason we propose a Fourier regularization approach (*Inverse Problem*, **33**(12), 2017), that is able to identify an iso-frequency manifold $\mathcal{S}$ in the parameter space, such that for all parameters in $\mathcal{S}$ the ODE solutions have the same frequency of the assigned data.

For the PDE-PIP, we consider as constraint a Reaction-Diffusion (RD) PDE system, whose solutions include the Turing patterns with particular spatial structures like labyrinths, spots, etc. Since the numerical approximation of Turing patterns is challenging from the computational point of view, we focus on the efficient discretization of the RD-PDEs, by analyzing the use of matrix-oriented approach. At each time step we solve Sylvester-matrix equations that allows to deal with significantly smaller matrices, showing that the computational cost can be made lower than that of the corresponding vector approaches, by working in

the reduced (spectral) space. Finally, we solve the PDE-PIP for the morphochemical model for electrodeposition (DIB) (*Journal of Solid State Electrochemistry*, **17**(2), 2013) describing the metal growth during the battery charging process for synthetic and experimental images. We show that, by following the Direct approach based on the least-squares minimization, the model can fit a rich variety of patterns arising in the experiments. Then, to further improve the search of the optimal parameters, we present the first results of the extension of the Fourier approach to the PDE case.

Contents

1 Parameter Identification Problem in differential equations 13

1.1 Formulation for ODEs 13

1.2 ODE Test Identification Problem 16

1.3 Direct approach: *Discretize-then-Optimize* 20

1.4 Indirect Approach: *Optimize-then-Discretize* 23

1.5 Literature review 27

2 Discretization issues: ODE case 29

2.1 Numerical methods for oscillating solutions 31

2.2 Runge-Kutta methods for partitioned systems 37

3 Comparison between Direct and Indirect Approaches 43

3.1 Analysis of cost functions 43

3.1.1 Cost in Direct Approach 44

3.1.2 Cost in the Indirect Approach 46

3.2 Gradient analysis 51

3.2.1 Gradient in the Direct Approach 51

3.2.2 Gradient in Indirect Approach 53

3.3 Summary 58

4 ODE-PIP with oscillatory dynamics 59

4.1 Test Identification problem for oscillating data (TIP-OD) 60

4.2 Fourier regularization: simulated data 64

4.3 Fourier regularization: experimental data 70

4.3.1 DIB-PIP: Fourier regularization for $m = 3$ parameters . . . 75

4.4 Application: Dynamics of zinc-air battery anodes 78

4.4.1 Relaxation oscillations 80
4.5 Summary 82

5 PDE-PIP **85**
5.1 Formulation for PDEs 85
5.2 Discretization issues for PDE-PIP 87
5.3 Matrix-oriented methods for the approximation of RD-PDEs 89
5.3.1 Classical vector methods and their matrix formulation . . . 92
5.3.2 Implementation details 97
5.3.3 RD-PDE systems: matrix approach 98
5.4 Summary 106

6 Application: PDE-PIP for a morphochemical model **108**
6.1 DIB model: description and formulation of DIB-PIP 108
6.2 DIB-PIP: numerical results 112
6.3 Fourier approach for DIB-PIP 114
6.3.1 Simulated data: *Square*-patterns 116
6.3.2 Experimental data 118
6.4 Summary 121

7 Future work **123**

A Fast Fourier Transform **125**

B Algebraic curvature **127**

Introduction

In many scientific fields the experimental data are usually given by time series or images, describing some chemico-physical phenomena of interest. One of the main problem in science and engineering is to define a mathematical model describing these phenomena and depending on a set of parameters physically traceable in the experimental processes. In this thesis we are interested in the Parameter Identification Problem (PIP) for differential models that can be formulated as a constrained minimization problem, where the cost function measures a certain distance between the data and the solution of the model. The constraints are represented by a systems of Ordinary Differential Equations (ODEs) or Partial Differential Equations (PDEs) and the unknowns are the parameters of the differential model. The parameter estimation related to ODEs and PDEs is extremely important in several scientific applications and it is widely studied, see e.g.[7, 15, 23, 32, 48, 81].

For a given target $\widetilde{\mathbf{u}}$, a general PIP in the continuous framework can be formulated as follows:

$$\min_{\mathbf{p}\in\Omega} J(\mathbf{u}, \widetilde{\mathbf{u}}, \mathbf{p}) \tag{1}$$

where $\mathbf{p} \in \Omega \subset \mathbb{R}^m$ is the set of parameters to identify, $\mathbf{u}$ is the solution of a differential model depending on the parameters $\mathbf{p}$, in particular

$$\mathbf{u} = \mathbf{u}(t; \mathbf{p}) \quad t > 0$$

in ODEs case,

$$\mathbf{u} = \mathbf{u}(x, y, t; \mathbf{p}) \quad (x, y) \in \mathcal{D} \subset \mathbb{R}^2,\ t > 0$$

in PDEs case and $\widetilde{\mathbf{u}}$ in the first case is an assigned time-dependent function $\widetilde{\mathbf{u}}(t)$, in the second case is a space dependent function $\widetilde{\mathbf{u}}(x, y)$; J is the *cost function* to minimize, that measures a certain distance between $\mathbf{u}$ and $\widetilde{\mathbf{u}}$. Therefore the

differential model represents the constraint of the PIP, and it is given by the Cauchy problem:

$$\begin{cases} \mathbf{u}'(t) = f(t, \mathbf{u}; \mathbf{p}), \quad t \in [t_0, T] \\ \mathbf{u}(t_0) = \mathbf{u}_0 \end{cases} \tag{2}$$

in ODE case, where $\mathbf{u}(t) : [t_0, T] \to \mathbb{R}^d$; or

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t}(x, y, t) = g(\mathbf{u}, \mathbf{u}_x, \mathbf{u}_{xx}, \mathbf{u}_y, \mathbf{u}_{yy}; \mathbf{p}), \quad (x, y) \in \mathcal{D} \subset \mathbb{R}^2, \ t \in [t_0, T] \\ \mathbf{u}(t_0) = \mathbf{u}_0, \end{cases} \tag{3}$$

with appropriate boundary conditions in the PDE case, and $\mathbf{u}(x, y, t) : \mathcal{D} \times [t_0, T] \to \mathbb{R}^d$.

The minimization problems (1)-(2) or (1)-(3) can be solved by using the Direct or Indirect approach. The Direct approach transforms the continuous problem into a constrained nonlinear programming problem in finite dimension, to be solved numerically by using an optimization method [63]. The Indirect approach uses the theory of the Optimal Control and the Pontryagin's maximum principle [49, 87] to determine the so-called *optimality conditions*. These necessary conditions are two differential equations (*state* and *adjoint* equations), which arises from the differentiation of the *Hamiltonian* function associated to the problem; generally, they are nonlinear and we do not have the analytic solutions, therefore they must be approximated numerically. In both cases, the minimization problem is discretized and solved numerically by using a suitable optimization algorithm. Typically, in the literature the Direct approach has been applied, see for example [24, 59, 92] for ODEs and [19, 32, 41, 60, 78] for PDEs. In the last years, another method for the PIP has been explored: the so-called Bayesian approach (see, for instance, [15, 43, 84]), that consists in a set of techniques built around Bayes' theorem, which, roughly speaking, allows one to compute the conditional probability of an event A given the probability of an event B, in terms of the reverse conditional probability, i.e. the probability of B given A. In the parameter identification framework, this will correspond to computing the probability of the parameters given the data, in terms of the probability of the data given the parameters. In this work we discuss the Direct and Indirect approaches and we analyze the numerical issues involved at each step of the discrete formulation. The thesis is divided in two parts: firstly we focus on the ODE-PIP, and then we

examine the PDE-PIP. In particular, in the first case we deal with time series with oscillatory behavior; in the second case we study images that represent peculiar spatial structures.

The resolution of a PIP implies different numerical issues, for example: the choice of the discrete cost, the suitable ODEs solver and the selection of an optimization algorithm. Hence, the multiple purposes of this thesis can be briefly summarized as follows:

- show the formulation of PIP for differential models;
- analyze the Direct and Indirect approaches from the numerical point of view;
- focus on the efficient numerical methods for the differential models (both ODEs and PDEs);
- provide examples and real applications of experimental interest for both ODEs and PDEs cases.

To describe in detail the Direct and Indirect approaches, in Chapter 1 we will refer to the PIP in the case of ODEs (ODE-PIP). We will formulate the continuous problem and show the discrete forms derived from the application of the two approaches by considering a simple case (Test Identification Problem); we will deduce some numerical issues for the discrete problem, that include: (i) the (weighted) norm for the cost function in the Direct approach; (ii) the quadrature formula for the approximation of the cost function in the Indirect approach (which is usually given by an integral); (iii) the ODE solvers in both cases, and the method for the adjoint equations to be solved backward in time and (iv) the optimization algorithm for the minimization (e.g. gradient descent, Newton's method, etc,...) . After a brief analysis of all of them, we present the numerical methods for ODEs with oscillatory dynamics, by identifying the method that minimize the dispersion error, and for the so-called partitioned systems (derived from the application of the Indirect approach).

Then we will present an error analysis for both approaches; in particular, in Chapter 3, we will demonstrate that the errors between the discrete costs (and their gradients, usually required in the optimization algorithms) and the cost (and

gradient) of the continuous problem involves two different contributions: one given by the method used to define the discrete cost functions and a second one given by the method for the approximation of the ODEs solutions. We prove that, in some cases, these errors do not vanish for $h \to 0$, where h represents stepsize for approximation in time.

In Chapter 4 we will present an application of ODE-PIP, where the target data and the solution of the ODEs have an oscillatory dynamics. In case of oscillating data, we will show that the cost function inherits the oscillating data behavior and has many different "low" minima. To face this ill-posedness, the cost function is usually corrected by a so-called regularization term in order to ensure the convexity and well-posedness of the problem. We will show that adding a classical regularization term actually does not improve the structure of the cost. Since in this situation any optimization algorithm is liable to fail in the approximation of a good solution, we propose a new approach which takes into considerations the oscillating nature of the data. Therefore, we rewrite and solve the original ODE-PIP in the Fourier space, by defining a new cost function based on the Discrete Fourier Transform (DFT) (see Appendix A) that compares frequencies of data and simulations. We present the results for simulated oscillatory data in the case of the two-parameters Schnakenberg model, in the Hopf regime. As a true application, we apply the Fourier-PIP regularization to follow original experimental data with the morphochemical model for electrodeposition (DIB) [10] in the case of two and three parameters. The results have been published in [20]. In the last Section of the Chapter, we will study the behavior of zinc-air batteries anodes, whose process can be rationalized within a mathematical model [8]. In particular, we show that it is possible to follow the oscillating regimes of current identifying the physical parameters in the zone of the so-called *relaxation oscillation* of the parameters space.

In the second part of the thesis, starting from Chapter 5, we formulate the PIP in case of PDEs. We will focus on a specific type of PDEs: the Reaction-Diffusion model, well-known in describing pattern formation in several scientific fields, whose solution display a wide range of behaviors including the formation of self-organized patterns like stripes or more intricate structures, the so-called "Turing patterns" [89]. The PIP for these patterns is a recent field of appli-

cation, and an increasing number of papers are devoted to this study, see e.g. [15, 31, 32, 33, 81]. We will formulate the PDE-PIP for a generic RD-model, then we will describe the numerical issues deriving from its discretization and optimization. In particular, the efficient approximation of the patterns will be studied in depth: this is an important point because any optimization procedure to solve the PIP-PDE will call several times the solver for the constraint given by the PDE model and the Turing pattern approximation poses several numerical challenges (longtime integration, high accuracy in space, etc..). For this reason, by exploring the advantages of the *matrix-oriented* formulation of the semi-discretized problem in space, we report our results in [21] and show that the matrix formulation provides a quite different perspective at the time discretization level than classical approaches, allowing to significantly reduce the memory and computational requirements. In fact, we show that at each time step we do not solve anymore linear systems (usually of big dimensions), but a so-called *Sylvester matrix equations* by means of the spectral decomposition of the coefficient matrices that improves the performance of the algorithm.

At least, we show a real application of PDE-PIP in case of a morphochemical reaction-diffusion model that describes the electrochemical pattern formation (briefly DIB model, from the name of the authors), studied in [10], starting from experimental morphochemical distributions. For DIB-PIP we will present two methods to localize the minimum in the parameters space: one is based on the classical minimization in the least square sense [81]; the other, for which we will show the first preliminary results, works in the Fourier space. In particular, we show that by means of the 2 dimensional - DFT of the experimental map we can obtain more information from the data and find admissible solutions in the parameters space, otherwise unobservable with the classical 2-norm.

The thesis is structured as follows.

In Chapter 1 we define the ODE-PIP, then we formulate a Test Identification Problem (TIP) and illustrate the Direct and Indirect approaches applied to this problem. The discretization issues derived from both formulations are described in Chapter 2. In particular, we focus on the suitable approximation of the ODEs in case of oscillatory behavior and in case of partitioned systems. Chapter 3

is focused on the analysis of the cost functions and gradients of the Direct and Indirect approaches. Then, we deal with a ODE-PIP with oscillatory dynamics, in Chapter 4. At first we show the usefulness of this approach in case of synthetic data, then we apply the same procedure to an experimental case. In Chapter 5 we formulate the PIP in case the constraint is a RD model and we explain the issues deriving from its the discretization. Then, we focus on the efficient approximation of the RD-PDEs solutions by introducing the matrix formulation of the semi-discretized problem, and suitably adapting the well-known classical vector method to this new formulation. In the Chapter 6, we show the application of PDE-PIP to the DIB model for synthetic and experimental data. In the last Chapter 7, we present some extensions and applications of our works, that could be objects of future researches.

Chapter 1

Parameter Identification Problem in differential equations

1.1 Formulation for ODEs

Let us consider the following minimization problem:

$$\min_{\mathbf{p}\in\Omega} J(y(t), \widetilde{y}(t), \mathbf{p}) \tag{1.1}$$

with

$$\begin{cases} y'(t) = G(t, y; \mathbf{p}), & t \in [t_0, T] \\ y(t_0) = y_0; \end{cases} \tag{1.2}$$

where $y(t) = y(t; \mathbf{p}) : [t_0, T] \to \mathbb{R}^d$, $d \geq 1$, is the solution of the system of Ordinary Differential Equations (ODEs) in (1.2) depending on the parameter set $\mathbf{p} \in \Omega \subset \mathbb{R}^m$, $m \geq 1$, $G : [t_0, T] \times \mathbb{R}^d \times \Omega \to \mathbb{R}^d$ is Lipschitz continuous in y, such that the solution $y(t; \mathbf{p})$ of (1.2) exists and is unique for all $\mathbf{p} \in \Omega$. Generally the parameter space is a hyper-rectangle $\Omega = \Pi_{i=1}^{m}[p_i^0, p_i^1]$. $\widetilde{y}(t) \in \mathbb{R}^d$ is a given *target function* and $J(y(t), \widetilde{y}(t), \mathbf{p})$ is a suitable cost function that measures a certain distance between $\widetilde{y}(t)$ and the ODEs solution $y(t; \mathbf{p})$.

We define (1.1)-(1.2) as an ODEs Parameter Identification Problem (ODE-PIP) for $\widetilde{y}(t)$ and an optimal solution will be a set of parameters $\mathbf{p}^*$ such that $y(\mathbf{p}^*) = y^*$ is the ODEs solution nearest to the target $\widetilde{y}$. The target function represents some desired configuration of the system and could be given, for example, from the observation of some physical phenomena (for this reason also said *observation*).

In real applications, the target function is often given in discrete form, that is in terms of some data set: $(\mathbf{t}, \widetilde{\mathbf{y}}) = (t_i, \widetilde{y}_i), i = 0, \dots N_t,\ \widetilde{y}_i \in \mathbb{R}^d$, on a given time grid. If the solution $y(t; \mathbf{p}) \in \mathbb{R}^d$ of the differential equations (1.2) is known in analytic form and the cost function is chosen as the classical residual minimization in the two norm, the ODE-PIP corresponds to an unconstrained (nonlinear) least squares problem, given by

$$\min_{\mathbf{p}\in\Omega}\ J_{2norm}(\mathbf{p}) = \min_{\mathbf{p}\in\Omega}\ \|\mathbf{y}(\mathbf{p}) - \widetilde{\mathbf{y}}\|_2^2 = \sum_{i=1}^{N_t}\sum_{k=1}^{d}(y_i^k(\mathbf{p}) - \widetilde{y}_i^k)^2, \tag{1.3}$$

where $\mathbf{y}(\mathbf{p})$ is the vector which contains the values $y_i(\mathbf{p}) = y(t_i; \mathbf{p}) \in \mathbb{R}^d$ for $i = 0, ..., N_t$, that can be solved by well known literature methods such as Gauss-Newton [6], Levenberg-Marquardt [28, 55], etc.

If the ODE solution $y(t; \mathbf{p})$ is not known in analytic form, then a numerical method must be used to approximate the ODE constraints in (1.2). This is the basis of the so-called Direct Approach [87]: the optimization problem in (1.1)-(1.2) can be written as a nonlinear constrained programming problem in $\mathbb{R}^m$, in the following discrete form:

$$\min_{\mathbf{p}\in\Omega}\ J_{2norm}(\mathbf{p}) = \min_{\mathbf{p}\in\Omega}\ \|\mathbf{u}(\mathbf{p}) - \widetilde{\mathbf{y}}\|_2^2 \tag{1.4}$$

where

$$\begin{cases} u_i = \mathcal{M}(h, u_{i-1}, u_i, t_{i-1}, t_i; \mathbf{p}), \quad u_i \in \mathbb{R}^d, \quad i = 1, \dots, N_t \\ u_0 = y_0 \end{cases} \tag{1.5}$$

with timestep $h = \frac{T-t_0}{N_t}$ on the uniform grid $t_i = t_0 + i\,h$ and where $u_i(\mathbf{p}) \approx y(t_i; \mathbf{p})$ the numerical approximation of the ODE solution at the time t_i. Here $\mathcal{M}$ represents the (implicit, explicit or semi-implicit) numerical method used to solve the ODEs system. The cost function is given by the least squares two norm, but in general it may be another kind of weight norm, as follows:

$$\min_{\mathbf{p}\in\Omega}\ J_{Wnorm}(\mathbf{p}) = \|\mathbf{u}(\mathbf{p}) - \widetilde{\mathbf{y}}\|_W^2 = (\mathbf{u}(\mathbf{p}) - \widetilde{\mathbf{y}})^T W (\mathbf{u}(\mathbf{p}) - \widetilde{\mathbf{y}})$$

where the matrix W is appropriately chosen according to the problem to be solved ($W = I$ yields the 2 norm). The Direct Approach is also indicated as *discretize-then-optimize* approach: the continuous problem is first discretized, then an optimization algorithm is used to approximate the optimal solution.

The Indirect Approach applied to ODE-PIP in continuous form (1.1)-(1.2) instead uses the theory of the Optimal Control and the Pontryagin's maximum principle [64, 87, 49] to determinate the so-called optimality conditions. The cost function (1.1) is usually given by:

$$J(y(t), \widetilde{y}(t), \mathbf{p}) = \|y(\mathbf{p}) - \widetilde{y}\|_{L^2_{[t_0,T]}} = \int_{t_0}^{T} (y(t;\mathbf{p}) - \widetilde{y}(t))^2 dt. \tag{1.6}$$

and by introducing the *adjoint variable* $\psi \in \mathbb{R}^m$, the necessary conditions arise from the differentiation of the following Hamiltonian function associated to the problem (1.6)-(1.2):

$$\mathcal{H}(t, y, \psi, \mathbf{p}) = J(y, \widetilde{y}, \mathbf{p}) + \psi G(t, y; \mathbf{p}). \tag{1.7}$$

From the differentiation of theHamiltonian function wrt the parameters, the so-called *optimality condition* can be derived:

$$\mathcal{H}_{\mathbf{p}} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}} = 0 \Rightarrow J_{\mathbf{p}} + \psi G_{\mathbf{p}} = 0 \tag{1.8}$$

solved in $\mathbf{p}$.

The differentiation of theHamiltonian function wrt the the state and the adjoint variables yields two new constraints, that is the state and adjoint ordinary differential equations given by:

$$\begin{cases} y'(t) = \frac{\partial \mathcal{H}}{\partial \psi}, \ y(t_0) = y_0 \\ \psi'(t) = \frac{\partial \mathcal{H}}{\partial y}, \ \lambda(T) = 0 \end{cases} \tag{1.9}$$

to be solved forward and backward in time, respectively. These equations are often difficult to solve analytically and they are approximated numerically, then also in the Indirect Approach a constrained optimization problem must be solved:

$$\min_{\mathbf{p} \in \Omega} \ J(\mathbf{u}, \widetilde{\mathbf{y}}, \mathbf{p}) \tag{1.10}$$

$$\begin{cases} u_i = \mathcal{M}(h, u_{i-1}, u_i, t_{i-1}, t_i; \mathbf{p}), \quad u_i \in \mathbb{R}^d \\ v_i = \mathcal{N}(h, v_{i+1}, v_i, t_{i+1}, t_i; \mathbf{p}), \quad v_i \in \mathbb{R}^m \qquad i = 1, \ldots, N \\ u_0 = y_0, \ v_{N+1} = 0 \end{cases} \tag{1.11}$$

where (1.10) is given by a quadrature method that approximates the integral in (1.6), $\mathcal{M}$ and $\mathcal{N}$ describe the numerical methods used to approximate forward and backward in time the ODEs in (1.9), with $u_i(\mathbf{p}) \approx y(t_i; \mathbf{p})$ and $v_i(\mathbf{p}) \approx \psi(t_i; \mathbf{p})$.

The Indirect Approach is also called *optimize-then-discretize* approach: in fact, at first the necessary optimality conditions are found and then proceed with the discretization.

It is well known that the identification problem is an inverse problem that often can be ill-posed: solutions might not exist or might not be unique. To face with the ill-posedness, the cost functions in (1.3) and (1.6) are usually corrected by a so-called regularization term [27] in order to ensure the convexity and well posedness of the problem, (see for example [38, 86, 13, 25, 42]). In our notations, for example the classical Tikhonov regularization for (1.3) and (1.6) would be given by:

$$J_{2norm}(\mathbf{p}) = \|\mathbf{y}(\mathbf{p}) - \widetilde{\mathbf{y}}\|_2^2 + \alpha\|\mathbf{p}\|_2^2 \tag{1.12}$$

or

$$J(y(t), \widetilde{y}(t), \mathbf{p}) = \int_{t_0}^{T} (y(t;\mathbf{p}) - \widetilde{y}(t))^2 dt + \beta\mathbf{p}^2 \tag{1.13}$$

with the regularization coefficients $\alpha > 0$ or $\beta > 0$ appropriately chosen.

Then in both Direct and Indirect approaches for PIP-ODE a (differently) constrained minimization problem in finite dimension must be solved numerically. In the following Section we compare them applied to a Test Identification Problem and we will show the construction of the discretization problems derived from both approaches in a simple case. Next we apply an optimization method to approach the optimal solution.

1.2 ODE Test Identification Problem

In this Section we consider a Test Identification Problem (TIP) in order to illustrate the two different approaches applied to a simple case. Let us define the ODE-TIP as follows:

$$\min_{\lambda} J(\lambda) = \int_0^{10} (y(t,\lambda) - \tilde{y}(t))^2 dt \tag{1.14}$$

with the constraint given by the ODE:

$$\begin{cases} y(t)' = \lambda y(t) + \frac{1}{2} \\ y(0) = 1 \end{cases} \tag{1.15}$$

where $\lambda \in \mathbb{R}$ is the *parameter* to identify and $\tilde{y}$ is the *target function*:

$$\tilde{y}(t) = \frac{1}{2}(e^{-t} + 1), \tag{1.16}$$

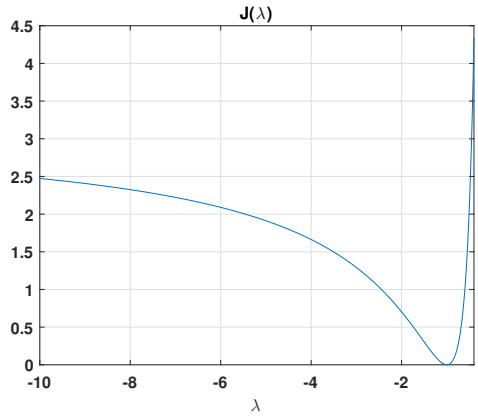

Figure 1.1: Cost functional $J(\lambda)$ in (1.19) for $\lambda \in [-10, -0.4]$, that will be blow-up in $\lambda = 0$.

the solution of the following system (1.15) with $\lambda = -1$:

$$\begin{cases} \tilde{y}(t)' = -\tilde{y}(t) + \frac{1}{2} \\ \tilde{y}(0) = 1. \end{cases} \tag{1.17}$$

Let us observe that we can obtain analytically the solutions of (1.15) as a function of the parameter λ:

$$y(t, \lambda) = \frac{(2\lambda + 1)e^{t\lambda} - 1}{2\lambda}. \tag{1.18}$$

Furthermore, we know the exact solution of the problem: $\lambda^* = -1$ is the *global minimum* of (1.14), such that $J(\lambda^*) = \min_\lambda J(\lambda)$.

Let us observe that we can derive explicitly the analytic form of the cost function in (1.14) as a function of the parameter λ:

$$\begin{aligned} J(\lambda) &= \int_0^{10} \left(\frac{(2\lambda + 1)e^{t\lambda} - 1}{2\lambda} - \frac{1}{2}(e^{-t} - 1) \right)^2 dt = \\ &= \left(8\,\lambda + \frac{6}{\lambda - 1} + 2\,\lambda\,(\lambda + 1) - \frac{(2\,\lambda + 1)^2}{2\,\lambda} + \frac{2}{\lambda} + \frac{\lambda^2}{2} + 12 \right) \frac{1}{4\,\lambda^2} + \\ &+ \left(20\,\lambda - \mathrm{e}^{10\,\lambda - 10} \left(4\,\lambda + \frac{6}{\lambda - 1} + 6 \right) - \frac{\lambda^2\,\mathrm{e}^{-20}}{2} + 10\,\lambda^2 - \mathrm{e}^{10\,\lambda} \left(4\,\lambda + \frac{2}{\lambda} + 6 \right) + \right. \\ &\left. + \frac{\mathrm{e}^{20\,\lambda}\,(2\,\lambda + 1)^2}{2\,\lambda} - 2\,\lambda\,\mathrm{e}^{-10}\,(\lambda + 1) + 10 \right) \frac{1}{4\,\lambda^2} \end{aligned} \tag{1.19}$$

In the Figure 1.1 we represent $J(\lambda)$ for $\lambda \in [-10, -0.4]$.

Moreover, it is possible to obtain the analytic form of the gradient $\nabla J(\lambda)$ as

a function of λ, as follows:

$$\begin{aligned}\nabla J(\lambda) = \frac{dJ(\lambda)}{d\lambda} = \frac{d}{d\lambda}\left[\int_0^{10} (y(\lambda,t) - \tilde{y}(t))^2\, dt\right] = \\ \int_0^{10} \frac{\partial}{\partial\lambda}\left[(y(\lambda,t) - \tilde{y}(t))^2\right] dt = 2\int_0^{10}\left[\frac{\partial y(\lambda,t)}{\partial\lambda}\,(y(\lambda,t) - \tilde{y}(t))\right] dt,\end{aligned} \tag{1.20}$$

replacing the expressions of $y(\lambda,t)$ and $\tilde{y}(t)$ in (1.18)-(1.16), yields:

$$\begin{aligned}&\int_0^{10} \frac{\partial}{\partial\lambda}\left[\left(\frac{(2\lambda+1)e^{t\lambda}-1}{2\lambda} - \frac{1}{2}(e^{-t}-1)\right)^2\right] dt = \\ &= \left[5\lambda - \frac{6}{(\lambda-1)^2} - \frac{8\lambda+4}{2\lambda} + \frac{(2\lambda+1)^2}{2\lambda^2} - \frac{2}{\lambda^2} + 10\right]\frac{1}{4\lambda^2} + \Bigg[(20\lambda + \\ &+ e^{10\lambda}\left(\frac{2}{\lambda^2} - 4\right) - 2\lambda e^{-10} - \lambda e^{-20} - 2e^{-10}(\lambda+1) - 10e^{10\lambda-10}\left(4\lambda + \frac{6}{\lambda-1} + 6\right) + \\ &+ e^{10\lambda-10}\left(\frac{6}{(\lambda-1)^2} - 4\right) - 10e^{10\lambda}\left(4\lambda + \frac{2}{\lambda} + 6\right) + \frac{e^{20\lambda}(8\lambda+4)}{2\lambda} + \frac{10e^{20\lambda}(2\lambda+1)^2}{\lambda} + \\ &- \frac{e^{20\lambda}(2\lambda+1)^2}{2\lambda^2} + 20\Bigg]\frac{1}{4\lambda^2} - \Bigg[20\lambda - \frac{\lambda^2 e^{-20}}{2} - e^{10\lambda-10}\left(4\lambda + \frac{6}{\lambda-1} + 6\right) + \\ &+ 10\lambda^2 - e^{10\lambda}(4\lambda + \frac{2}{\lambda} + 6) + \frac{e^{20\lambda}(2\lambda+1)^2}{2\lambda} - 2\lambda e^{-10}\lambda + 1 + 10\Bigg]\frac{1}{2\lambda^3} + \\ &- \left(8\lambda + \frac{6}{\lambda-1} + 2\lambda(\lambda+1) - \frac{(2\lambda+1)^2}{2\lambda} + \frac{2}{\lambda} + \frac{\lambda^2}{2} + 12\right)\frac{1}{2\lambda^3}.\end{aligned} \tag{1.21}$$

The gradient is shown in Figure 1.2 for $\lambda \in [-10, -0.1]$.

The cost function $J(\lambda)$ and its gradient have some important features of interest for the minimization process:

- $J(\lambda)$ has a global minimum in $\lambda^* = -1$;
- In a right neighborhood of λ^*, for $\lambda \to 0$, the function increases very quickly, and the gradient value is very large, as shown in Figure 1.2 on the right; in fact, it is easy to verify from (1.21) that: $\lim_{\lambda\to 0} \nabla J(\lambda) \approx \frac{1}{\lambda^4}$;
- For $\lambda < \lambda^*$, $J(\lambda)$ increases very slowly and it is flat; the gradient value becomes very small and almost constant, Figure 1.2 on the left;
- $J(\lambda)$ has an inflection point in $\lambda^F \approx -1.51$, as we can observe in Figure 1.1.

Some of these aspects imply several problems in the application of classical optimization methods to find the absolute minimum λ^* of the function $J(\lambda)$. We now describe briefly the iterative optimization method we use below. It has the

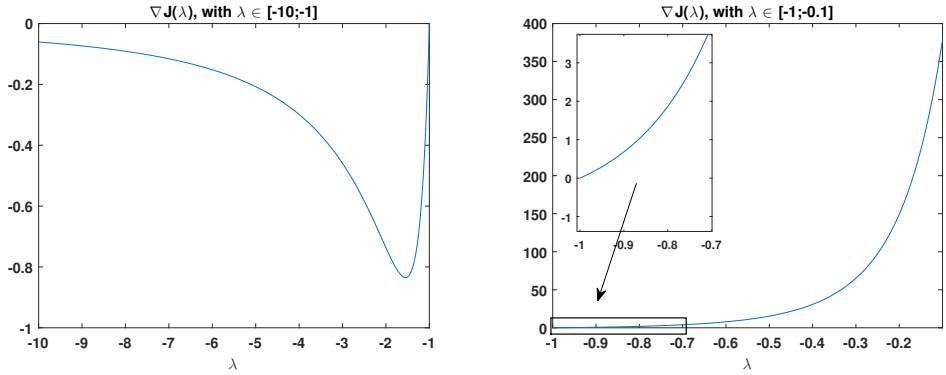

Figure 1.2: Gradient $\nabla J(\lambda)$ as functions of $\lambda \in [-10, -1]$ (plot (a)) and as functions of $\lambda \in [-1, -0.1]$ (plot (b))

following structure:

$$\begin{cases} \lambda_{k+1} = \lambda_k - \alpha_k d_k \qquad k = 0, 1, 2, ... \\ \lambda_0 \quad given \end{cases} \tag{1.22}$$

where:

- α_k is called *step length*, computed by the *Armijo conditions* [63];
- d_k is the *descent direction* chosen as steepest (gradient) direction: $d_k = \nabla J(\lambda_k)$; another choice could be the Newton direction: $d_k = (\nabla^2 J(\lambda_k))^{-1} \nabla J(\lambda_k)$, where $\nabla^2 J(\lambda_k)$ is the hessian of $J(\lambda)$, [63].

The iterative algorithm stops when one of the following *stopping criteria* occurs, where ϵ is called *tolerance* and is a fixed parameter:

- $|J(\lambda_k)| < \epsilon$: the cost is "enough" small (residual criterion);
- $|J(\lambda_k) - J(\lambda_{k-1})| < \epsilon$: in order to stop the iteration if the cost value does not change very much from one iteration to the next (cost increment criterion);
- $|\nabla J(\lambda_k)| < \epsilon$: the gradient value is very small;
- $|\lambda_k - \lambda_{k-1}| < \epsilon$: in order to stop the iteration if the parameter value does not change very much from one iteration to the next (increment criterion).

Let us observe that in general ϵ can be assume a different value for each criteria [22].

As we expect from theory [63], the following issues were observed by using different d_k:

1. Steepest descent. We distinguish three different cases:

   - if $\lambda_0 < \lambda^F$ the convergence is very slow because the function $J(\lambda)$ is flat and the gradient value is very small;

   - if $\lambda^F \leq \lambda_0 < \lambda^*$ the convergence behavior is good, even if this subregion is very small;

   - if $\lambda_0 > \lambda^*$ d_0 is very large and the first iterate $\lambda_1 = \lambda_0 - \alpha_0 d_0$ moves on the left of λ^*. Moreover the distance of λ_1 from λ^* increases if λ_0 is close to 0 and the convergence becomes even more slow.

2. Newton. We can distinguish two cases:

   - if $\lambda_0 < \lambda^F$ the second order derivative is negative and $d_k < 0$, so the method depart from the minimum;

   - if $\lambda_0 > \lambda^F$ Newton direction can be used and fast convergence arises.

In the following Sections we will apply the steepest descent method as optimization algorithm. Other techniques could be considered, but the aim of this Chapter is to focus on the approaches (Direct and Indirect) to build the discrete optimization problem for the problem in the equations (1.14)-(1.15), rather then on its final numerical minimization.

1.3 Direct approach: *Discretize-then-Optimize*

Let us consider the Direct Approach to solve the ODE-TIP. As we have observed we need to discretize (1.15) by a numerical method for ODE. As an illustrative case we choose the Explicit Euler (EE) method; let us fix $h = \frac{T-t_0}{N}$ as step size for time discretization ($N+1$ is the number of the discretization points of the time integration interval). h will be our *discrete parameter*; then it results:

$$u_{n+1} = u_n + h\left(\lambda u_n + \frac{1}{2}\right) \qquad n = 0, ..., N-1.$$

If we call $\mathbf{u} = [u_0, u_1, ..., u_N]$ the vector which contains all the time approximation of y, for the linearity of the equation we can derive the matrix form as follows:

$$\mathbf{u} = (1 + h\lambda) \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ & \ddots & & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix}}_{A} \mathbf{u} + \underbrace{\begin{bmatrix} y_0 \\ \frac{h}{2} \\ \vdots \\ \frac{h}{2} \end{bmatrix}}_{\mathbf{c}}$$

then:

$$\mathcal{M}(\lambda, h)\mathbf{u} = \mathbf{c}(h) \tag{1.23}$$

with:

$$\mathcal{M}(\lambda, h) = (I - (1 + h\lambda) A) =$$

$$= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -1 - h\lambda & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 - h\lambda & 1 & \cdots & 0 & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ 0 & & \cdots & & 1 & 0 \\ 0 & & \cdots & & -1 - h\lambda & 1 \end{bmatrix}$$

where $I \in \mathbb{R}^{(N+1)\times(N+1)}$ is the identity matrix. Let us observe that by using a different method for the ODE approximation, the matrix formulation can be derived in the same way but the matrix $\mathcal{M}(\lambda, h)$ is different: for example, if we chose an implicit method, $\mathcal{M}(\lambda, h)$ will be upper triangular.

The finite N-dimensional problem becomes:

$$\min_{\lambda} J^{DIR}(\lambda, h) = \|\mathbf{u}(\lambda) - \tilde{\mathbf{y}}\|_2^2 \tag{1.24}$$

subject to:

$$\mathcal{M}(\lambda, h)\mathbf{u} = \mathbf{c}(h)$$

where $\tilde{\mathbf{y}} = [\tilde{y}(t_0), \tilde{y}(t_1), ..., \tilde{y}(t_N)]^T = [\tilde{y}_0, \tilde{y}_1, ..., \tilde{y}_N]^T$ the solution of the system (1.16) evaluated on the time grid.

We can observe the difference between the cost functional (1.14) and that one used in this approach $J^{DIR}(\lambda, h)$. Let us define

$$err_{DIR}(\lambda, h) = |J(\lambda) - J^{DIR}(\lambda, h)| \tag{1.25}$$

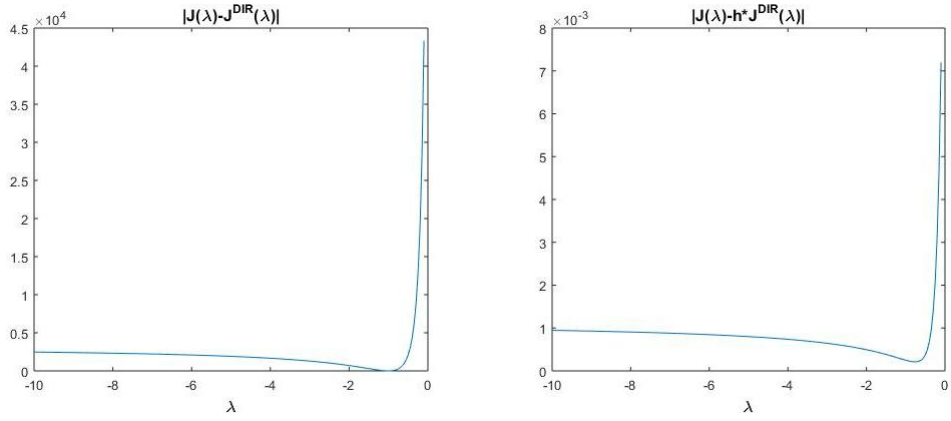

Figure 1.3: Direct Approach: $err_{DIR}(\lambda, h)$ in (1.25) for $h = 10^{-3}$

represented in Figure 1.3 on the left. We have fixed the time step $h = 1e-03$ and $\lambda \in [-10, -0.1]$. As we can observe, the error is very large: in fact, as we will explain in detail in Chapter 3, the norm does not approximate the continuous cost function in (1.14) for construction. Instead, if we compare $J(\lambda)$ with respect to $hJ^{DIR}(\lambda, h)$, Figure 1.3 on the right, the order of error becomes almost h. Further details will be explained in the Chapter 3.

Now let us apply the steepest descent method to solve (1.24), where the descent direction d_k is the gradient of $J^{DIR}(\lambda, h)$:

$$\begin{aligned} d_k = \nabla J^{DIR}(\lambda) = \frac{\partial}{\partial \lambda} \|\mathbf{u}(\lambda) - \widetilde{\mathbf{y}}\|_2^2 = \frac{\partial}{\partial \lambda} \sum_{k=0}^{N} (u_k(\lambda) - \widetilde{y}_k)^2 \\ = 2 \sum_{k=0}^{N} (u_k(\lambda) - \widetilde{y}_k) \frac{\partial}{\partial \lambda} u_k(\lambda) \end{aligned} \tag{1.26}$$

In this case (1.23) yields: $\mathbf{u}(\lambda) = \mathcal{M}(\lambda)^{-1}\mathbf{c}$, then it is possible to compute explicitly the gradient as follows:

$$\begin{aligned} d_k = \frac{\partial}{\partial \lambda} \left[(\mathcal{M}(\lambda)^{-1}\mathbf{c} - \tilde{\mathbf{y}})^T (\mathcal{M}(\lambda)^{-1}\mathbf{c} - \tilde{\mathbf{y}}) \right] = \\ = 2(h\mathcal{M}(\lambda)^{-1} A \mathcal{M}(\lambda)^{-1}\mathbf{c})^T (\mathcal{M}(\lambda)^{-1}\mathbf{c} - \tilde{\mathbf{y}}) \end{aligned} \tag{1.27}$$

Let us observe that, if a non-linear ODE model is considered, the computation of $\nabla J^{DIR}(\lambda, h)$ may be not so immediate, because we can not derive the matrix form and explicit $\mathcal{M}$ as in (1.23); therefore, the gradient could be computed by using the automatic differentiation or approximated, for example, by using the finite difference.

Let us fix the parameters of the algorithm as follows: $h = 1e-03$ for the time discretization, $\alpha_0 = 1$, $\epsilon = 1e-03$ for the tolerance. The results are summarized in

Table 1.1: we consider different values for the starting parameter $\lambda_0 \in [-3, -0.7]$ and report the number of the iterations $\bar{k}$, the final value $\lambda_{\bar{k}}$ which results from the optimization, the type of convergence, the final cost and the absolute error between the optimal solution and the approximation defined as follows: $|\lambda^* - \lambda_{\bar{k}}|$.

Let us observe that for $\lambda_0 \geq -0.4$, the optimization stops far from the optimum $\lambda^* = -1$. This depends on the algorithm we are using for the optimization and on the numerical method for the ODEs: in fact, the first iteration of the steepest descent produces the new approximation of the parameter λ_1 that is located on the left of the minimum, because the gradient value d_0 in λ_0 is very large, then $\lambda_1 = \lambda_0 - \alpha d_0 << \lambda_0$. From the stability analysis of EE [65] it is well known that h must be chose in appropriate way to have the convergence of the method and the bound is: $h < \frac{-2}{\lambda}$. After the first iteration this bound is not satisfied, then EE becomes unstable and does not produce any solution.

Remark 1.3.1. *The problem becomes ill-conditioned when λ_0 increases. Then the choice of the numerical method for the ODEs is crucial for the numerical optimization.*

1.4 Indirect Approach: *Optimize-then-Discretize*

Let us use the Indirect approach to solve the ODE-TIP. In this approach, we need the optimality conditions derived by Optimal Control theory [49] to solve the constrained minimization problem in (1.14)-(1.15) as follows. We write the Hamiltonian associated to (1.14)-(1.15):

$$\mathcal{H}(y, \lambda, \psi) = y^2 - 2y\tilde{y} + \tilde{y}^2 + \psi\left(\lambda y + \frac{1}{2}\right) \tag{1.28}$$

where ψ is the *adjoint variable* that solves the following ODE:

$$\begin{cases} \psi(t)' = -\frac{\partial \mathcal{H}}{\partial y} = -2y(t) + 2\tilde{y(t)} - \psi(t)\lambda \\ \psi(10) = 0 \end{cases} \tag{1.29}$$

where $\tilde{y}(t)$ is given by (1.16). (1.29) is called *adjoint equation*. Let us note that the adjoint equation has to be solved backwards in time.

The gradient of (1.28) [49] is:

$$\frac{d\mathcal{H}}{d\lambda} = \mathcal{H}'(\lambda) = \int_0^{10} \psi(t) y(t) dt. \tag{1.30}$$

First of all we construct the discrete finite-dimensional problem associated with (1.14)-(1.15)-(1.29) to be optimized with respect to λ, then we use the steepest descent method to approximate the minimum. To obtain the finite-dimensional problem we need an ODE solver for the approximation of the constraints (forward in time for (1.15) and backward in time for (1.29)), and a quadrature formula for the integrals in (1.14) and (1.30) (cost and gradient calculation). As example, we choose the Explicit Euler method [65] as ODE solver and the composite trapezoidal rule [65] as quadrature formula. For all schemes we fix $h = \frac{T-t_0}{N}$. Hence let $J^{IND}(\lambda, h)$ be the approximation of $J(\lambda)$, $\mathbf{u} = [u_0, ..., u_N]^T$ the numerical solution of the ODE (1.15) and $\mathbf{v} = [v_0, ..., v_N]^T$ the numerical solution of the adjoint system (1.29). The finite dimensional problem becomes:

$$\begin{aligned} &\min_{\lambda} J^{IND}(\lambda, h) = \sum_{n=0}^{N} w_n (u_n - \tilde{y}_n)^2 \\ &u_0 = 1, \quad u_{n+1} = u_n + h\left(\lambda u_n + \frac{1}{2}\right) \qquad n = 0, ..., N-1 \\ &v_n = 0, \quad v_n = v_{n+1} - h(-2u_n + 2\tilde{y}_n - v_n\lambda) \qquad n = N-1, ..., 0 \end{aligned} \tag{1.31}$$

where w_n, $n = 0, ..., N$, are the weights of the numerical quadrature method [44], such that $\sum_{n=0}^{N} w_n = T - t_0$, and $\tilde{y}_n = \tilde{y}(t_n) \quad \forall n \in \{0, ..., N\}$. Let us observe that the adjoint equation is solved backward in time using a so-called *reflected* method, in particular in this case we use the reflected Explicit Euler method (see Section 2.2 for details).

Furthermore, $\mathcal{H}'(\lambda)$ is approximated by

$$\mathcal{H}'(\lambda, h) = \sum_{n=0}^{N} w_n v_n u_n \tag{1.32}$$

where we have chosen the same composite quadrature rule used for the approximation of the cost function in (1.31).

Let $err_{IND}(\lambda, h)$ be the difference between the cost function and its numerical approximation:

$$err_{IND}(\lambda, h) = |J(\lambda) - J^{IND}(\lambda, h)|. \tag{1.33}$$

In Figure 1.4 we can see the graphic representation of $err_{IND}(\lambda, h)$ as function of λ, fixed the step $h = 1e-03$.

Now let us solve the N-dimensional problem (1.31) applying the steepest descent method, where the descent direction is $d_k = \mathcal{H}'(\lambda_k, h)$. Let us fix the

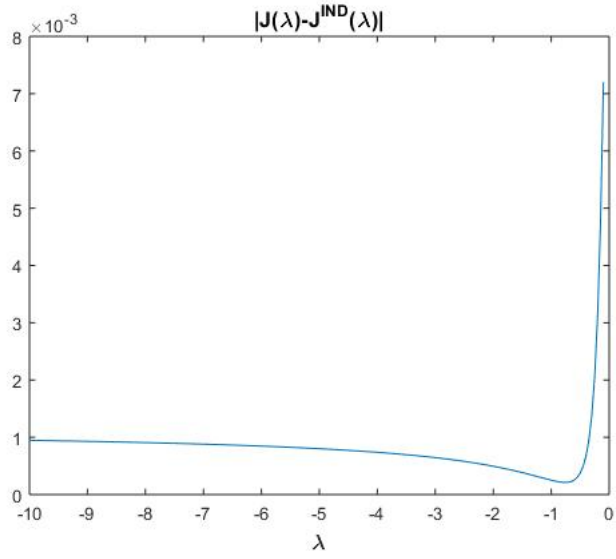

Figure 1.4: Indirect Approach: $err_{IND}(\lambda, h)$ in (1.33) for $h = 1e-03$

parameters of the algorithm: $h = 1e-03$ for the time discretization, $\alpha_0 = 1$, $\epsilon = 0.001$ for the tolerance. The results of the optimization algorithm are summarized in the Table 1.2, where we consider different values for the starting parameter $\lambda_0 \in [-3, -0.4]$ and report the number of iterations $\bar{k}$, the final value $\lambda_{\bar{k}}$ which results from the optimization, the type of convergence, the final cost and the absolute error between the optimal solution and the approximation defined as follows: $|\lambda^* - \lambda_{\bar{k}}|$.

Let us observe that for $\lambda \geq -0.2$ the iterations stop after just one step: in λ_0 the gradient is very large as in the previous case, and the next iteration starts from $\lambda_1 << \lambda_0$ where the gradient is small: $d_1 = -3.0739e-04 < \epsilon$, then the optimization arrests even if λ_1 is far from the minimum $\lambda^* = -1$.

Remark 1.4.1. *The number of iterations $\bar{k}$ becomes very large when λ_0 increases, and the convergence is slower and slower, as we have observed in Section 1.2.*

To sum up, in this Sections we have shown that the two constrained optimization problem (1.31) and (1.24) (arising by the indirect and direct approach respectively) have the following features:

1. (1.31) and (1.24) are different not equivalent discrete problems;

2. the optimization with respect to λ for fixed $h > 0$ by the steepest descent method implies that

λ_0	$\bar{k}$ iterations	$\lambda_{\bar{k}}$	convergence	final cost	absolute error
-3	6	-1.0004	cost	3.1874e-04	4.9898e-04
-2	4	-1.0002	cost	5.7109e-05	2.1116e-04
-1.5	6	-1.0006	cost	3.9511e-04	5.5559e-04
-1	1	-1	gradient	0	0
-0.9	5	-1.0005	cost	2.6235e-04	4.5268e-04
-0.7	4	-0.9994	cost	4.5598e-04	5.9623e-04
-0.6	7	-1.0005	cost	3.2108e-04	5.0081e-04
-0.5	5	-0.9996	cost	2.5847e-04	4.4895e-04
-0.4	1	-3.0782e+04	-	-	-3.078e+04
-0.2	1	-1.4915e+05	-	-	-1.492e+05

Table 1.1: Direct Approach: Results of the optimization algorithm with different starting value λ_0, for $h = 1e-03$

λ_0	iterations $\bar{k}$	$\lambda_{\bar{k}}$	convergence	final cost	absolute error
-3	6	-1.0042	cost	2.2441e-05	0.0042
-2	4	-0.9937	cost	5.0201e-05	0.0063
-1.5	7	-1.0105	cost	1.3907e-04	0.0105
-1	1	-1	gradient	0	0
-0.9	2	-0.9813	cost	4.6381e-04	0.0187
-0.7	5	-1.0006	cost	5.9955e-07	6.3299e-04
-0.6	21	-0.9796	cost	5.5286e-04	0.0204
-0.5	222	-0.9875	cost	2.0273e-04	0.0125
-0.4	1559	-0.9922	cost	7.8176e-05	0.0078
-0.2	1	-149.3725	gradient	1.5395	148.3725

Table 1.2: Indirect Approach: Results of the optimization algorithm with different starting value λ_0, for $h = 1e-03$

  - for (1.24) the problem become ill-conditioned for $\lambda > c_1$, for some $c_2 \in (0.5, 0.4]$
  - for (1.31) the number of iterations increases for $\lambda_0 > -1$ and the optimization fails for $\lambda > c_2$, for some $c_2 \in (0.4, 0.2]$

3. the optimization strongly depend on the numerical method chosen for the

ODEs and the quadrature formula for the integral in the indirect approach.

1.5 Literature review

The comparison between the Direct and Indirect methods is widely discussed in literature, see e.g. [5, 12, 18, 66, 67, 91, 85, 87, 88] and the choice of the approach to use is often determined by problem to be solved and its complexity. As we have seen, the Direct and Indirect methods originate from different philosophies: the direct approach finds the optimal solution rewriting the infinite-dimensional problem as a finite-dimensional problem to be solved by well-known optimization techniques; the indirect approach solves the problem by converting the optimal control problem to a boundary-value problem, then the optimal solution is found by solving a ODEs system.

Remarkable surveys on the numerical methods for trajectory optimization are in particular [5, 66, 87], where both indirect and direct methods are presented and analyzed. Furthermore, in [66], the author discussed important computational issues and described several different software tools for solving optimal control problems.

Each of the two methods have different advantages and disadvantages compared to the other, as studied mostly in [5, 67, 91, 87]. In particular, the direct method presents the following advantage on indirect method: it does not require any a-priori theoretical study, it is more robust and the model can be easily modified; the method is less sensitive to the choice of the initial conditions [91, 87]. On the contrary it is difficult with the direct method to reach the precision provided by the indirect approach. Furthermore, the direct method requires a large amount of memory, and it becomes inefficient if the dimension is too large [87]; sometimes it converges to local minima, which are introduced by the discretization [91]. The advantage of the indirect method is its extremely good numerical accuracy [91, 87]; but in general it is based on the maximum principle, that is just a necessary condition for the optimality; it is not easy to introduce state constraints, because this requires to apply a maximum principle with state constraints (and it is more complicated respect to the standard maximum principle).

There is not a conventional answer for the choice of the method, and it should be guided by the practical problem under consideration and by the experience

[5]. In [12, 18, 91, 88] the authors proposed and demonstrated that a combination of direct and indirect methods is a very promising way to obtain a numerical solution, combining the advantages of both approaches.

The purpose of the thesis is not to decide which method to use for the solution an optimization problem, but rather to provide an analysis of the two approaches from a numerical point of view and to bring out the numerical problems linked to their discretization.

Chapter 2

Discretization issues: ODE case

We have seen the two different well-known different methodologies to solve the ODE-PIP: the Direct Approach and the Indirect Approach. Briefly the Direct approach transforms the optimal control problem into a nonlinear programming problem to be solved numerically by using an optimization method [63]. The Indirect approach uses the theory of the Optimal Control and the Pontryagin's maximum principle [49, 87] to determine the so-called *optimality conditions*, which must be satisfied by the optimal solution. The necessary conditions are two differential equations which arise from the differentiation of a *Hamiltonian function* associated to the problem: the *state equation* (to be solved forward in time) and the *adjoint equation* (to be solved backward in time). These equations are often difficult to solve: generally the problem is nonlinear and therefore do not have analytic solutions. Hence the solutions are approximated numerically and an optimization algorithm is used to approach the minimum. Therefore in both cases a continuous minimization problem is discretized and solved numerically and it is necessary to use numerical tools for the construction and resolution of the finite-dimensional problem:

- a (weighed) norm to define the cost function in the Direct approach;
- a quadrature method for the cost functional in the Indirect approach;
- a numerical optimization algorithm;
- ODEs solver;

We briefly analyze all of them, then we focus on the analysis of the ODEs solver

in two particular cases as explained below.

(weighed) norm: For the Direct approach we have by construction a discrete cost, defined by a norm. Given a PIP in continuous form the use of a (weighted) norm as cost function is certainly the most immediate and easy method: we do not need to approximate any integral and therefore we do not have to use any quadrature formula. The accuracy of the norm can be improved by using, for example, a suitable weighting matrix: in this way we can give more importance to some elements of the experimental data, considered most significant for the identification purposes. When we consider simulated data in the test problems, where we know the exact solution of the parameter identification, we use the classical 2-norm (with $W = I$), because the aim will be to show the characteristics of the problem and the issues derived from the discretization rather than the effective minimization.

Quadrature formula: In this thesis we will not dwell on the choice of the quadrature method for the approximation of the integrals (cost function and its gradient) in the Indirect approach. We will refer to it only for the convergence analysis in the next Chapter, in particular we will be interested in the convergence order of the quadrature formula.

Optimization algorithm: In literature the crucial point in solving the PIP is how to chose the optimization algorithm; in fact, a widely amount of papers concern the numerical methods for optimization: see for example [71, 32, 39, 83]. Moreover, the optimization in the direct approach generally requires the gradient value of the norm: when its value can not be provided exactly (and this also depends, as we have seen, on the numerical method for ODEs), it can be approximated by using the finite differences or it can be generated through the so-called automatic differentiation, see e.g. [34].

ODEs solver: As we have seen, in both approaches the differential equations must be solved, then an suitable numerical method for ODEs must be provided. The choice of the numerical method must be done carefully, since the optimization process also depends on the accuracy of the ODEs solver. In fact, as an example,

in the Direct Approach applied to the TIP, the chosen method determined the failure of the convergence of the algorithm. In particular, we focus on the suitable approximation of ODEs with oscillatory solutions which have an interesting recent application in the electrodeposition process [47, 79]. It is worth noting that, specially in the case of oscillating dynamics, the cost function strongly depends on the accuracy of the numerical method $\mathcal{M}$ in (1.5) that must be able to reproduce carefully the oscillations. In the following Chapters we will normalize both data and simulations, therefore we are interested in methods that minimize the dispersion (phase) error rather than dissipation error. Therefore in Section 2.1 we report a dispersion error analysis for a selection of numerical schemes. Secondly, if the Indirect approach is used, then the numerical resolution of (1.11) requires two numerical methods for the approximation of the ODEs forward and backward in time. In Section 2.2 we present the Reflecting and Transposing Runge-Kutta methods widely used for the so-called *partitioned systems* [73].

2.1 Numerical methods for oscillating solutions

In this Section we discuss the appropriate numerical method to solve the ODE system with oscillating solution that is the constraint of the PIP in (1.1)-(1.2). We start by considering the following linear test problem as prototype of ODE with oscillating solution:

$$w'(t) = i\beta w, \quad t \in [0, T]. \tag{2.1}$$

The exact solution in analytic form is:

$$w(t) = ce^{i\beta t} \tag{2.2}$$

with the parameter c that depends on the initial condition $w(0) = w_0$. If $w_0 = \rho_0 e^{i\phi_0}$, then

$$w(t) = \rho_0 e^{i(\beta t + \phi_0)} = \rho_0(\cos(\beta t + \phi_0) + i\sin(\beta t + \phi_0)) \tag{2.3}$$

$\beta > 0$ is referred to as the *inner frequency* [90], ρ_0 is the *amplitude* of the solution and ϕ_0 is its *phase*. Let us define $u(t) = Re(w(t))$ and $v(t) = Im(w(t))$ for all $t \in [0, T]$. Then the ODE in (2.1) can be written equivalently as the following

linear ODE system:

$$\begin{cases} u'(t) = -\beta v \\ v'(t) = \beta u \\ u(0) = u_0 = \rho_0 \cos(\phi_0),\ v(0) = v_0 = \rho_0 \sin(\phi_0) \end{cases} \tag{2.4}$$

and its analytical solution is:

$$\begin{cases} u(t) = \rho_0 \cos(\beta t + \phi_0) \\ v(t) = \rho_0 \sin(\beta t + \phi_0) \end{cases} \Leftrightarrow \begin{bmatrix} u(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} \cos(\beta t) & -\sin(\beta t) \\ \sin(\beta t) & \cos(\beta t) \end{bmatrix} \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \tag{2.5}$$

Let us consider the time discretization $t_n = nh$ for all $n = 0, 1, 2, ...$, with h time stepsize; let us define the variable $\nu = \beta h \in \mathbb{R}$. Therefore the exact solution (2.3) at each step $t_n = nh$ is given by:

$$w(t_n) = w_0 e^{in\nu} = \rho_0 e^{i(n\nu + \phi_0)}. \tag{2.6}$$

Similarly, the solution in (2.5) evaluated at each t_n is:

$$\begin{bmatrix} u(t_n) \\ v(t_n) \end{bmatrix} = \begin{bmatrix} \cos(n\nu) & -\sin(n\nu) \\ \sin(n\nu) & \cos(n\nu) \end{bmatrix} \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \tag{2.7}$$

We consider the one-step methods for the approximation of (2.1) and (2.4). We are interested in schemes that approximate with a certain degree of accuracy the amplitude and the phase of oscillating solutions. The error on the modulus is called *dissipation error*, and the error on the phase is called the *dispersion error* according to [90]. Here we focus on the following methods: modified Euler's method, trapezoidal rule, explicit Runge-Kutta 4 and the symplectic Euler and Stormer-Verlet methods [37, 36, 65].

The *modified Euler's method* is a 2-stages Runge-Kutta method with order of consistency $p = 2$. The scheme, applied to (2.1), is:

$$w_{n+1} = w_n + h\left[i\beta\left(w_n + \frac{h}{2} i\beta w_n\right)\right] = \left(1 + i\nu - \frac{\nu^2}{2}\right) w_n \quad n = 0, 1, ... \tag{2.8}$$

The term $z(v) = 1 - \frac{\nu^2}{2} + i\nu$ is the so-called *stability function* of the method, that define the stability region where $|z(\nu)| < 1$.

The *trapezoidal rule* is an implicit method; it is A-stable and its order of consistency is $p = 2$. Applying the method to the equation (2.1):

$$w_{n+1} = w_n + \frac{h}{2}\left(i\beta w_{n+1} + i\beta w_n\right) = w_n + \frac{i\nu}{2}\left(w_{n+1} + w_n\right) \quad n = 0, 1, ... \ . \tag{2.9}$$

we can make explicit w_{n+1}:

$$w_{n+1} = \frac{2+i\nu}{2-i\nu} w_n = \frac{4-\nu^2+4i\nu}{4+\nu^2} w_n \quad n = 0, 1, \dots . \tag{2.10}$$

Hence its stability function is: $z(\nu) = \frac{4-\nu^2}{4+\nu^2} + i\frac{4\nu}{4+\nu^2}$.

Runge Kutta 4 is a 4-stages explicit method, and its order of consistency is $p = 4$ [37]. The scheme applied to the equation (2.1) is:

$$\begin{cases} k_1 = i\beta y_n, \\ k_2 = i\beta(y_n + \frac{h}{2}k_1), \\ k_3 = i\beta(y_n + \frac{h}{2}k_2), \qquad \Rightarrow \\ k_4 = i\beta(y_n + \frac{h}{2}k_3), \\ w_{n+1} = w_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{cases} \tag{2.11}$$

$$\Rightarrow w_{n+1} = \left(1 + i\nu - \frac{\nu^2}{2} - i\frac{\nu^3}{6} + \frac{\nu^4}{24}\right) w_n \quad n = 0, 1, \dots$$

with the stability function given by: $z(\nu) = 1 - \frac{\nu^2}{2} + \frac{\nu^4}{24} + i(\nu - \frac{\nu^3}{6})$.

As we have seen the stability function can be written as: $z(\nu) = A_s(\nu) + iB_s(\nu)$, where $A_s(\nu)$ and $B_s(\nu)$ are polynomials in the variable ν defined by the method.

Hence the numerical solution of (2.1) obtained by a one-step method can be written in the form:

$$w_n = z(\nu)^n w_0 = \rho(\nu)^n e^{in\omega(\nu)} w_0 = \rho_0 \rho^n e^{i(n\omega(\nu)+\phi_0)} \tag{2.12}$$

where $\rho = |z|$ is the modulus of the stability function $z(\nu)$ and $\omega = \omega(\nu)$ is its argument.

The *symplectic Euler* method [37] is applied directly to the real system (2.4):

$$\begin{cases} u_{n+1} = u_n + h(\beta v_n) = u_n - \nu v_n \\ v_{n+1} = v_n + h(\beta u_{n+1}) = \nu u_n + \nu u_{n+1} \end{cases} \Rightarrow \begin{cases} u_{n+1} = u_n - \nu v_n \\ v_{n+1} = \nu u_n + (1-\nu^2) v_n \end{cases} \tag{2.13}$$

then we can define the matrix $M(\nu)$ associated to (2.13):

$$M(\nu) = \begin{bmatrix} 1 & -\nu \\ \nu & 1-\nu^2 \end{bmatrix} \tag{2.14}$$

and rewrite (2.13):

$$\begin{bmatrix} u_{n+1} \\ v_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & -\nu \\ \nu & 1-\nu^2 \end{bmatrix} \begin{bmatrix} u_n \\ v_n \end{bmatrix} \quad n = 1, 2, ... \tag{2.15}$$

The *Stormer-Verlet* [36] scheme for the system (2.4) is given by:

$$\begin{cases} v_{n+\frac{1}{2}} = v_n + \frac{h}{2}(\beta u_n) = v_n + \frac{\nu}{2} u_n \\ u_{n+1} = u_n + \frac{h}{2}(-2\beta v_{n+\frac{1}{2}}) = u_n - \nu v_{n+\frac{1}{2}} \qquad \Rightarrow \\ v_{n+1} = v_{n+\frac{1}{2}} + \frac{h}{2}(\beta u_{n+1}) = v_{n+\frac{1}{2}} + \frac{\nu}{2} u_{n+1} \end{cases} \tag{2.16}$$

$$\Rightarrow \begin{cases} u_{n+1} = (1 - \frac{\nu^2}{2}) u_n - \nu v_n \\ v_{n+1} = (\nu - \frac{\nu^3}{4}) u_n + (1 - \frac{\nu^2}{2}) v_n \end{cases}$$

therefore the matrix $M(\nu)$ is:

$$M(\nu) = \begin{bmatrix} 1 - \frac{\nu^2}{2} & -\nu \\ \nu\left(1 - \frac{\nu^2}{4}\right) & 1 - \frac{\nu^2}{2} \end{bmatrix}. \tag{2.17}$$

then:

$$\begin{bmatrix} u_{n+1} \\ v_{n+1} \end{bmatrix} = \begin{bmatrix} 1 - \frac{\nu^2}{2} & -\nu \\ \nu\left(1 - \frac{\nu^2}{4}\right) & 1 - \frac{\nu^2}{2} \end{bmatrix} \begin{bmatrix} u_n \\ v_n \end{bmatrix} \quad n = 1, 2, ... \tag{2.18}$$

Recursively the numerical solution of (2.4) can be written in matrix form as:

$$\begin{bmatrix} u_n \\ v_n \end{bmatrix} = M(\nu)^n \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \quad n = 1, 2, ... \tag{2.19}$$

where $M(\nu) \in \mathbb{R}^{2\times 2}$ is the matrix associated to the method. Let us note that, if the eigenvalues of the matrix M are $(\lambda_M)(\nu)_{+,-} = \rho_M e^{\pm\omega_M} = \rho_M(\cos(\omega_M) \pm i\sin(\omega_M))$, then we have:

$$\cos(\omega_M(\nu)) = \frac{trace(M(\nu))}{2\sqrt{det(M(\nu))}}.$$

If the eigenvalues are real, the definition can be easily extended, as explained in [76].

Let us introduce two errors that allows us to study the proprieties of the numerical solutions: in particular we examine the dispersion and the dissipation of the solutions. To study the *dissipation* we analyze the *dissipation error* defined as follows [90], from (2.12):

$$\delta(\nu) = 1 - \rho(\nu) = 1 - \sqrt{A_s^2(\nu) + B_s^2(\nu)}. \tag{2.20}$$

To examine the dephasing in the solution, i.e. the *dispersion*,we recall the *dispersion error* [90]:

$$\phi(\nu) = \nu - \omega(\nu) = \nu - \arctan\left(\frac{B_s(\nu)}{A_s(\nu)}\right). \tag{2.21}$$

If we consider the matrix form (2.15) or (2.18), it result [90]:

$$\delta(\nu) = 1 - \rho_M(\nu) = 1 - \sqrt{det(M(\nu))}. \tag{2.22}$$

and

$$\phi(\nu) = \nu - \omega_M(\nu) = \nu - \arccos\left(\frac{trace(M(\nu))}{2\sqrt{det(M(\nu))}}\right). \tag{2.23}$$

Definition 2.1.1. *If $\phi(\nu) = O(\nu^{q+1})$ the method is called* dispersive of order q*; if $\delta(\nu) = O(\nu^{r+1})$ the method is called* dissipative of order r *[90].*

Let us analyze the methods for which we have already obtained the stability functions $z(\nu)$.

Modified Euler: Its stability function is $z(\nu) = 1+i\nu-\nu^2/2$; therefore $A_s(\nu) = 1-\nu^2/2$ and $B_s(\nu) = \nu$. Let us compute the errors in (2.21) and (2.20):

$$\begin{aligned}
\phi(\nu) &= \nu - \arctan\left(\frac{2\nu}{2-\nu^2}\right) \approx \nu - \left(\nu + \frac{1}{6}\nu^3 + o(\nu^4)\right) = -\frac{1}{6}\nu^3 + O(\nu^4) \\
\delta(\nu) &= 1 - \sqrt{1+\frac{\nu^4}{4}} \approx 1 - \left(1 + \frac{3}{24}\nu^4 + n(\nu^5)\right) = -\frac{3}{24}\nu^4 + O(\nu^5)
\end{aligned} \tag{2.24}$$

Therefore, the Modified Euler method is dispersive of order $q = 2$ and dissipative of order $r = 3$.

The *trapezoidal rule*: The stability function is $z(\nu) = \frac{4-\nu^2}{4+\nu^2} + i\nu\frac{4}{4+\nu^2}$, then $A_s(\nu) = \frac{4-\nu^2}{4+\nu^2}$ and $B_s(\nu) = \frac{4\nu}{4+\nu^2}$. From the definitions of the errors (2.21) and (2.20), it results:

$$\begin{aligned}
\phi(\nu) &= \nu - \arctan\left(\frac{4\nu}{4-\nu^2}\right) \approx \nu - \left(\nu - \frac{1}{12}\nu^3 + o(\nu^5)\right) = \frac{1}{12}\nu^3 + O(\nu^5) \\
\delta(\nu) &= 1 - \sqrt{\frac{(4-\nu^2)^2}{(4+\nu^2)^2} + \frac{16\nu^4}{(4+\nu^2)^2}} = 0.
\end{aligned} \tag{2.25}$$

Hence, the dispersion order is $q = 2$ and the method is *zero-dissipative*.

Runge-Kutta 4: Its stability function is: $z(\nu) = 1 + i\nu - \frac{\nu^2}{2} - \frac{i\nu^3}{6} + \frac{\nu^4}{24}$, then $A_s(\nu) = 1 - \frac{\nu^2}{2} + \frac{\nu^4}{24}$ and $B_s(\nu) = \nu - \frac{\nu^3}{6}$. The dispersion error and the dissipative

Table 2.1: Order of consistency p, order of dispersion q and order of dissipation r of the methods.

	p	q	r
Symplectic Euler	1	2	∞
Modified Euler	2	2	3
Stormer-Verlet	2	2	∞
trapezoidal rule	2	2	∞
Runge-Kutta 4	4	4	5

error are respectively:

$$\begin{aligned}\phi(\nu) &= \nu - \arctan\left(\frac{24\nu - 4\nu^3}{24 - 12\nu^2 + \nu^4}\right) \approx \nu - \left(\nu - \frac{1}{20}\nu^5 + O(\nu^7)\right) \\ &= \frac{1}{20}\nu^5 + O(\nu^7) \\ \delta(\nu) &= 1 - \sqrt{\left(1 - \frac{\nu^2}{2} + \frac{\nu^4}{24}\right)^2 + \nu^2\left(1 - \frac{\nu^2}{6}\right)^2} \approx \frac{5}{6!}\nu^6 + O(\nu^7).\end{aligned} \tag{2.26}$$

Clearly the order of dispersion is $q = 4$ and the order of dissipation is $r = 5$.

Symplectic Euler: by the matrix $M(\nu)$ in (2.14) we have: $det(M(\nu)) = 1$ and $trace(M(\nu)) = 2 - \nu^2$; hence:

$$\begin{aligned}\phi(\nu) &= \nu - \arccos\left(\frac{2 - \nu^2}{2}\right) \approx \frac{-\nu^3}{24} + O(\nu^5) \\ \delta(\nu) &= 0.\end{aligned} \tag{2.27}$$

Stormer-Verlet: for the matrix $M(\nu)$ associated to the method in (2.17) it result: $det(M(\nu)) = 1$ and $trace(M(\nu)) = 2 - \nu^2$; therefore

$$\begin{aligned}\phi(\nu) &= \nu - \arccos\left(\frac{2 - \nu^2}{2}\right) \approx \frac{-\nu^3}{24} + O(\nu^5) \\ \delta(\nu) &= 0.\end{aligned} \tag{2.28}$$

Hence both methods, Symplectic Euler and Stormer-Verlet are *zero-dissipative* and dispersive of order $q = 2$.

Let us summarize the properties of the methods seen so far in Table 2.1, where p is the order of consistency, q the order of dispersion and r the order of dissipation.

The symplectic methods have no dissipation error, in fact for their preservation proprieties [37] the amplitude of the oscillation in the solution is conserved. But

the orders of dispersion of Symplectic Euler and Stormer-Verlet are less than that of Runge-Kutta 4. As we will see in the next Sections, we will compare the data with the numerical solutions of the ODEs normalizing the data values, such that the amplitude of oscillations is always in $[0, 1]$. We are interested in minimizing the dispersion error. Therefore we use Runge-Kutta 4 as ODE solver because it is explicit and has the largest order of dispersion q.

2.2 Runge-Kutta methods for partitioned systems

In this Section we consider the Runge-Kutta method to solve an ODE system and we define the corresponding reflected and transposed RK methods, examining their construction and meaning.

Let us consider the following generic D-dimensional differential system:

$$\begin{cases} y'(t) = F(t, y) & t \in [t_0, T] \\ y(t_0) = y_0 \end{cases} \tag{2.29}$$

with $F : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$, $y_0 \in \mathbb{R}^d$, $d \geq 1$ (typically $d = 2$).

A RK method with s stages is specified by $s^2 + 2s$ numbers [37]:

$$a_{ij} \quad i, j = 1, ..., s \qquad b_i, c_i \quad i = 1, ..., s \tag{2.30}$$

and finds the approximations y_n of $y(t_n)$, $n = 0, ..., N$, $t_n = t_o + nh$, $h = \frac{T - t_0}{N}$, recursively as follows:

$$\begin{aligned} y_{n+1} &= y_n + h_n \sum_{i=1}^{s} b_i K_{n,i} \\ K_{n,i} &= F(t_n + c_i h_n, Y_{n,i}) \\ Y_{n,i} &= y_n + h_n \sum_{j=1}^{s} a_{ij} K_{n,j} \end{aligned} \tag{2.31}$$

The coefficients b_i, c_i and a_{ij} can be collected in the following *Butcher's Tableau*:

$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array}$$

Scherer and Türke [74] associated with the set of RK coefficients two new sets called the *reflection* and the *transposition* of the original.

The reflected RK has coefficients given by:

$$a^r_{ij} = b_j - a_{ij}, \qquad b^r_i = b_i, \qquad c^r_i = 1 - c_i \quad i,j = 1,...,s \tag{2.32}$$

and the transposed RK has coefficients defined by:

$$a^t_{ij} = b_j a_{ji}/b_i, \qquad b^t_i = b_i, \qquad c^t_i = 1 - c_i \quad i,j = 1,...,s \tag{2.33}$$

The operations of reflection and transposition commute, that is:

$$a^{tr} = a^{rt}_{ij} = b_j - b_j a_{ji}/b_i, \qquad b^{tr} = b^{rt}_i = b_i, \qquad c^{tr} = c^{rt}_i = c_i \quad i,j = 1,...,s \tag{2.34}$$

and both the operation are involutions, that is:

$$(a^r_{ij})^r = a_{ij}, \quad (b^r_j)^r = b_j, \quad (c^r_j)^r = c_j \quad i,j = 1,...,s$$

$$(a^t_{ij})^t = a_{ij}, \quad (b^t_j)^t = b_j, \quad (c^t_j)^t = c_j \quad i,j = 1,...,s$$

For example, here we construct the reflected, transposed and reflected-transposed methods for the Explicit Euler, Crank-Nicolson, Runge-Kutta 3 and Runge-Kutta 4 methods [65, 72], reported in the Tables 2.2, 2.3, 2.4 and 2.5 respectively and used for the numerical simulation in the following Chapter.

0	0
	1

1	1
	1

1	0
	1

0	1
	1

Table 2.2: Explicit Euler: Butcher's Tableau for original, reflected, transposed and reflected-transposed.

0	0	0
1	$\frac{1}{2}$	$\frac{1}{2}$
	$\frac{1}{2}$	$\frac{1}{2}$

1	$\frac{1}{2}$	$\frac{1}{2}$
0	0	0
	$\frac{1}{2}$	$\frac{1}{2}$

1	0	$\frac{1}{2}$
0	0	$\frac{1}{2}$
	$\frac{1}{2}$	$\frac{1}{2}$

0	$\frac{1}{2}$	0
1	$\frac{1}{2}$	0
	$\frac{1}{2}$	$\frac{1}{2}$

Table 2.3: Crank-Nicolson: Butcher's Tableau for original, reflected, transposed and reflected-transposed.

What does the reflection and transposition mean?

The interpretation of reflection is well known [37]: a step of length $-h$ by the reflected RK method inverts the transform $y_n \mapsto y_{n+1}$ performed with a step of

0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0
1	-1	2	0
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

1	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$
$\frac{1}{2}$	$-\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{6}$
0	$\frac{7}{6}$	$-\frac{4}{3}$	$\frac{1}{6}$
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

1	0	2	-1
$\frac{1}{2}$	0	0	$\frac{1}{2}$
0	0	0	0
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

0	$\frac{1}{6}$	$\frac{4}{3}$	$\frac{7}{6}$
$\frac{1}{2}$	$\frac{1}{6}$	$\frac{2}{3}$	$-\frac{1}{3}$
0	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

Table 2.4: Runge-Kutta 3: Butcher's Tableau for original, reflected, transposed and reflected-transposed.

0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
1	0	0	1	0
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

1	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$
$\frac{1}{2}$	$-\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$
$\frac{1}{2}$	$\frac{1}{6}$	$-\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{6}$
0	$\frac{1}{6}$	$\frac{1}{3}$	-$\frac{2}{3}$	$\frac{1}{6}$
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

1	0	1	0	0
$\frac{1}{2}$	0	0	$\frac{1}{2}$	0
$\frac{1}{2}$	0	0	0	$\frac{1}{2}$
0	0	0	0	0
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

0	$\frac{1}{6}$	$-\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{6}$
$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{3}$	$-\frac{1}{6}$	$\frac{1}{6}$
$\frac{1}{2}$	$\frac{1}{6}$	$-\frac{1}{3}$	$\frac{1}{3}$	$-\frac{1}{3}$
1	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Table 2.5: Runge-Kutta 4: Butcher's Tableau for original, reflected, transposed and reflected-transposed.

length h by the original method. The reflection preserves the covergence order of the original method, as proved in [73].

The transposition is useful in order to construct a symplectic partitioned Runge-Kutta out of a given RK method, as explained in [72]. In some application the vector y in the ODE system (2.29) appears partitioned into two blocks: $y = [x^T, z^T]^T$, $x \in \mathbb{R}^{d-p}$, $z \in \mathbb{R}^p$, $d > p$. As an example we can consider the Hamiltonian problems, where $d = D/2$. Let us denote by $F = [f^T, g^T]^T$, $f \in \mathbb{R}^{d-p}$, $g \in \mathbb{R}^p$, the partitioning of F induced by the partitioning $[x^T, z^T]^T$ of y, so that

the ODE system in (2.29) is given by:

$$\frac{d}{dt}x = f(x,z,t) \qquad \frac{d}{dt}z = g(x,z,t). \tag{2.35}$$

In this case it may make sense to use a set of coefficients (2.30) for the integration of the block x, and a second set

$$A_{ij} \quad i,j = 1,...,s \qquad B_i, C_i \quad i = 1,...,s \tag{2.36}$$

for the integration of z. The overall method is called *Partitioned Runge-Kutta (PRK) scheme*. Therefore the PRK method, for the ODE system (2.35), becomes:

$$\begin{aligned} x_{n+1} &= x_n + h\sum_{i=1}^{s} b_i k_{n,i}, \qquad z_{n+1} = z_n + h\sum_{i=1}^{s} B_i l_{n,i}, \quad n = 0,...,N-1 \\ k_{n,i} &= f(X_{n,i}, Z_{n,i}, t_n + c_i h), \qquad l_{n,i} = g(X_{n,i}, Z_{n,i}, t_n + C_i h), \\ X_{n,i} &= x_n + h\sum_{j=1}^{s} a_{ij} k_{n,j}, \qquad Z_{n,i} = z_n + h\sum_{j=1}^{s} C_{ij} l_{n,j}. \end{aligned} \tag{2.37}$$

Clearly an RK scheme may be regarded as a particular case of PRK method where the two sets (2.30)-(2.36) coincide.

Remark 2.2.1. *It is important to note that, if the PRK scheme* (2.30)-(2.36) *has order q, then the RK scheme with coefficients* (2.30) *and the RK scheme with coefficients* (2.36) *have both order q. The converse is not true: if* (2.30) *and* (2.36) *are coefficient of two RK schemes of order q, then the combined PRK scheme may have order less than q.*

For a PRK scheme the following Theorem holds:

Theorem 2.2.1 ([72]). *Assume that $I(\cdot,\cdot)$ is a real-valued bilinear map on $\mathbb{R}^p \times \mathbb{R}^{d-p}$ such that the solution $y(t) = [x(t)^T, z(t)^T]^T$ of (2.29), (2.35), satisfies*

$$\frac{d}{dt}I(x(t), z(t)) \equiv 0 \qquad \forall t. \tag{2.38}$$

If between the coefficients of the PRK scheme hold:

$$\begin{aligned} &b_i = B_i \quad i = 1,...,s \\ &c_i = C_i \quad i = 1,...,s \\ &b_i A_{ij} + B_j a_{ij} - b_i B_j \quad i,j = 1,...,s \end{aligned} \tag{2.39}$$

then, for each PRK trajectory, $I(x_n, z_n) = cost$ is independent of n.

Proof. Refer to [72]

It can be proved that for all I and all partitioned differential systems the conditions (2.39) to preserve after the discretization the continuous invariant property in (2.38).

Furthermore, for the preservation of a symplectic structure, the following theorem can be proved:

Theorem 2.2.2 ([72]). *Assume that the system* (2.35) *is Hamiltonian. The relations* (2.39) *garantee that the mapping* $(x_n, z_n) \mapsto (x_{n+1}, z_{n+1})$ *is symplectic.*

Proof. Refer to [72]

The conditions (2.39) are essentially necessary for symplectiness.

Let us observe that the conditions in (2.39) indeed coincide with the relations in (2.34), that is the construction of the reflected-transposed of a RK method. In fact the relations of the coefficients b_i and c_i is the same in (2.34) and in (2.39); the relations of a_{ij} holds as follows:

$$\begin{aligned} A_{ij} &= B_j - B_j a_{ij}/b_i \quad i,j = 1,...,s \\ &\Downarrow (B_i = b_i) \\ A_{ij} &= b_j - b_j a_{ij}/b_i \quad i,j = 1,...,s \\ &\Downarrow (2.34) \\ A_{ij} &= a^{rt}_{ij} \quad i,j = 1,...,s \end{aligned} \tag{2.40}$$

A Hamiltonian system is by definition a partitioned ODE system: it is completely described by a scalar function $H(x,z)$. The evolution equations, that form the partitioned ODE system, are given by the *Hamilton's equations*:

$$\begin{cases} x'(t) = \frac{dx}{dt} = \frac{\partial H}{\partial z}, \\ z'(t) = \frac{dz}{dt} = -\frac{\partial H}{\partial x}. \end{cases} \tag{2.41}$$

It is well-known that the Hamiltonian function is a constant of the motion [73].

On the other hand, also an optimal control problem can be set in partitioned form where the state equation and the adjoint equation, obtained in the indirect approach, can be combine as a single partitioned system as (2.35).

In fact, as we can see the systems (1.9) and (2.41) are in the same form. Therefore, we can use a RK method (2.30) to solve the state equation and a

different RK method (2.36) to solve the adjoint, so that together form a PRK scheme. In order to preserve the Hamiltonian in (1.7), the coefficients have to satisfy the relations in (2.39), so that the method for the adjoint equation must be the reflected-transposed of the method used for the state equation.

But we have seen that the adjoint equation is to be solved backward in time, hence the preservation requires that such backward integration of p be performed with the transposition of the coefficients used to propagate y forward in time, as explained in [72].

In general, the result that the transposition does not preserve the order of convergence, as proved in [74]. A necessary and sufficient condition to preserve the order is that the original method satisfies a certain condition denoted $D(k)$ [14], which establishes a relation among the coefficient of the Buthcher's tableau.

Chapter 3

Comparison between Direct and Indirect Approaches

The choice of the numerical method for the ODE changes the properties of the cost function and its gradient of the finite-dimensional problem deriving from the discretizaion of the continuous PIP, as well as the quadrature formula for the approximation of the integrals in the Indirect approach. In this Chapter we study the convergence properties of the cost functions and the gradients to their continuous counterpart, and we show the convergence results numerically for the TIP studied in the Chapter 1.

3.1 Analysis of cost functions

As we have seen in Sections 1.3-1.4 for the TIP, the approaches yield two different discretization of the cost functional $J(\lambda)$ (and two different constrained optimization problem in finite dimension) as follows:

- The Direct approach is based on the minimization of the 2-norm:

$$J^{DIR}(\lambda, h) = \|\mathbf{u} - \tilde{\mathbf{y}}\|_2^2 . \tag{3.1}$$

- In the Indirect approach we have approximated the integral in (1.14) by the quadrature formula:

$$J^{IND}(\lambda, h) = \sum_{k=0}^{N} w_k (u_k(\lambda) - \tilde{y}_k)^2. \tag{3.2}$$

Note that (1.14), the original minimization function, corresponds to the L^2 norm of $f(\lambda,t) = (y(\lambda,t) - \tilde{y}(t))$. By discretizing this continuous norm by the composite rectangle rule we have:

$$J(\lambda) = \|f(\lambda,t)\|^2_{L^2} \approx \sum_{n=0}^{N} h f_i^2(\lambda) = h\|f(\lambda)\|^2_{2,h} = h J_c^{DIR}(\lambda,t) \tag{3.3}$$

where $f_i(\lambda) = f(\lambda,t_i)$ and $J_c^{DIR}(\lambda,t) = \|\mathbf{y} - \tilde{\mathbf{y}}\|_2^2$ is the discrete cost in (3.1) with the exact solution of (1.15) evaluated on the mesh grid defined by $h = \frac{T-t_0}{N}$. Therefore $J_c^{DIR}(\lambda,h)$ is the approximation of the cost in (1.14) divided by h. For this reason when $h \to 0$, $J^{DIR}(\lambda,h)$ does not converge to the exact cost $J(\lambda)$.

In Figure (1.4)-(1.3) we have seen the errors err_{IND} and err_{DIR} as function of λ, fixing the discrete parameter h. Let us now analyze how these errors changes when we use different methods to solve the ODE system (1.15) and different values of the parameter h.

3.1.1 Cost in Direct Approach

Proposition 3.1.1. *Let $J^{DIR}(\lambda,h)$ be the cost functional in the Direct approach in (3.1) and $h = \frac{T-t_0}{N}$. Let q be the convergence order of the numerical method used to solve the ODE. For all λ_0 it results:*

$$\begin{aligned} &err_{DIR}(\lambda,h) = |J(\lambda) - J^{DIR}(\lambda,h)| \leq \\ &K(\lambda,T) + C_{ode}(T)h^{q-1} + C_{ode}(T)h^{2q-1} + C_{ode}^2 h^{2q}. \end{aligned} \tag{3.4}$$

Proof. Let $J_c^{DIR}(\lambda,h)$ be the cost functional in the Direct approach, seen in (3.3), where the solution of the system (1.15) is analytically computed and evaluated on the mesh grid: $\mathbf{y} = [y(t_0), y(t_1), ..., y(t_N)]^T = [y_0, y_1, ..., y_N]^T$. Let $\tilde{\mathbf{y}} = [\tilde{y}(t_0), \tilde{y}(t_1), ..., \tilde{y}(t_N)]^T = [\tilde{y}_0, \tilde{y}_1, ..., \tilde{y}_N]^T$. It It results:

$$\begin{aligned} &J^{DIR}(\lambda,h) \overset{(3.1)}{=} \sum_{k=0}^{N}(u_k(\lambda) - \tilde{y}_k)^2 \approx \sum_{k=0}^{N}(y_k(\lambda) + C_{ode}h^q - \tilde{y}_k)^2 = \\ &\sum_{k=0}^{N}(y_k(\lambda) - \tilde{y}_k)^2 + 2\sum_{k=0}^{N} C_{ode}h^q(y_k(\lambda) - \tilde{y}_k) + \sum_{k=0}^{N} C_{ode}^2 h^{2q} = \\ &J_c^{DIR}(\lambda,h) + 2\sum_{k=0}^{N} C_{ode}h^q(y_k(\lambda) - \tilde{y}_k) + \sum_{k=0}^{N} C_{ode}^2 h^{2q} \leq \\ &J_c^{DIR}(\lambda,h) + 2NC_{ode}h^q \|\mathbf{y}(\lambda) - \tilde{\mathbf{y}}\|_\infty + (N+1)C_{ode}^2 h^{2q} = \\ &J_c^{DIR}(\lambda,h) + 2(T-t_0)C_{ode}h^{q-1} \|\mathbf{y}(\lambda) - \tilde{\mathbf{y}}\|_\infty + \\ &(T-t_0)C_{ode}h^{2q-1} + C_{ode}^2 h^{2q}. \end{aligned} \tag{3.5}$$

Therefore:

$$
\begin{aligned}
&err_{DIR}(\lambda,h) = |J(\lambda) - J^{DIR}(\lambda,h)| = \\
&|J(\lambda) - J_c^{DIR}(\lambda,h) + J_c^{DIR}(\lambda,h) - J^{DIR}(\lambda,h)| \leq \\
&\underbrace{|J(\lambda) - J_c^{DIR}(\lambda,h)|}_{E_{n2}\text{: error ``2-norm''}} + \underbrace{|J_c^{DIR}(\lambda,h) - J^{DIR}(\lambda,h)|}_{E_{ode}\text{: error numerical method}} = \\
&\frac{T-t_0}{2}K(\lambda) + 2(T-t_0)C_{ode}h^{q-1}\left\|\mathbf{y}(\lambda) - \tilde{\mathbf{y}}\right\|_\infty + \\
&(T-t_0)C_{ode}h^{2q-1} + C_{ode}^2 h^{2q}.
\end{aligned}
\tag{3.6}
$$

where $K(\lambda)$ is a constant that depends on the value of λ and the length of the vectors. □

Let us say:

$$E_{n2}(\lambda,h) = \frac{T-t_0}{2}K(\lambda) \tag{3.7}$$

denotes the error due to the L-2 discrete norm, and

$$E_{ode}(\lambda,h) = 2(T-t_0)C_{ode}h^{q-1}\left\|\mathbf{y}(\lambda) - \tilde{\mathbf{y}}\right\|_\infty + (T-t_0)C_{ode}h^{2q-1} + C_{ode}^2 h^{2q} \tag{3.8}$$

the contribution to the error of the ODEs method. As we can observe, the global error depends on λ, the length of the integration interval and the value of the step h. Furthermore, $Err_{ode}(\lambda,h)$ vanish only if the convergence order of the ODEs method is greater that 1. From this analysis we expect that if we use Explicit Euler, this error do not vanish for $h \to 0$. Let us compute the errors for different values of λ and represent their behavior; the results are shown in Figure 3.1: as we can see the $err_{DIR}(\lambda,h)$ does not vanish when $h \to 0$, as well as the error due by the 2-norm $E_{n2}(\lambda,h)$ and $E_{ode}(\lambda,h)$.

Let us use alternative methods to solve the ODE: for example we consider Crank-Nicolson (of order $q=2$) and Runge-Kutta 4 (of order $q=4$). The results are represented in the Figures 3.2-3.3 respectively. As we have shown the error E_{ode} goes to zero and the convergence order is that one of the numerical method used minus 1.

As we expect E_{n2} never goes to zero. If we take $hJ^{DIR}(\lambda,h)$, that is we multiply by h our cost function, we can observe that the covergence order of the total error is 1 as the covergence order of the composite rectangle rule, as shown in Figure 3.4.

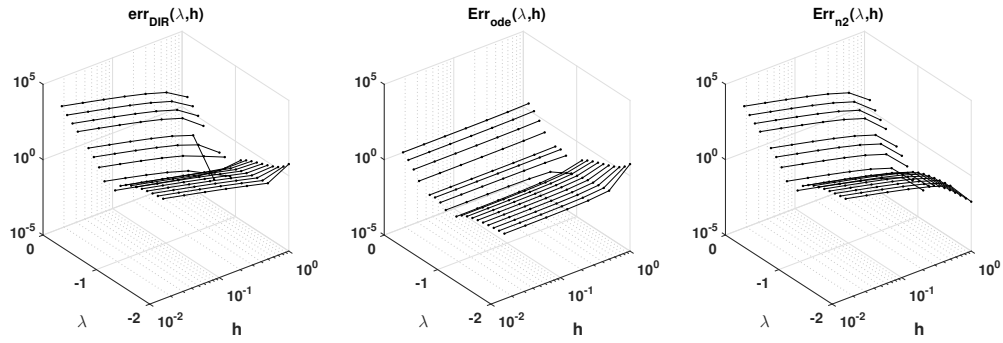

Figure 3.1: Direct Approach: The method used to solve the ODEs is Explicit Euler. From the left to the right the plots represent: $err_{DIR}(\lambda, h)$ in (3.4), $E_{ode}(\lambda, h)$ in (3.8) and $Err_{n2}(\lambda, h)$ in (3.7), for different values of $\lambda \in [-2, -0.1]$, with $\lambda \neq -1, -0.5$. As we can observe $\forall\lambda$ the total error grows as well as the $Err_{n2}(\lambda, h)$, and this confirms the theoretical results. Err_{ode} remains constant $\forall h_t$, with the exception of the first value of h_t for $\lambda < -1$. The errors values depend on the λ: they decrease for $\lambda \to -1$.

Clearly the errors have different behavior in the different values of λ: as we can see from the previous Figures, the error $err_{DIR}(\lambda, h)$ decreases when $\lambda \to -1$ but begins to increase when $\lambda \to 0$.

3.1.2 Cost in the Indirect Approach

Proposition 3.1.2. *Let now $J^{IND}(\lambda, h)$ be the cost functional in the Indirect approach in (3.2), $h = \frac{T-t_0}{N}$. Let q be the convergence order of the numerical method used to solve the ODE and r the convergence order of a composite quadrature formula to approximate the integral. It results:*

$$\begin{aligned} &err_{IND}(\lambda, h) = |J(\lambda) - J^{IND}(\lambda, h)| \leq \\ &Z(\lambda)h^r + C_{ode}(T)h^q + C^2_{ode}(T)h^{2q}. \end{aligned} \tag{3.9}$$

Proof. Let $J^{IND}_c(\lambda, h)$ be the cost functional in the Indirect approach where the solution of the system (1.15) is analytically computed evaluated on the mesh grid: $\mathbf{y} = [y(t_0), y(t_1), ..., y(t_N)]^T = [y_0, y_1, ..., y_N]^T$. Let $\tilde{\mathbf{y}} = [\tilde{y}(t_0), \tilde{y}(t_1), ..., \tilde{y}(t_N)]^T =$

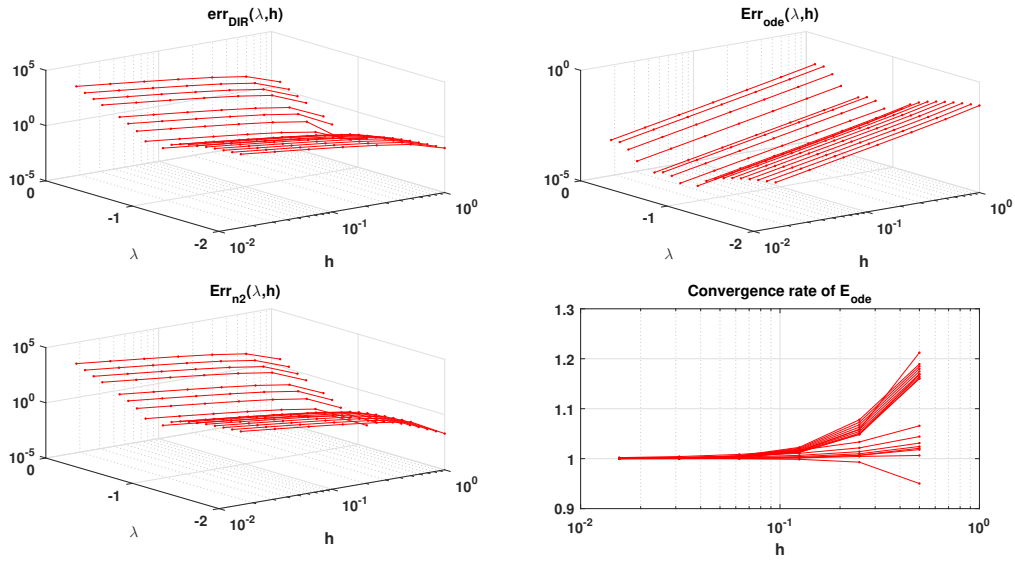

Figure 3.2: Direct Approach: The method used to solve the ODE is Crank-Nicolson, with $\lambda \in [-2, -0.1]$, $\lambda \neq -1, -0.5$. The plot on the top left represents $err_{DIR}(\lambda, h)$ in (3.4). Plot at the top right is $E_{ode}(\lambda, h)$ in (3.8); plot at the bottom left represent the error due by the 2-norm in (3.7) and the plot on the bottom right is the convergence order p of E_{ode}. As we can observe, also in this case the global error grows, following the behavior of Err_{n2}. Err_{ode} decreases and the convergence order reflects the theoretical result.

$[\tilde{y}_0, \tilde{y}_1, ..., \tilde{y}_N]^T$. It results:

$$
\begin{aligned}
&J^{IND}(\lambda, h) \overset{(3.2)}{=} \sum_{k=0}^{N} w_k (u_k(\lambda) - \tilde{y}_k)^2 \approx \sum_{k=0}^{N} w_k (y_k(\lambda) + C_{ode} h^q - \tilde{y}_k)^2 = \\
&\sum_{k=0}^{N} w_k (y_k(\lambda) - \tilde{y}_k)^2 + 2 \sum_{k=0}^{N} w_k C_{ode} h^q (y_k(\lambda) - \tilde{y}_k) + \sum_{k=0}^{N} w_k C_{ode}^2 h^{2q} = \\
&J_c^{IND}(\lambda, h) + 2 \sum_{k=0}^{N} w_k C_{ode} h^q (y_k(\lambda) - \tilde{y}_k) + \sum_{k=0}^{N} w_k C_{ode}^2 h^{2q} \leq \\
&J_c^{IND}(\lambda, h) + 2h^q \left\| \mathbf{y}(\lambda) - \tilde{\mathbf{y}} \right\|_\infty \sum_{k=0}^{N} w_k + C_{ode}^2 h^{2q} \sum_{k=0}^{N} w_k = \\
&J_c^{IND}(\lambda, h) + 2(T - t_0)^2 C_{ode} h^q \left\| \mathbf{y}(\lambda) - \tilde{\mathbf{y}} \right\|_\infty + (T - t_0) C_{ode}^2 h^{2q}.
\end{aligned}
\tag{3.10}
$$

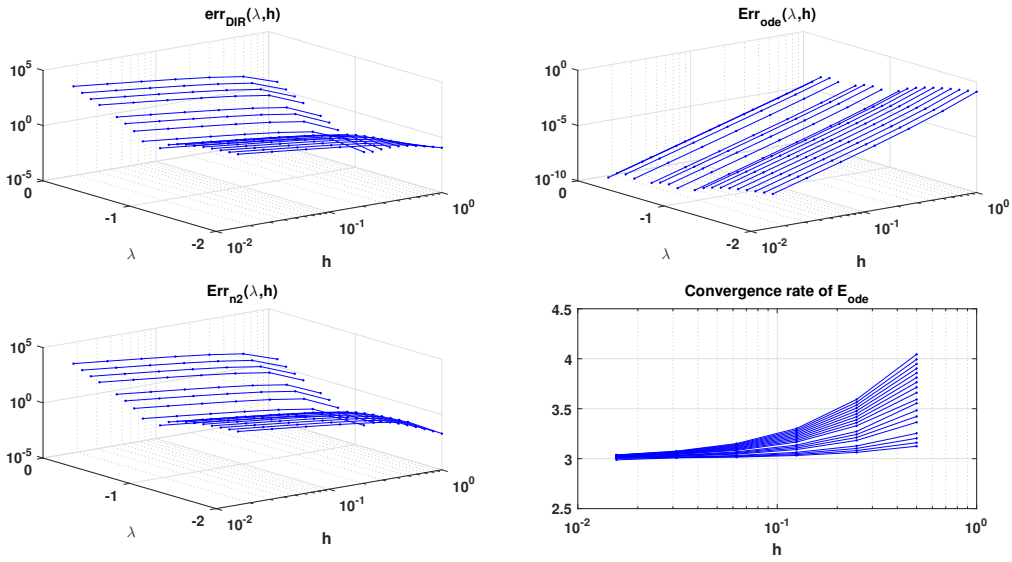

Figure 3.3: Direct Approach: The method used to solve the ODE is Runge-Kutta 4, with $\lambda \in [-2, -0.1]$, $\lambda \neq -1, -0.5$. The plot on the top left represents $err_{DIR}(\lambda, h)$ in (3.4). Plot at the top right is $E_{ode}(\lambda, h)$ in (3.8); plot at the bottom left represent the error due by the 2-norm in (3.7) and the plot on the bottom right is the convergence order p of E_{ode}. As we can observe, also in this case the global error grows, following the behavior of Err_{n2}. Err_{ode} decreases and the convergence order reflects the theoretical result.

Therefore:

$$\begin{aligned} &err_{IND}(\lambda, h) = |J(\lambda) - J^{IND}(\lambda, h)| = \\ &|J(\lambda) - J_c^{IND}(\lambda, h) + J_c^{IND}(\lambda, h) - J^{IND}(\lambda, h)| \leq \\ &\underbrace{|J(\lambda) - J_c^{IND}(\lambda, h)|}_{E_{quadr}\text{: error quadrature}} + \underbrace{|J_c^{IND}(\lambda, h) - J^{IND}(\lambda, h)|}_{E_{ode}\text{: error numerical method}} = \\ &Z(\lambda)h^r + 2(T - t_0)^2 C_{ode} h^q \left\|\mathbf{y}(\lambda) - \tilde{\mathbf{y}}\right\|_\infty + (T - t_0) C_{ode}^2 h^{2q}. \end{aligned} \tag{3.11}$$

□

Let us call

$$E_{quadr}(\lambda, h) = Z(\lambda)h^r$$

and

$$E_{ode}(\lambda, h) = 2(T - t_0)^2 C_{ode} h^q \left\|\mathbf{y}(\lambda) - \tilde{\mathbf{y}}\right\|_\infty + (T - t_0) C_{ode}^2 h^{2q}.$$

In this case the $err_{IND}(\lambda, h)$ coverges to zero and the covergence order will

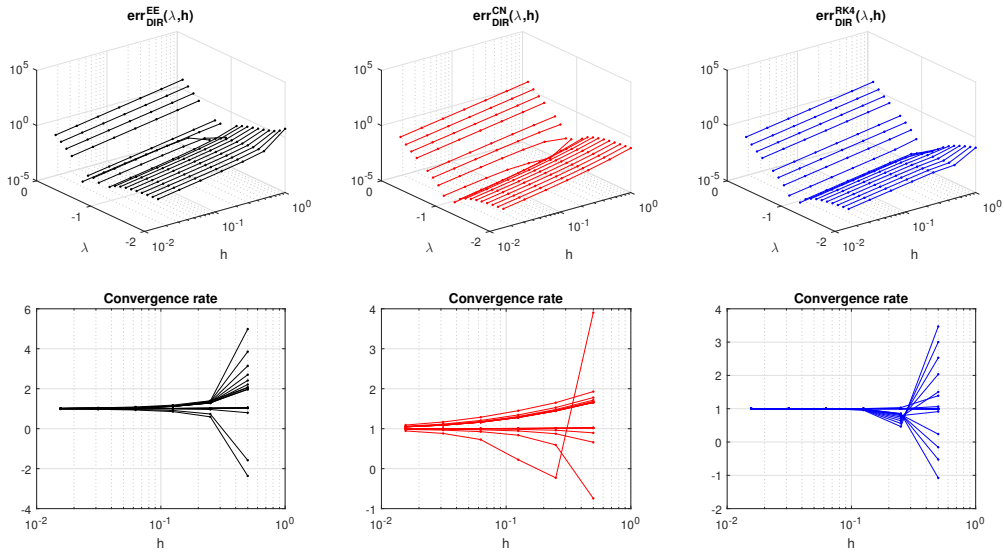

Figure 3.4: Direct Approach: $\lambda \in [-2, -0.1]$, $\lambda \neq -1, -0.5$. The method used to solve the ODE is Explicit Euler (left), Crank-Nicolson (center) and Runke-Kutta 4 (right). On the bottom the respective convergence orders are represented.

be

$$p = \min\{r, q\}. \tag{3.12}$$

In Figures (3.5), (3.6) and (3.7) we report the error $err_{IND}(\lambda, h)$ in logarithmic scale and the covergence order p. The numerical method to solve the ODE are Explicit Euler ($q = 1$), Crank-Nicolson ($q = 2$) and Runge Kutta 4 ($q = 4$) respectively; in all case the quadrature formula used in composite trapezoidal rule ($r = 2$). The final value of the convergence order p coincides with our estimation (3.12).

Also in this case the error $err_{IND}(\lambda, h)$ has an evident dependence on the parameter λ and its value increases when $\lambda \to 0$.

To sum up, in this Section we have studied the cost functions in the Direct and Indirect approaches, by varying the parameters λ and h. In particular we analyzed the two errors $err_{DIR}(\lambda, h)$ end $err_{IND}(\lambda, h)$ given by the differences between the analytical cost $J(\lambda)$ and its approximations $J^{DIR(\lambda,h)}$ and $J^{IND}(\lambda, h)$ respectively. As we have seen that errors are the sum of two contributions: one given by the numerical approximation of the integral in (1.14) and a second one deriving from the use of a numerical solver for the ODE in (1.15):

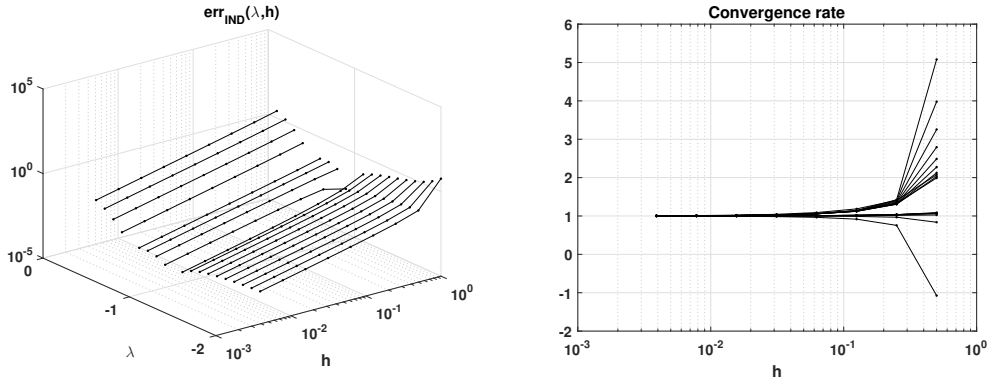

Figure 3.5: Indirect Approach: The method used to solve the ODE is Explicit Euler, with $\lambda \in [-2, -0.1]$, $\lambda \neq -1, -0.5$. The plot on the left represent $err_{DIR}(\lambda, h)$ in (3.9). Plot on the right shows the covergence order p (3.12).

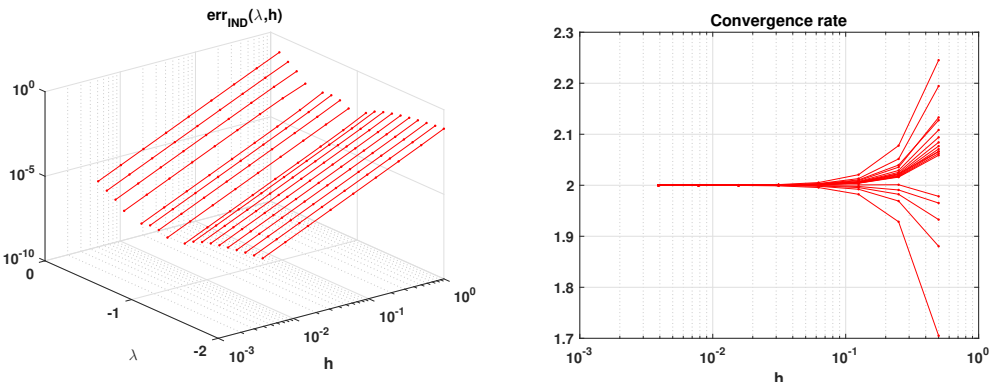

Figure 3.6: Indirect Approach: The method used to solve the ODE is Crank-Nicolson, with $\lambda \in [-2, -0.1]$, $\lambda \neq -1, -0.5$. The plot on the left represent $err_{DIR}(\lambda, h)$ in (3.9). Plot on the right shows the covergence order p (3.12).

- $err_{DIR}(\lambda, h) = E_{n2}(\lambda, h) + E_{ode}(\lambda, h)$;
- $err_{IND}(\lambda, h) = E_{quadr}(\lambda, h) + E_{ode}(\lambda, h)$.

In both the approaches $E_{ode}(\lambda, h) \to 0$ for $h \to 0 \forall \lambda$, with a convergence order p that depends on the convergence order q of the numerical method used for the ODE:

- Direct approach: $p = q - 1$;
- Indrect approach: $p = q$.

Furthermore, the different approximation of the cost in (1.14) implies:

1. Direct approach: $E_{n2} \not\to 0$ for $h \to 0$;

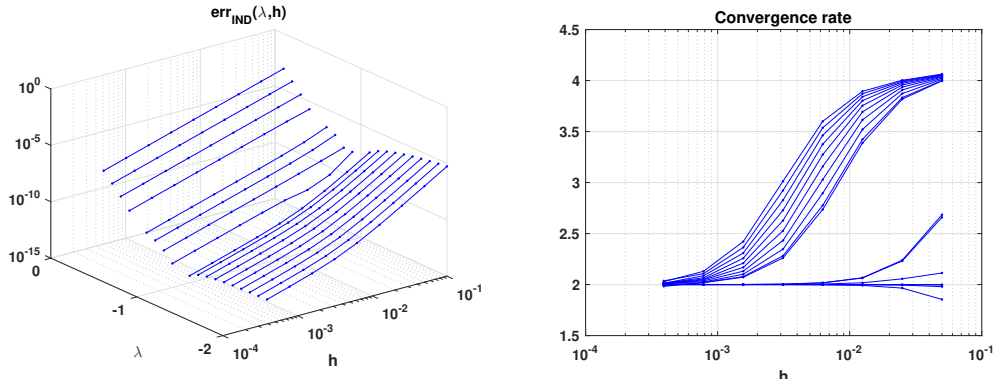

Figure 3.7: Indirect Approach: The method used to solve the ODE is Runge-Kutta 4, with $\lambda \in [-2, -0.1]$, $\lambda \neq -1, -0.5$. The plot on the left represent $err_{DIR}(\lambda, h)$ in (3.9). Plot on the right shows the covergence order p (3.12).

2. Indirect approach: $E_{quadr} \to 0$ for $h \to 0$ with a convergence order r. Therefore, as we have seen the convergence order of $err_{IND}(\lambda, h)$ is $p = \min\{q, r\}$ in (3.12): if the convergence order of the quadrature formula in small, then the implementation of a ODE solver with high convergence order is useless.

3.2 Gradient analysis

As we have seen, in the Direct and Indirect methods we had to solve a minimization problem in finite dimension and we used the steepest descent method to search for the minimum. Therefore in both approaches the values of the gradients were required for the minimization. For the test problem we know the analytical form of the gradient of $J(\lambda)$, given in (1.21). We analyze how the gradient is computed in Direct and Indirect approach respectively, then we will study the error between the gradient (1.21) and its discretizations.

3.2.1 Gradient in the Direct Approach

From the equation (1.26) and with similar argument in (3.3), it results:

$$\frac{dJ(\lambda)}{d\lambda} = \int_0^T 2(y(\lambda,t) - \widetilde{y}(t))\frac{\partial}{\partial\lambda}y(\lambda,t) \approx h\sum_{k=0}^{N} 2(y_k(\lambda) - \widetilde{y}_k)\frac{\partial}{\partial\lambda}y_k(\lambda) = h\frac{\partial}{\partial\lambda}J_c^{DIR}(\lambda,h) \tag{3.13}$$

then $\nabla J_c^{DIR}(\lambda, h)$ is the approximation of the gradient $\nabla J(\lambda)$ by the composite rectangle rule divided by h. Therefore we expect that $\nabla J_c^{DIR}(\lambda, h)$ does not converge to $\nabla J(\lambda)$ for $h \to 0$, as we have seen in Section 3.1.1 for the cost function.

Proposition 3.2.1. *Let $\nabla J^{DIR}(\lambda, h)$ be the gradient in the direct approach and $h = \frac{T-t_0}{N}$. Let q be the convergence order for the method used for the ODE. It follows:*

$$\begin{aligned} &err_{\nabla J^{DIR}}(\lambda, h) = |\ \nabla J(\lambda) - \nabla J^{DIR}(\lambda, h)| \leq \\ &K(\lambda, T) + \frac{d}{d\lambda} C_{ode}(T)[h^{q-1} + C_{ode}(T)h^{2q-1}] + C_{ode}(T)h^{q-1}[1 + h^{q+1}]. \end{aligned} \tag{3.14}$$

Proof. Let $\nabla J_c^{DIR}(\lambda, h)$ be the approximation of the gradient, $\mathbf{y}(\lambda) = [y(t_0), ..., y(t_N)]^T = [y_0, ..., y_N]^T$ are analytically computed and evaluated on the mesh grid. Let $\mathbf{u}(\lambda) = [u_0, u_1, ..., u_N]^T$ be the numerical approximation of $\mathbf{y}$. It result:

$$\begin{aligned} &\nabla J^{DIR}(\lambda, h) = 2\sum_{k=0}^{N}(u_k(\lambda) - \widetilde{y}_k)\frac{d}{d\lambda}u_k(\lambda) \approx \\ &2\sum_{k=0}^{N}(y_k(\lambda) + C_{ode}h^q - \widetilde{y}_k)\frac{d}{d\lambda}(y_k(\lambda) + C_{ode}h^q) = \\ &2\sum_{k=0}^{N}(y_k(\lambda) - \widetilde{y}_k)\frac{d}{d\lambda}(y_k(\lambda)) + 2\frac{d}{d\lambda}(C_{ode}h^q)\sum_{k=0}^{N}(y_k(\lambda) - \widetilde{y}_k) + \\ &2C_{ode}h^q\sum_{k=0}^{N}\frac{d}{d\lambda}(y_k(\lambda) + C_{ode}h^q) \leq \\ &\nabla J_c^{DIR} + 2Nh^q\|\mathbf{y}(\lambda) - \widetilde{\mathbf{y}}\|_\infty\frac{d}{d\lambda}(C_{ode}) + \\ &2C_{ode}h^q\sum_{k=0}^{N}\frac{d}{d\lambda}(y_k(\lambda)) + 2C_{ode}h^q(N+1)\frac{d}{d\lambda}(C_{ode}h^q) \leq \\ &\nabla J_c^{DIR} + 2(T-t_0)h^{q-1}\|\mathbf{y}(\lambda) - \widetilde{\mathbf{y}}\|_\infty\frac{d}{d\lambda}(C_{ode}) + 2C_{ode}h^q(N+1)\|\frac{d}{d\lambda}\mathbf{y}(\lambda)\|_\infty + \\ &2C_{ode}h^{2q}(N+1)\frac{d}{d\lambda}(C_{ode}) = \\ &\nabla J_c^{DIR} + 2\left[(T-t_0)h^{q-1}\|\mathbf{y}(\lambda) - \widetilde{\mathbf{y}}\|_\infty + C_{ode}(T-t_0)h^{2q-1} + C_{ode}h^{2q}\right]\frac{d}{d\lambda}(C_{ode}) + \\ &2C_{ode}(T-t_0)h^{q-1}\|\frac{d}{d\lambda}\mathbf{y}(\lambda)\|_\infty. \end{aligned} \tag{3.15}$$

Therefore:

$$
\begin{aligned}
&err_{\nabla J^{DIR}}(\lambda,h) = |\nabla J(\lambda) - \nabla J^{DIR}(\lambda,h)| \\
&|\nabla J(\lambda) - \nabla J_c^{DIR} + \nabla J_c^{DIR} - \nabla J^{DIR}(\lambda,h)| \leq \\
&\underbrace{|\nabla J(\lambda) - \nabla J_c^{DIR}|}_{E_{n2}\text{: error ``2-norm''}} + \underbrace{|\nabla J_c^{DIR} - \nabla J^{DIR}(\lambda,h)|}_{E_{ode}\text{: error numerical method}} = \\
&\frac{T-t_0}{2}K(\lambda) + 2[(T-t_0)h^{q-1}\|\mathbf{y}(\lambda) - \widetilde{\mathbf{y}}\|_\infty + \\
&C_{ode}(T-t_0)h^{2q-1} + C_{ode}h^{2q}]\frac{d}{d\lambda}(C_{ode}) + \\
&2C_{ode}(T-t_0)h^{q-1}\|\frac{d}{d\lambda}\mathbf{y}(\lambda)\|_\infty.
\end{aligned}
\tag{3.16}
$$

□

We have proved that the gradient, as well as the cost function, does not converge to its continuous counterpart. Moreover the error E_{ode} does not vanish if we use a first order method for ODE, like Explicit Euler. The numerical behavior of the error for different λ is quite similar to the case of the cost function. Then we do not graphically represent these behaviors and we move on to study the gradient in the Indirect approach that presents different and interesting properties.

3.2.2 Gradient in Indirect Approach

As we have seen, the gradient used during the optimization in the Indirect approach is $\mathcal{H}'(\lambda,h)$ in (1.32), the discretization of (1.30) that we recall below:

$$
\mathcal{H}'(\lambda) = \int_0^{10} \psi(t,\lambda)y(t,\lambda)dt. \tag{3.17}
$$

We can compute the analitical forms of $\psi(\lambda,t)$, $\psi(\lambda,t)y(\lambda,t)$ and $\mathcal{H}'(\lambda)$:

$$
\begin{aligned}
\psi(\lambda,t) = &\frac{e^{-\lambda t}}{2\lambda^2(\lambda-1)}\Big[2\lambda^2(e^{(\lambda-1)t} - e^{10(\lambda-1)}) + \\
&- 2(\lambda-1)(2\lambda+1)(e^{2\lambda t} - e^{20\lambda}) + 2(\lambda-1)(\lambda+1)(e^{\lambda t} - e^{10\lambda})\Big],
\end{aligned}
\tag{3.18}
$$

$$
\psi(\lambda,t)y(\lambda,t) = \psi(\lambda,t)\frac{\mathrm{e}^{\lambda\, t}\ (2\,\lambda+1)-1}{2\lambda}, \tag{3.19}
$$

$$\mathcal{H}'(\lambda) = [\mathrm{e}^{-10\,\lambda}\,\mathrm{e}^{-10}\,(4\,\lambda^3\,\mathrm{e}^{10\,\lambda} - 4\,\lambda^4\,\mathrm{e}^{10\,\lambda} + 9\,\mathrm{e}^{10\,\lambda}\,\mathrm{e}^{10} - 12\,\mathrm{e}^{20\,\lambda}\,\mathrm{e}^{10} + 3\,\mathrm{e}^{30\,\lambda}\,\mathrm{e}^{10} +$$
$$- 4\,\lambda^2\,\mathrm{e}^{10}\,\mathrm{e}^{10\,\lambda-10} + 4\,\lambda^3\,\mathrm{e}^{10}\,\mathrm{e}^{10\,\lambda-10} + 38\,\lambda\,\mathrm{e}^{10\,\lambda}\,\mathrm{e}^{10} + 40\,\lambda\,\mathrm{e}^{20\,\lambda}\,\mathrm{e}^{10} - 18\,\lambda\,\mathrm{e}^{30\,\lambda}\,\mathrm{e}^{10} +$$
$$- 59\,\lambda^2\,\mathrm{e}^{10\,\lambda}\,\mathrm{e}^{10} - 32\,\lambda^3\,\mathrm{e}^{10\,\lambda}\,\mathrm{e}^{10} + 56\,\lambda^4\,\mathrm{e}^{10\,\lambda}\,\mathrm{e}^{10} + 68\,\lambda^2\,\mathrm{e}^{20\,\lambda}\,\mathrm{e}^{10} - 128\,\lambda^3\,\mathrm{e}^{20\,\lambda}\,\mathrm{e}^{10} +$$
$$- 48\,\lambda^4\,\mathrm{e}^{20\,\lambda}\,\mathrm{e}^{10} + 80\,\lambda^5\,\mathrm{e}^{20\,\lambda}\,\mathrm{e}^{10} - 49\,\lambda^2\,\mathrm{e}^{30\,\lambda}\,\mathrm{e}^{10} + 60\,\lambda^3\,\mathrm{e}^{30\,\lambda}\,\mathrm{e}^{10} + 84\,a^4\,\mathrm{e}^{30\,\lambda}\,\mathrm{e}^{10} +$$
$$- 80\,\lambda^5\,\mathrm{e}^{30\,\lambda}\,\mathrm{e}^{10} + 4\,\lambda^2\,\mathrm{e}^{10\,\lambda}\,\mathrm{e}^{10}\,\mathrm{e}^{10\,\lambda-10} - 48\,\lambda^3\,\mathrm{e}^{10\,\lambda}\,\mathrm{e}^{10}\,\mathrm{e}^{10\,\lambda-10} - 48\,\lambda^4\,\mathrm{e}^{10\,\lambda}\,\mathrm{e}^{10}\,\mathrm{e}^{10\,\lambda-10} +$$
$$+ 80\,\lambda^5\,\mathrm{e}^{10\,\lambda}\,\mathrm{e}^{10}\,\mathrm{e}^{10\,\lambda-10})]\frac{1}{8\,\lambda^4\,(\lambda-1)^2}. \tag{3.20}$$

By a computer algebra program, we can verify that (1.21) and (3.20) indeed coincide:

$$\mathcal{H}'(\lambda) = \nabla J(\lambda), \qquad \forall\lambda. \tag{3.21}$$

We can prove that the discretization of $\mathcal{H}'(\lambda)$ coincides with the derivative of the discrete cost $J^{IND}(\lambda, h)$ with respect to λ, that is

$$\mathcal{H}'(\lambda, h) = \nabla J^{IND}(\lambda, h), \tag{3.22}$$

where

$$\nabla J^{IND}(\lambda, h) = \frac{\partial}{\partial\lambda} J^{IND}(\lambda, h) = \sum_{k=0}^{N} w_k \frac{\partial}{\partial\lambda}(u_k(\lambda) - \tilde{y}_k)^2. \tag{3.23}$$

In fact, let us define the discretization of (1.20) $\nabla_h J(\lambda)$ as follows:

$$\nabla_h J(\lambda) = 2\sum_{k=0}^{N} w_k \frac{\partial u_k(\lambda)}{\partial\lambda}(u_k(\lambda) - \tilde{y}_k). \tag{3.24}$$

It is easy to see that, if the composite quadrature formula used in (3.23) and (3.24) coincide, it holds:

$$\nabla J^{IND}(\lambda, h) = \nabla_h J(\lambda)$$

therefore:

$$\mathcal{H}'(\lambda) = \nabla J(\lambda) \Rightarrow \mathcal{H}'(\lambda, h) = \nabla_h J(\lambda) \Rightarrow \mathcal{H}'(\lambda, h) = \nabla J(\lambda, h).$$

Error analysis We compute the error between $\mathcal{H}'(\lambda)$ and its numerical approximation $\mathcal{H}'(\lambda, h)$ for $\lambda \in [-2, -0.1]$ and calculate the covergence order p. Let us define:

$$err_{\mathcal{H}'}(\lambda, h) = |\mathcal{H}'(\lambda) - \mathcal{H}'(\lambda, h)|. \tag{3.25}$$

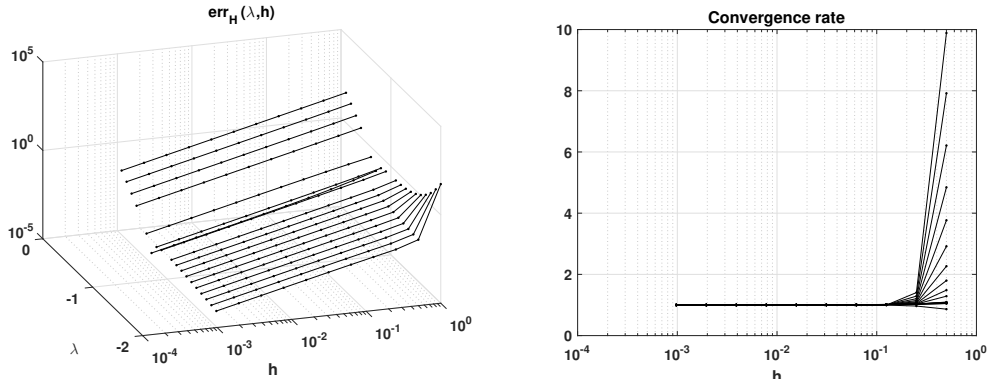

Figure 3.8: Indirect Approach: The method used to solve the ODE is Explicit Euler, with $\lambda \in [-2, -0.1]$, $\lambda \neq -1, -0.5$. The plot on the left represent $err_{\mathcal{H}'}(\lambda, h)$ in (3.25). Plot on the right shows the convergence order p.

In Figure 3.8 we can see a numerical simulation, where we have used Explicit Euler and its *reflected* scheme to solve the state equation and the adjoint equation respectively, and the composite trapezoidal quadrature rule to approximate the integral. In spite of the quadrature formula has convergence order $r = 2$, the convergence order of the error is 1.

Let us analyze the theoretical difference between the gradients.

Proposition 3.2.2. *Let $\mathcal{H}'(\lambda, h)$ be the approximation of the gradient $\mathcal{H}'(\lambda)$ and $h = \frac{T-t_0}{N}$. Let q be the convergence order for the method used for the state equation, q_a the convergence order of the method used to solve the adjoint equation and r the convergence order of a composite quadrature formula to approximate the integral. It results:*

$$\begin{aligned} &err_{\mathcal{H}'}(\lambda, h) = |\mathcal{H}'(\lambda) - \mathcal{H}'(\lambda, h)| \leq \\ &\leq Z(\lambda)h^r + C(T)h^q + C_a(T)h^{q_a} + C(T)C_a(T)h^{q_a+q-1}. \end{aligned} \tag{3.26}$$

Proof. Let $\mathcal{H}'_c(\lambda, h)$ be the approximation of the gradient $\mathcal{H}'(\lambda)$ where $\psi(\lambda) = [\psi(t_0), ..., \psi(t_N)]^T = [\psi_0, ..., \psi_N]^T$ and $\mathbf{y}(\lambda) = [y(t_0), ..., y(t_N)]^T = [y_0, ..., y_N]^T$ are analytically computed and evaluated on the mesh grid. Let $\mathbf{u}(\lambda) = [u_0, u_1, ..., u_N]^T$ be the numerical approximation of $\mathbf{y}$ and $\mathbf{v}(\lambda) = [v_0, v_1, ..., v_N]^T$ be the numerical

approximation of $\mathbf{p}$. It follows that:

$$\begin{aligned}
&\mathcal{H}'(\lambda,h)=\sum_{k=0}^{N}w_k v_k(\lambda)u_k(\lambda)\approx\sum_{k=0}^{N}w_k(\psi_k(\lambda)+C_a h^{q_a})(y_k(\lambda)+Ch^q)=\\
&\sum_{k=0}^{N}w_k\psi_k(\lambda)y_k(\lambda)+\sum_{k=0}^{N}w_k(\psi_k(\lambda)Ch^q+y_k(\lambda)C_a h^{q_a}+C_a h^{q_a}Ch^q)=\\
&\mathcal{H}'_c(\lambda,h)+Ch^q\sum_{k=0}^{N}w_k\psi_k(\lambda)+C_a h^{q_a}\sum_{k=0}^{N}w_k y_k(\lambda)+(T-t_0)CC_a h^{q_a+q-1}\leq\\
&\mathcal{H}'_c(\lambda,h)+(T-t_0)Ch^q\left\|\psi(\lambda)\right\|_\infty+(T-t_0)C_a h^{q_a}\left\|\mathbf{y}(\lambda)\right\|_\infty+\\
&(T-t_0)CC_a h^{q_a+q-1}.
\end{aligned}\tag{3.27}$$

Therefore:

$$\begin{aligned}
&err_{\mathcal{H}'}(\lambda,h)=|\mathcal{H}'(\lambda)-\mathcal{H}'(\lambda,h)|\\
&|\mathcal{H}'(\lambda)-\mathcal{H}'_c(\lambda,h)+\mathcal{H}'_c(\lambda,h)-\mathcal{H}'(\lambda,h)|\leq\\
&\underbrace{|\mathcal{H}'(\lambda)-\mathcal{H}'_c(\lambda,h)|}_{E_{quadr}:\ error\ quadrature}+\underbrace{|\mathcal{H}'_c(\lambda,h)-\mathcal{H}'(\lambda,h)|}_{E_{ode}:\ error\ numerical\ method}=\\
&Z(\lambda)h^r+(T-t_0)Ch^q\left\|\psi(\lambda)\right\|_\infty+(T-t_0)C_a h^{q_a}\left\|\mathbf{y}(\lambda)\right\|_\infty+\\
&(T-t_0)CC_a h^{q_a+q-1}.
\end{aligned}\tag{3.28}$$

□

Let us call

$$E_{quadr}(\lambda,h)=Z(\lambda)h^r$$

and

$$E_{ode}(\lambda,h)=(T-t_0)Ch^q\left\|\psi(\lambda)\right\|_\infty+(T-t_0)C_a h^{q_a}\left\|\mathbf{y}(\lambda)\right\|_\infty+(T-t_0)CC_a h^{q_a+q-1}.$$

where the first term is due by the approximation of the state ODE equation and the second one is due by the approximation of the adjoint ODE equation.

Therefore the covergence order p of $err_{\mathcal{H}'}(\lambda,h)$ is given by:

$$p=\min\{r,q,q_a\}.\tag{3.29}$$

If we use different methods to solve the differential equations, we observe how the covergence order of the error changes. For example, by using Crank-Nicholson ($q=2$) and Runge-Kutta 4 ($q=4$) to solve the state ODE equation end their

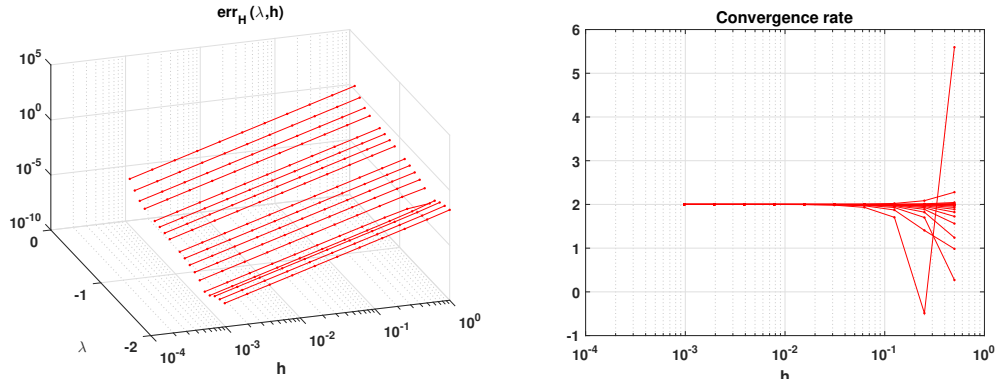

Figure 3.9: Indirect Approach: The method used to solve the ODE is Crank-Nicolson, with $\lambda \in [-2, -0.1]$, $\lambda \neq -1, -0.5$. The plot on the left represent $err_{\mathcal{H}'}(\lambda, h)$ in (3.25). Plot on the right shows the convergence order p (3.29)

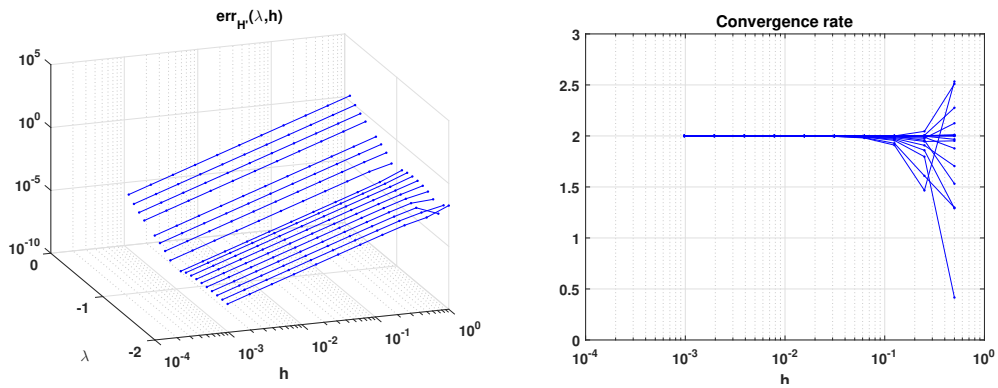

Figure 3.10: Indirect Approach: The method used to solve the ODE is Runge-Kutta 4, with $\lambda \in [-2, -0.1]$, $\lambda \neq -1, -0.5$. The plot on the left represent $err_{\mathcal{H}'}(\lambda, h)$ in (3.25). The plot on the right shows the convergence order p (3.29)

reflected for the adjoint equation, we obtain the following Figures 3.9 and 3.10, where it is represented $err_{\mathcal{H}'}(\lambda, h)$ and the covergence order p. Let us observe that in these cases $q_a = q$, because the reflection preserves the order of the method (as detailed in the 2.2).

Remark 3.2.1. *As suggested by Sanz-Serna in [72], in optimal control problem it would be necessary to use a Partitioned Runge-Kutta method because of the structure of the problem. But a PRK for optimal control is obtained by the use of a RK method for the state equation and its* transposed *to solve the adjoint, as explained in Section 2.2. The problem is the follow: generally, the transposition does not preserve the order of the method [74].*

3.3 Summary

In this Chapter we examined the cost functions derived from the two approaches and their gradients. As we have seen, in both approaches the error between the discrete costs (and gradients) and their continuous counterpart involve two different contributions: one given by the method used in defining the cost function and a second one due by the method used for the ODE resolution.

In the Direct approach we have by construction a discrete cost, defined by a norm and therefore structurally different from the integral in the continuous problem; then, mainly a suitable method for resolution of the ODE should be provided. Furthermore, the error Err_{ode} does not vanish, for $h \to 0$, if the convergence order of the chosen method is 1.

The Indirect approach expects the minimization of a cost functional defined by an integral, that is approximated by an suitable quadrature formula as well as its gradient. Furthermore, it is necessary to provide suitable ODE methods for the state and the adjoint equations. From the analysis we deduced that it is useless to consider an ODEs method of high order if the quadrature formula has a lower order: indeed the approximation of lower order will predominate in the approximation.

In the following, we will analyze more complex problems, in terms of target data and strongly non-linear differential models (which represent the constraints); therefore we decide to consider the Direct approach and study the issues derived from this discretization. In particular, we will compare the classical cost function with another suitably defined, resulting from the analysis of experimental data of interest. The Indirect approach, as we have seen, raises further different issues that would add to the complexity of the problems, and we expect that can be object of future works.

Chapter 4

ODE-PIP with oscillatory dynamics

In this Chapter we will show that the cost function (1.3) for the ODE-PIP on oscillating data inherits the oscillating data behaviour and has many different "low" minima, then we show that adding a classical regularization term as in (1.13) actually does not improve this structure of the cost. Since in this situation any optimization algorithm is liable to fail in the approximation of a good solution, we propose a new approach which takes into considerations the oscillating nature of the data. To avoid the bad features of the classical cost function, we propose to rewrite and solve the original ODE-PIP in the Fourier space, by defining a new cost function based on the discrete Fourier transform and comparing frequencies of data and simulations. We will show that the multiple minima of the two norm are correlated and indeed belong to a sub-manifold $\mathcal{S}$ of codimension-one in the m-dimensional parameter space $\Omega \subset \mathbb{R}^m$. These results have been published in [20].

PIP for periodic data were studied, for example, by S. Röblitz et al. [69] and S.P. Ellner et al. [26]. In [69] a model of 33 differential equations is analyzed with more then one hundred parameters and 14 of these parameters are identified by a Gauss-Newton method. In [26, 51] the so-called *gradient matching* is used, rather than the most used trajectory matching: in a preliminary smoothing step, the time series data are interpolated; then, in a second step, the parameters of the ODEs are optimized, so as to minimize some metric measuring the difference between the slopes of the tangents to the interpolants, and the time derivatives from the ODEs.

In this way, the ODEs never have to be solved explicitly. Nevertheless in this Chapter the questions related to the presence of multiple minima are addressed only in terms of suitable numerical strategies such that the optimization method can avoid local minima.

4.1 Test Identification problem for oscillating data (TIP-OD)

To present the problems derived from the study of oscillating data, in this Section we propose a simple form of PIP, that we call Test Identification Problem for oscillating data (TIP-OD), such that: (i) the constraint is a linear ODE system of two equations with known exact solution; (ii) only one component $u(t)$ of the solution system has to be compared with the target function; (iii) the target is a given oscillating function, such that the cost function has one known global minimum; (iv) only one parameter must be identified. Hence, let us consider a general linear PIP as follows:

$$\min_{p_1 \in \Omega} \; J(\mathbf{y}(t), \widetilde{\mathbf{y}}(t), p_1) \tag{4.1}$$

$$\mathbf{y}'(t) = A(p_1)\mathbf{y}, \quad \mathbf{y}(t_0) = \mathbf{y}_0, \tag{4.2}$$

where $\mathbf{y} = (u, v)^T$, $p_1 = \beta$, $A(\beta) = \begin{bmatrix} 0 & -\beta \\ \beta & 0 \end{bmatrix}$. In particular, we consider the TIP:

$$\min_{\beta \in [\beta_0, \beta_f]} J(u(t), \widetilde{u}(t), \beta) \tag{4.3}$$

$$\begin{cases} u'(t) = -\beta v \\ v'(t) = \beta u \\ u(t_0) = u_0, \; v(t_0) = v_0, \quad t \in [t_0, T] \end{cases} \tag{4.4}$$

such that the exact solution of (4.4) is given by $u(t;\beta) = u_0 \cos(\beta t), v(t;\beta) = v_0 \sin(\beta t)$. Here $\beta = 2\pi f_e$ is the *parameter* to be identified (f_e is the *frequency*), and $\widetilde{u}(t) = u_0 \cos(\widetilde{\beta} t)$ is the assigned *target function* with fixed $\widetilde{\beta} = 2\pi \widetilde{f}_e$. By following the direct approach [87], given a timestep $h = \frac{T - t_0}{N}$ and the time grid $t_i = t_0 + h\, i$ for $i = 0, ..., N$, if $\mathbf{U}(\beta) = [u(t_i)]$ and $\widetilde{U} = [\widetilde{u}(t_i)]$, the cost function is

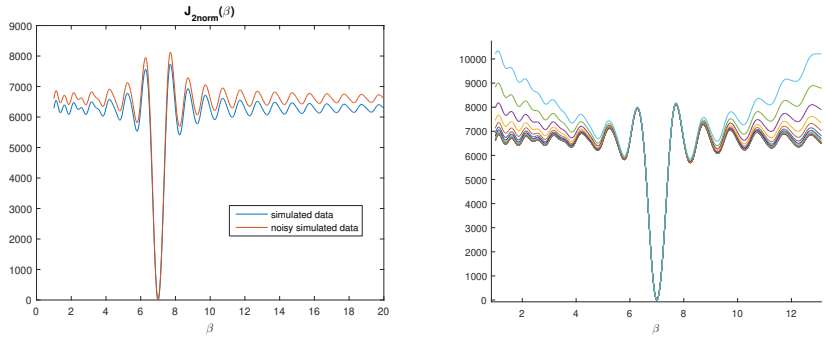

Figure 4.1: TIP: Least squares cost functions (left) and regularization (4.6) for $\alpha \in [0.1, 10]$ (right).

given by:

$$J_{2norm}(\beta) = \|\mathbf{U}(\beta) - \widetilde{\mathbf{U}}\|_2^2 = \sum_{i=0}^{N} (u_0 \cos(\beta t_i) - u_0 \cos(\widetilde{\beta} t_i))^2 = \\ = \sum_{i=0}^{N} 4u_0^2 \sin^2 \frac{(\beta + \widetilde{\beta})t_i}{2} \sin^2 \frac{(\beta - \widetilde{\beta})t_i}{2}, \tag{4.5}$$

that evidently inherits an oscillating behaviour and $J_{2norm}(\beta) = 0 \iff \beta = \widetilde{\beta}$ or $\beta = -\widetilde{\beta}$. For example, let us fix: $t_0 = 0$, $T = 2\pi$, $u_0 = 1$, $v_0 = 0$, $\widetilde{\beta} = 7$ (then $\widetilde{f}_e \approx 1.114$) and $h = 0.001$, As usual, we can add some noise to the simulated data such that $\widetilde{\mathbf{U}}^r = \widetilde{\mathbf{U}} + 10^{-4} rand$, where rand is a random perturbation with uniform distribution. For $\beta \in [1, 20]$ the cost functions (4.5) without and with noised data are shown in Figure 4.1. Clearly an optimization algorithm will fail in the search of the global minimum, because these functions have many different local minima at (almost) the same level that cannot be easily avoided since, as well known, the convergence depends on the starting point. It is easy to see that the bad behaviour does not depend on the addition of noise to the data.

Let us try now to solve the problem of unconstrained optimization, looking for the minimum of the cost in function in (4.5) by using the MATLAB [56] command $fminunc$ which find the minumum of unconstrained function given in input. The used algorithm is the BFGS Quasi-Newton method with a cubic line search procedure [11, 29] and the tolerance tol is defined by the default value $1e-06$ for all the stopping criteria. The results are summarize in Table 4.1. As we expect the algorithm fails in the minimization when the starting point β_0 is not enough close to the minumum: the search stops in a local minimum.

Analogous behavior results by using the MATLAB [56] function $lsqnonlin$,

β_0	iterations $\bar{k}$	$\beta_{\bar{k}}$	convergence	final cost
2	2	1.85	gradient	78.03
5	5	4.80	gradient	76.02
6.9	2	6.999	cost	3.7e-06
7.1	1	6.999	cost	1.37e-8
9	5	9.26	gradient	76.01
15	5	15.28	gradient	78.20

Table 4.1: TIP-OD: Results of the optimization with *fminunc* and different starting values β_0

β_0	iterations $\bar{k}$	$\beta_{\bar{k}}$	convergence	final cost
2	22	-3.82	increment	77.67
5	19	4.80	increment	76.82
6.9	3	7	gradient	2.13e-13
7.1	3	7	gradient	1.34e-11
9	19	9.26	cost increment	76.01
15	24	15.28	increment	78.20

Table 4.2: TIP-OD: Results of the optimization with *lsqnonlin* and different starting values β_0

preferred choice in the minimization of 2-norm. By default, *lsqnonlin* uses the trust-region-reflective algorithm described in [17, 16], and a tolerance $tol = 1e-6$. The results are shown in the following Table 4.2.

If a classical regularization term is added to (4.5), as for example, the one suggested in [42, 38], we have:

$$J^R_{2norm}(\beta) = \|\mathbf{U}(\beta) - \widetilde{\mathbf{U}^r}\|_2^2 + \alpha\|\beta\|_2^2, \tag{4.6}$$

the plot of which (for noised data) in a neighborhood of $\widetilde{\beta}$ is shown in Figure 4.1 (right) for $\alpha \in [0.1, 10]$. It is clear also that the regularized cost function is not convex and the occurrence of many local minima is not avoided. For this reason and in order to take into account the features of the data, we introduce a Fourier regularization approach based on the Fourier transformation of data and

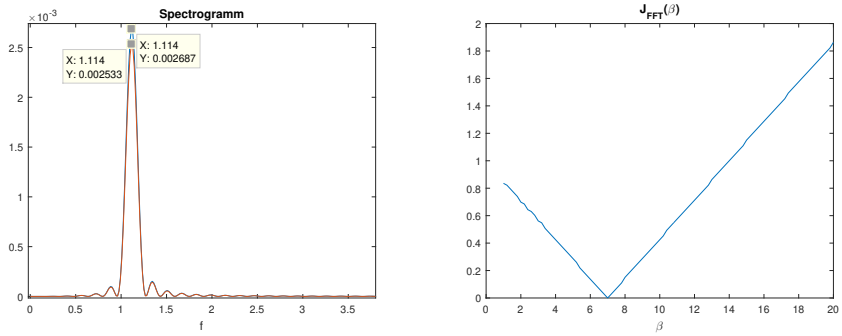

Figure 4.2: TIP: FFT power spectra $\widetilde{P}=|\widetilde{M}|^2$ for noised and unnoised data (left); cost function (4.7) for $\beta \in [0, 20]$ (right)

simulations as follows. Let us consider the new cost function

$$J_{FFT}(\beta) = \frac{|f(\beta) - \widetilde{f}|}{\widetilde{f}}, \tag{4.7}$$

that is the relative error between the dominant frequencies $\widetilde{f}$ and $f(\beta)$ of the (noised) data $\widetilde{\mathbf{U}^r}$ and of the numerical solutions (here known exactly) $\mathbf{U}(\beta)$, respectively. $\widetilde{f}$ and $f(\beta)$ are obtained by the Discrete Fourier Transform (DFT) of data and simulations as follows. Let

$$\widetilde{M} = \text{fft}(\widetilde{\mathbf{U}}) \quad M_\beta = \text{fft}(\mathbf{U}(\beta))$$

be the Fast Fourier Transforms (FFT) of $\widetilde{\mathbf{U}^r}$ and $\mathbf{U}(\beta)$, calculated by the MATLAB [56] function fft (see appendix A for more details). The first dominant frequency can be, as usual, extracted as the abscissa of the maximum of the corresponding power spectra $\widetilde{P} = |\widetilde{M}|^2$ and $P_\beta = |M_\beta|^2$. The spectrograms of the noised and unnoised target data are shown in Figure 4.2, where in both cases $\widetilde{f} \approx 1.114$. Note that $\widetilde{f}$ and $f(\beta)$ are the numerical approximations of the exact frequencies $\widetilde{f}_e = \frac{\widetilde{\beta}}{2\pi}$ and $f_e(\beta) = \frac{\beta}{2\pi}$. Then for the calculated Fourier cost function (4.7), shown in Figure 4.2 with the (obvious) global minimum in $\beta^* = \widetilde{\beta} = 7$, we have

$$J_{FFT}(\beta) \approx \frac{|f_e(\beta) - \widetilde{f}_e|}{\widetilde{f}_e} = \frac{|\frac{\beta}{2\pi} - \frac{\widetilde{\beta}}{2\pi}|}{\frac{\widetilde{\beta}}{2\pi}} = \frac{|\beta - \widetilde{\beta}|}{\widetilde{\beta}}. \tag{4.8}$$

It is worth noting that (4.7) is not sufficiently regular to allow the use of methods such as, for example, Newton-like ones, but other derivative-free algorithms could be considered to approximate the global minimum. Nevertheless, our aim here is not to minimize numerically (4.7), but rather to present the TIP as a toy

parameter estimation problem showing that, working in the Fourier space (by using the FFT), in the case of oscillating data we can devise a cost function that avoids the drawbacks of usual least squares. Note that, in the construction of the TIP, since we use as initial condition for the ODE system exactly u_0 as in the simulated target, we do not account for amplitude and phase differences between data and simulations. The Fourier regularization for this simplified TIP is thus able to track only the (main) frequency present in the data. We will show in the next Sections that for PIP with a nonlinear ODE system constraint and more than one parameter to be identified, the Fourier regularization approach will provide a "tool" to minimize also the phase error between (normalized) data and simulations.

4.2 Fourier regularization: simulated data

In this Section we apply our approach to solve a PIP where the constraint is given by the ODE version of the well-known Schnackenberg model introduced in [75] to describe an autocatalytic chemical reaction with possible oscillatory behaviors. This PDE model is a prototype of nonlinear reaction-diffusion system with spatial pattern formation due to Turing or diffusion-driven instability. As recent papers [50, 68] show, this model receives great attention for the two following main reasons: (i) it has a very simple structure; (ii) its patterns are qualitatively similar to classical ones found in biological experiments. Here, we are interested in the identification of the reaction kinetic parameters of the model without the diffusion terms.

Let us define the Schnackenberg-PIP as follows:

$$\min_{(\alpha,\beta)\in\Omega} J(v(t),\widetilde{v}(t),\alpha,\beta) \tag{4.9}$$

$$\begin{cases} u'(t) = a - u + u^2 v, \\ v'(t) = b - u^2 v, \qquad\qquad t \in]t_0, T] \end{cases} \tag{4.10}$$

where $v(t)$ is a solution of the ODE system in (4.10) and $\widetilde{v}(t)$ is a given oscillating target. Let us observe that we decided to include only $v(t)$ and not $u(t)$ in the cost function; this to have an analogous problem to the experimental case considered in the next Section, where we will have the experimental data only for one of the

two variables of the differential model.

The equilibrium point for (4.10) is given by $P_e = (u_e, v_e) = (a+b, \frac{b}{(a+b)^2})$. Let $\beta = a + b$ and $\alpha = a - b$, then the parameter assumption $a > 0$ and $b > 0$ is equivalent to $\beta > 0$ and $-\beta < \alpha < \beta$. To have oscillating solutions, we consider (α, β) in a neighbourhood of the Hopf line, where the eigenvalues $\lambda_{1,2}$ of $J(u_e, v_e)$, the Jacobian of the kinetics in (4.10) evaluated at the equilibrium point, are complex conjugate with very small positive real part, such that a limit cycle $(u(t), v(t))$ around P_e is expected as asymptotic solution. As detailed in [50], this yields the following bound for the parameter space Ω: $\alpha \neq 0$ small and $|\alpha| \ll |\beta|$. Hence, we will identify indirectly the parameters a and b by estimating $(\alpha, \beta) \in \Omega = [\alpha_0, \alpha_f] \times [\beta_0, \beta_f] = [0.5, 1] \times [0.5, 1] \in \mathbb{R}^2$.

As before, we apply the direct approach to the ODE-PIP (4.9)-(4.10). We fix $t_0 = 0$ and the final time of integration to $T = 150$. We consider the timestep $h = 0.005$ and the explicit RK4 to approximate (4.10). Let be $\mathbf{V}(\alpha, \beta) = [V_i]$, $V_i(\alpha, \beta) \approx v(t_i, \alpha, \beta)$ the numerical approximation of $v(t)$ on the uniform grid $t_i = i\,h, i = 0, \ldots, N$. We generate simulated target data $\widetilde{\mathbf{V}}$ on the same grid by fixing $\widetilde{\alpha} = 0.88$ and $\widetilde{\beta} = 0.9$. (We avoid to add noise for the reasons highlighted in the previous Section on TIP.) In our analysis of the direct cost we decide to consider $\widetilde{\mathbf{V}}$ and $\mathbf{V}(\alpha, \beta)$ only for $t \in [\bar{t}, T]$ to avoid the comparison in the transient dynamics and to focus only on the "asymptotic" standing oscillations. Moreover, as anticipated above, we normalize both data and simulations, so that we do not need to compare the amplitude of the oscillations. From now on, for abuse of notations, let $\widetilde{\mathbf{V}}, \mathbf{V}(\alpha, \beta) \in \mathbb{R}^N$ be the normalized vectors extracted from the previous ones for $t \in [\bar{t}, T]$, $\bar{t} = 100$. Hence, we want to study the behaviour of the classical two-norm (without classical regularization, for the reasons highlighted in the previous Section on TIP):

$$J_{2norm}(\alpha, \beta) = \|\mathbf{V}(\alpha, \beta) - \widetilde{\mathbf{V}}\|_2^2. \tag{4.11}$$

To this aim, we discretise the parameter space Ω by using $h_\alpha = h_\beta = 0.0025$ and we evaluate (4.11) for each sample pair $(\alpha_i, \beta_j), i, j = 1, \ldots, N_h$. As we can see in Figure 4.4 (left) the obtained approximation of (4.11) has different very low local minima and an optimization algorithm could fail in the search for the global (simulated) one. For this reason, we apply our Fourier regularization and

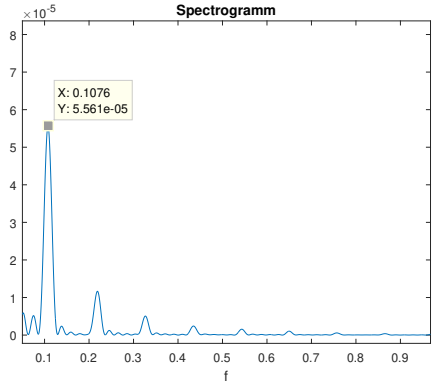

Figure 4.3: Schnackeberg-PIP: spectrogram of the simulated target data.

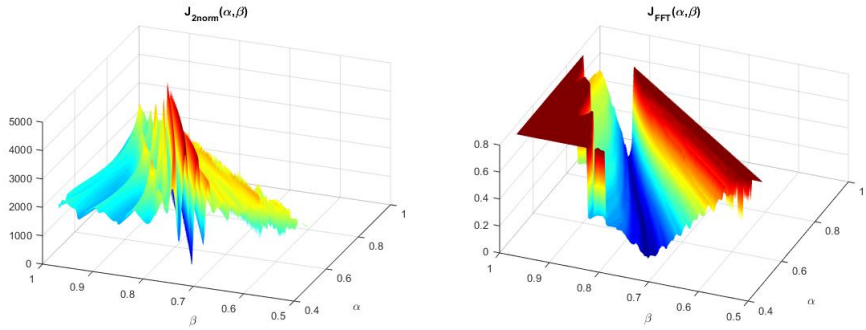

Figure 4.4: Schnackeberg-PIP: least squares (4.11) (left) and Fourier (4.12) (right) costs

we define the new Fourier cost function for Schnackeberg-PIP as:

$$J_{FFT}(\alpha,\beta) = \frac{|f_1(\alpha,\beta) - \widetilde{f}_1|}{\widetilde{f}_1}, \tag{4.12}$$

where $\widetilde{f}_1$ is the first dominant frequency of $\widetilde{\mathbf{V}}$ extracted from the *power spectrum* $\widetilde{P} = |\widetilde{M}|^2$, with $\widetilde{M} = \text{fft}(\widetilde{\mathbf{V}})$ the FFT of $\widetilde{\mathbf{V}}$. The corresponding spectrogram is shown in Figure 4.3. Similarly, $f_1(\alpha,\beta)$ is the first dominant frequency of the numerical solutions $\mathbf{V}(\alpha_i,\beta_j)$, for all $(\alpha_i,\beta_j) \in \Omega_h$, computed in the same way from the spectrogram $P_{\alpha,\beta} = |M_{\alpha,\beta}|^2$. The Fourier cost (4.12) is shown in Figure 4.4 (right).

It is evident that the new cost presents a long valley of "correlated" minima for the values (α,β) for which $J_{FFT}(\alpha,\beta) = 0$, that is where $\mathbf{V}(\alpha,\beta)$ and $\widetilde{\mathbf{V}}$ have the same frequency. Computationally, we can define the following discrete set:

$$\Omega_h^{FFT} = \{(\alpha_i,\beta_j) \in \Omega_h \quad | \quad J_{FFT}(\alpha_i,\beta_j) = 0\} \tag{4.13}$$

that is represented in Figure 4.5 (left, 'o' symbols): a single iso-frequency curve

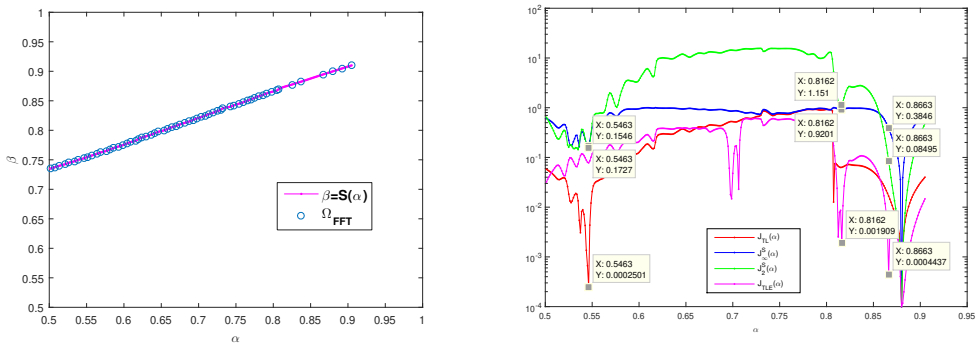

Figure 4.5: Schnackeberg-PIP. Left: the set Ω_h^{FFT} in (4.13) and the interpolating iso-frequency Hermite curve $\beta = \mathcal{S}(\alpha)$. Right: the cost functions (4.14), (4.15), (4.16) and (4.17) restricted to $\mathcal{S}(\alpha)$.

in the plane α-β can be identified. (Note that, since the frequencies correspond to the abscissae in a spectrogram, J_{FFT} is exactly zero because we compare the spectrograms $P_{\alpha,\beta}$ and $\widetilde{P}$ for numerical solutions and target on the same grid points.) By numerical interpolation, we can find a continuous form of this curve, say $\beta = \mathcal{S}(\alpha),\ \alpha \in [\alpha_0, \alpha_f]$, for example by approximating the values in Ω_h^{FFT} by the piecewise cubic Hermite interpolating polynomial. To this aim, here we apply the default MATLAB [56] command *pchip*: the result $\mathcal{S}(\alpha)$ is shown in the same Figure.

On the parametric curve $(\alpha, \mathcal{S}(\alpha))$, that is a sub-manifold of co-dimension one in the parameter space $\Omega \subset \mathbb{R}^m$, $m = 2$, we can evaluate different cost functions to complete the approximation of our two-parameters ODE-PIP. We consider "global measures" like the original least-squares cost and the infinity norm (i.e. the maximum error) projected on $\mathcal{S}$ given by:

$$J_2^{\mathcal{S}}(\alpha) = J_{2norm}(\alpha,\beta)_{|\mathcal{S}} = \|\mathbf{V}(\alpha,\beta) - \widetilde{\mathbf{V}}\|_2^2, \quad \alpha \in [\alpha_0, \alpha_f], \tag{4.14}$$

$$J_\infty^{\mathcal{S}}(\alpha) = J_\infty(\alpha,\beta)_{|\mathcal{S}} = \|\mathbf{V}(\alpha,\beta) - \widetilde{\mathbf{V}}\|_\infty, \quad \alpha \in [\alpha_0, \alpha_f]. \tag{4.15}$$

To look for $\mathbf{p} = (p_1, p_2) = (\alpha, \beta) \in \mathcal{S}$ that yields the minimum phase errors, we compute the so-called time-lags between the data and the simulations at the first and the last times t_1 and t_N of the grid, that is:

$$J_{TL}(\alpha) = J_{TimeLag}(\alpha,\beta)_{|\mathcal{S}} = |V_1(\alpha,\beta) - \widetilde{V}_1|, \qquad \alpha \in [\alpha_0, \alpha_f], \tag{4.16}$$

$$J_{TLE}(\alpha) = J_{TimeLagEnd}(\alpha,\beta)_{|\mathcal{S}} = |V_N(\alpha,\beta) - \widetilde{V}_N|, \qquad \alpha \in [\alpha_0, \alpha_f]. \tag{4.17}$$

Figure 4.5 (right) shows all these costs (4.14)–(4.17) for $\alpha \in [0.5, 0.95]$ and $t_1 = 100$, $t_N = 150$ for time-lag errors. The exact simulated minimum $\widetilde{\alpha} = 0.88$, $\widetilde{\beta} = \mathcal{S}(\widetilde{\alpha}) = 0.9$ is shared by all costs such that $J_*(\widetilde{\alpha}, \widetilde{\beta}) \approx$ 1e-15: it is evidently a global one for the two- and infinity norms. Instead, the time-lag errors have further few low minima.

The initial time-lag $J_{TL}(\alpha)$ presents another very low minimum in $(\alpha_1, \beta_1) = (\alpha_1, \mathcal{S}(\alpha_1)) = (0.5463, 0.7525)$, where $J_{TL}(\alpha_1) =$ 2.5e-4 is very small, but the global measures $J_2^{\mathcal{S}}(\alpha_1) = 0.1546$ and $J_\infty^{\mathcal{S}}(\alpha_1) = 0.1727$ are not. The comparison between the target data and the corresponding numerical simulation $\mathbf{V}(\alpha_1, \beta_1)$ is shown in Figure 4.6(a): they start very near and then a dephasing develops.

Also the final time-lag $J_{TLE}(\alpha)$ has some other "low" local minima: by inspection we look for the minima in correspondence of which also $J_2^{\mathcal{S}}$ and $J_\infty^{\mathcal{S}}$ are low. We find

$$(\alpha_2, \beta_2) = (\alpha_2, \mathcal{S}(\alpha_2)) = (0.8663, 0.8945), \quad (\alpha_3, \beta_3) = (\alpha_3, \mathcal{S}(\alpha_3)) = (0.8150, 0.8738).$$

We have $(J_{TLE}(\alpha_2), J_2^{\mathcal{S}}(\alpha_2), J_\infty^{\mathcal{S}}(\alpha_2)) =$ (4.4e-4, 0.085, 0.3846), and $|\alpha_2 - \widetilde{\alpha}|/|\widetilde{\alpha}| = 1.5\%$ and $|\beta_2 - \widetilde{\beta}|/|\widetilde{\beta}| = 0.61\%$, then this parameter set is indeed very close to the simulated one.

On the other hand, $(J_{TLE}(\alpha_3), J_2^{\mathcal{S}}(\alpha_3), J_\infty^{\mathcal{S}}(\alpha_3)) =$ (1.91e-3, 1.151, 0.9201), but $|\alpha_3 - \widetilde{\alpha}|/|\widetilde{\alpha}| = 7.39\%$ and $|\beta_3 - \widetilde{\beta}|/|\widetilde{\beta}| = 2.91\%$, such that a true minimum is identified. The corresponding numerical solutions are shown in Figure 4.6(b) and (c): simulations and data match well in the final time, but still present a dephasing.

In Figure 4.6(d) we report the time behaviour of the absolute errors $Err(\alpha_i, \beta_i) = |\mathbf{V}(\alpha_i, \beta_i) - \widetilde{\mathbf{V}}|, i = 1, 2, 3$ for all the optimal parameter sets presented above. The solution for (α_2, β_2) seems to be the better, but it produces high errors in the points corresponding to the minima of the data. On the other hand, we can extract a best approximation $\mathbf{p}^* = (\alpha^*, \beta^*)$ such that

$$J^*(\mathbf{p}^*) = \min_{i=1,2,3} (\max\{J_{TL}(\alpha_i), J_{TLE}(\alpha_i), J_2^{\mathcal{S}}(\alpha_i), J_\infty^{\mathcal{S}}(\alpha_i)\}) =$$

$$\min_{i=1,2,3} \{J_\infty^{\mathcal{S}}(\alpha_i)\}.$$

In this way we have that $\mathbf{p}^* = (\alpha_1, \beta_1)$ with $J^* = 0.1727$ (see Figure 4.6(a)) can be considered a good PIP-Fourier solution, different from the simulated one.

The above Fourier procedure for the Schnackenberg-PIP could be used in a similar way to solve any ODE-PIP with only two parameters to be identified. In

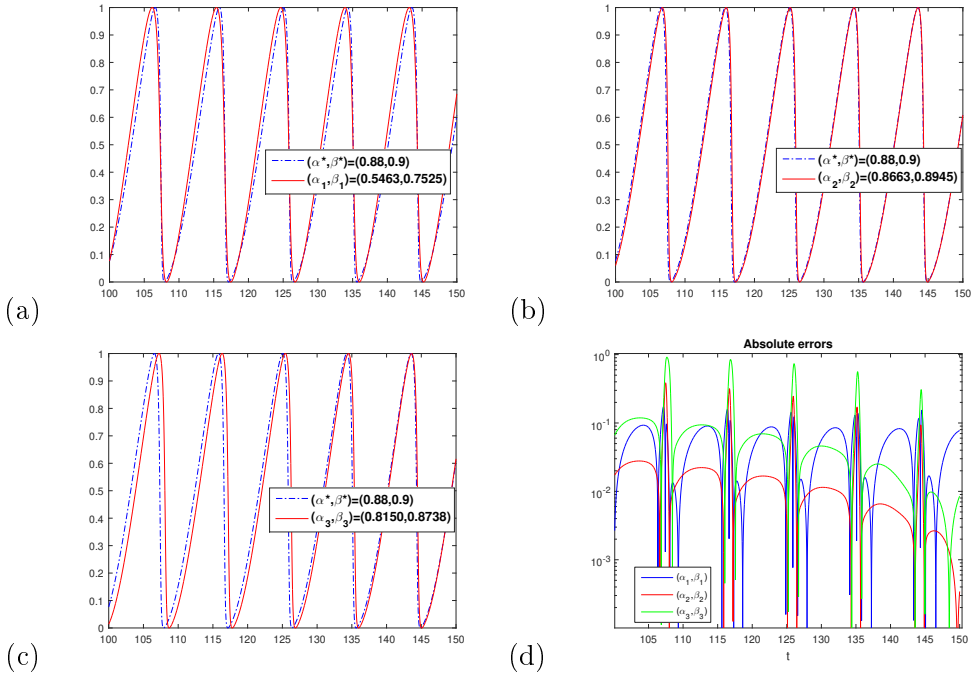

Figure 4.6: Schnackenberg-PIP: Numerical solutions for minima along the curve $\mathcal{S}(\alpha)$ obtained for (α_1, β_1) in (a), (α_2, β_2) in (b) and (α_3, β_3) in (c). (d): Corresponding absolute errors $Err(\alpha_i, \beta_i), i = 1, 2, 3$.

the next Section, we will present the Fourier-PIP approach for a recently proposed model for electrodeposition used to model experimental oscillating data.

Remark 4.2.1. *Looking at the spectrogram of the target data (see e.g. Figure 4.3), in the Fourier cost we could consider also the contribution of a certain number K of dominant frequencies f_k, $k = 2, 3, .., K$ and define a more general cost accounting for more spectral information, that is:*

$$J_{FFT}(\mathbf{p}) = \sum_{k=1}^{K} \frac{|f_k(\mathbf{p}) - \widetilde{f}_k|}{\widetilde{f}_k}. \tag{4.18}$$

Nevertheless, in the present investigation, in order to keep the computational complexity of the problem to a minimum without loss of methodological power and to handle numerical results in a way that allows a clear insight into the proposed Fourier approach, we decide to take into consideration just the first ones (K=1). In this respect it is worth noting that: (i) higher Fourier expansion terms can be incorporated straightforwardly in (4.18)*, without need of further mathematical development and (ii) the first Fourier component bears the most diagnostic information from the point of view of experimental physical chemistry. Of course,*

the key limitation of the present approach is that the information contained in the fine structure of the data oscillations is neglected. Nevertheless, the increment in understanding of the physical problem achieved by identifying the dominant harmonic is dramatic with respect to the state-of-the-art.

4.3 Fourier regularization: experimental data

In this Section we consider a PIP (1.1)-(1.2) where the constraint is the ODE version of the morphochemical reaction-diffusion PDE system modeling an electrodeposition process introduced in [10], for brevity also called DIB model from the names of the authors. The DIB-PIP minimization problem is given by:

$$\min_{\mathbf{p}\in\Omega} J(\theta(t), \widetilde{\theta}(t), \mathbf{p}) \tag{4.19}$$

$$\begin{cases} \eta'(t) = f(\eta, \theta) \\ \theta'(t) = g(\eta, \theta) \\ \eta(0) = \eta_0,\ \theta(0) = \theta_0, \quad t \in [t_0, T] \end{cases} \tag{4.20}$$

where the kinetics are given by:

$$f(\eta, \theta) = A_1(1-\theta)\eta - A_2\eta^3 - B(\theta - \alpha), \tag{4.21}$$

$$g(\eta, \theta) = C(1 + k_2\eta)(1-\theta)(1-\gamma(1-\theta)) - D(\theta(1-\gamma\theta) + k_3\eta\theta(1+\gamma\theta)). \tag{4.22}$$

The key idea behind the reaction-diffusion model, proposed in [10], is the coupling of one equation for the morphology η with one for the surface chemistry θ. $\eta \in \mathbb{R}$ is adimensional and expresses the instantaneous increment of the electrodeposit profile during the electrochemical process. $0 \leq \theta \leq 1$ is the surface coverage with the functionally crucial adsorbed chemical species. The nonlinear kinetics (4.21)-(4.22) account for generation (deposition) and loss (corrosion) of the relevant material during an electrodeposition process. All parameters in (4.21)-(4.22) are taken as real positive or equal to zero, with $0 < \gamma \leq 1$, $k_3 < k_2$, $D = C\frac{(1-\alpha)(1-\gamma+\gamma\alpha)}{\alpha(1+\gamma\alpha)}$. $P_e = (\eta_e, \theta_e) = (0, \alpha)$ is a spatially independent equilibrium for any choice of parameter values. Since P_e is characterized by $\eta_e = 0$, it corresponds to a flat electrode surface, from which corrugation and morphology can develop and for this reason it is relevant from the physical point of view.
In principle, the parameters for a DIB-PIP would be $\mathbf{p} = (C, B, A_1, A_2, \alpha, k_2, k_3, \gamma, D) \in$

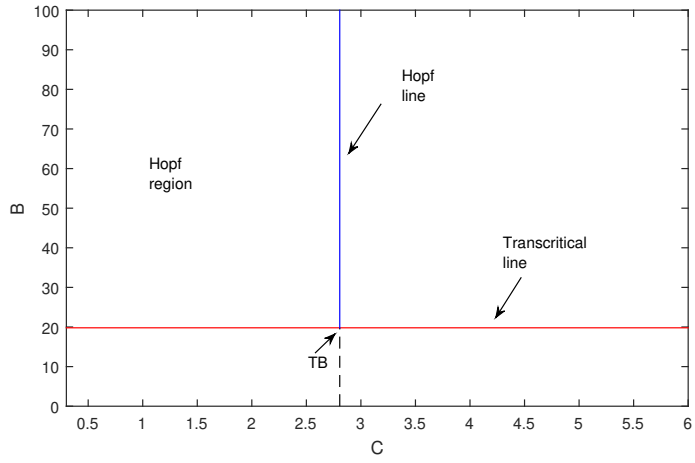

Figure 4.7: Bifurcation diagram in the parameter space (C, B). The Hopf region where oscillatory solutions are present is shown. $(C_{TB}, B_{TB}) = (2.8061, 19.7979)$ is the bifurcation point where transcritical and Hopf lines meet.

$\mathbb{R}^9$, but here first of all we consider $\mathbf{p} = (p_1, p_2) = (C, B) \in \Omega \subset \mathbb{R}^2$. In fact, theoretical stability analysis [47] allows to describe the main properties of the DIB model in terms of only two bifurcation parameters, such that stationary (Turing pattern) and oscillatory solutions are present (Hopf and Turing-Hopf instabilities). For this reason, we fix all the other parameters as in [47], that is: $\alpha = 0.5$, $\gamma = 0.2$, $k_2 = 2.5$, $k_3 = 1.5$, $A_1 = 10$, $A_2 = 30$. Since we are interested in the ODE (spatially independent) model with oscillatory solutions, we focus only on the Hopf instability, so that we shall consider parameters $\mathbf{p} = (p_1, p_2) = (C, B)$ belonging to the Hopf region. In Figure 4.7, we report the bifurcation diagram in the parameter space (C, B) showing the Hopf region for the above choice of the other parameters. From the analysis in [47], the transcritical and Hopf lines meet in $(C_{TB}, B_{TB}) = (2.8061, 19.7979)$. Therefore, the analysis reported in [47] suggests to study the DIB-PIP (4.19)–(4.22) in the parameter space $\Omega = [C_0, C_f] \times [B_0, B_f] = [1, 2.8] \times [21, 80]$. As before, the "classical" direct approach uses the 2-norm:

$$J_{norm2}(C, B) = \|\Theta(C, B) - \widetilde{\Theta}_{exp}\|_2^2 \tag{4.23}$$

where $\Theta(C, B) = [\Theta_0, ..., \Theta_N]$ is the numerical approximation of $\theta(t)$ in (4.20) with timestep $h = \frac{T - t_0}{N}$, that is $\Theta_k \approx \theta(t_k), \quad k = 0, .., N$ on $t_k = t_0 + k\,h$. $\widetilde{\Theta}_{exp}$ is the vector of the target data. Here $\widetilde{\Theta}_{exp}$ represents experimental data,

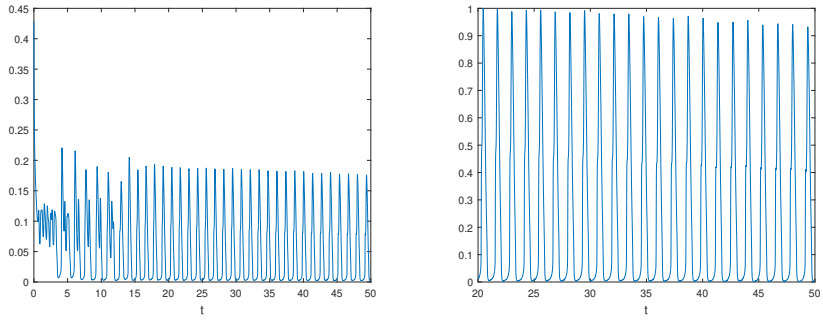

Figure 4.8: DIB-PIP: Left: Original experimental data of $\widetilde{\Theta}_{exp}$ for Zn dissolution in systems relevant to alkaline metal-air batteries. Right: Normalized and reduced data values $\widetilde{\Theta}_{exp}$ for $t \in [20, 50]$.

such that $\widetilde{\Theta}_{exp} = [\widetilde{\Theta}_0, ..., \widetilde{\Theta}_{\widetilde{N}}] \in \mathbb{R}^{\widetilde{N}+1}$ on the time interval $t \in [0, T] = [0, 50]$ obtained every timestep $h_{exp} = 0.01$. The initial conditions in (4.20) are chosen as $\eta_0 = 0.1$, $\theta_0 = \widetilde{\Theta}_0 = 0.4283$. Figure 4.8 (left) reports a piece of these original experimental data and is an anticipation of a more comprehensive numerical and physico-chemical study in preparation on the dynamics of the oscillatory behaviour of zinc (Zn) dissolution in systems relevant to alkaline metal-air batteries. The specific case reported corresponds to a representative interval of the current density oscillations obtained by fixing the electrode potential to -1070 mV vs Hg/HgO (mercury/mercury oxide electrode) for a mechanically polished Zn electrode in contact with 3M NaOH (sodium hydroxide solution). In [46] we have shown that the space-time dynamics of the electrochemical behaviour of Zn in aqueous alkaline solution can be described within the framework of the DIB model.

As in the previous Section dealing with simulated data, in the PIP we neglect the transient dynamics and we focus on the asymptotic regular oscillatory behavior of normalised data (maximum amplitude equal to one). We obtain $\Theta(C, B)$, the numerical solutions of the DIB-ODE system, by RK4 method with timestep $h = 0.005$, we normalize them and extract the values for $t \in [\bar{t}, T] = [20, 50]$. We normalize also the data $\widetilde{\Theta}_{exp}$ and we interpolate them on the ODE grid, in order to have in (4.23) vectors of same size (Figure 4.8, right). For abuse of notation, we continue to call $\Theta(C, B), \widetilde{\Theta}_{exp} \in \mathbb{R}^{N_s}$ the reduced vectors.

We discretize the parameter space Ω by using $h_C = 0.025$ and $h_B = 0.25$ and

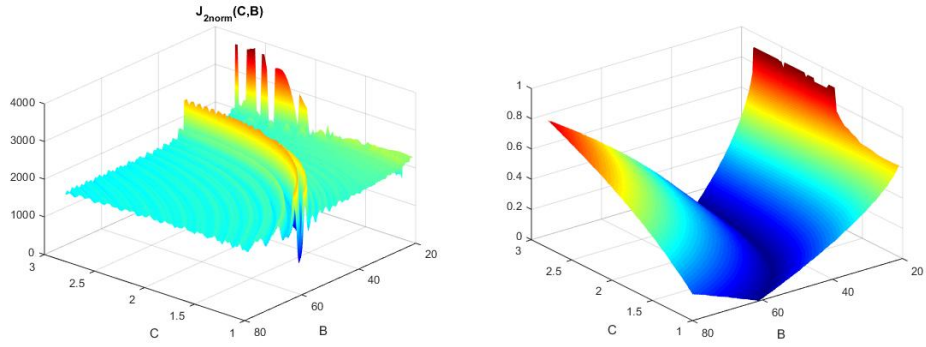

Figure 4.9: DIB-PIP: Least squares cost function (4.23) (top) and Fourier cost (4.24) (bottom)

we construct the discrete set Ω_h where we evaluate the 2-norm (4.23), that is shown in Figure 4.9 (left). Also in this case the cost function has many different low local minima. To apply the Fourier regularization we consider the following Fourier cost

$$J_{FFT}(C,B) = \frac{|f_1(C,B) - \widetilde{f}_1|}{\widetilde{f}_1}, \tag{4.24}$$

where $\widetilde{f}_1 = 0.7599$ is the first frequency of $\widetilde{\Theta}_{exp}$ calculated as before by the FFT spectrogram of the true data shown in Figure 4.10, and $f_1(C,B)$ is the first dominant frequency of the numerical solution $\Theta(C,B)$, $(C,B) \in \Omega_h$, computed in the same way. $J_{FFT}(C,B)$, shown in Figure 4.9 (right), presents a valley of (C,B) values in which $J_{FFT}(C,B) = 0$: for these pairs of parameters it follows that $\Theta(C,B)$ and the experimental data $\widetilde{\Theta}_{exp}$ have the same first frequency. Then we can define the discrete set:

$$\Omega_h^{FFT} = \{(C_i,B_j) \in \Omega_h \quad | \quad J_{FFT}(C_i,B_j) = 0\}. \tag{4.25}$$

The set Ω_h^{FFT} is represented in Figure 4.11 (left, 'o' symbol): clearly we can identify an iso-frequency curve in the C-B plane. By numerical interpolation we find a continuous form of this curve: $B = \mathcal{S}(C)$, $C \in [C_0, C_f]$. For this goal we calculate a cubic spline approximation by applying the default MATLAB [56] command *spline*, $\mathcal{S}(C)$ is also reported in Figure 4.11, left. On the iso-frequency curve $(C, \mathcal{S}(C))$, we can evaluate different cost functions: the original least squares and the infinity norms given by

$$J_\infty^{\mathcal{S}}(C) = J_\infty(C,B)_{|\mathcal{S}} = \|\Theta(C,B) - \widetilde{\Theta}_{exp}\|_\infty, \quad C \in [C_0, C_f], \tag{4.26}$$

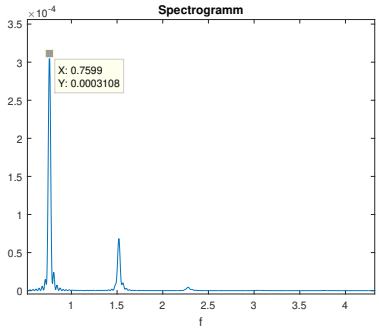

Figure 4.10: DIB-PIP: spectrogram of the experimental data $\widetilde{\Theta}_{exp}$ in Figure 4.8. The first dominant frequency is $\widetilde{f}_1 = 0.7599$.

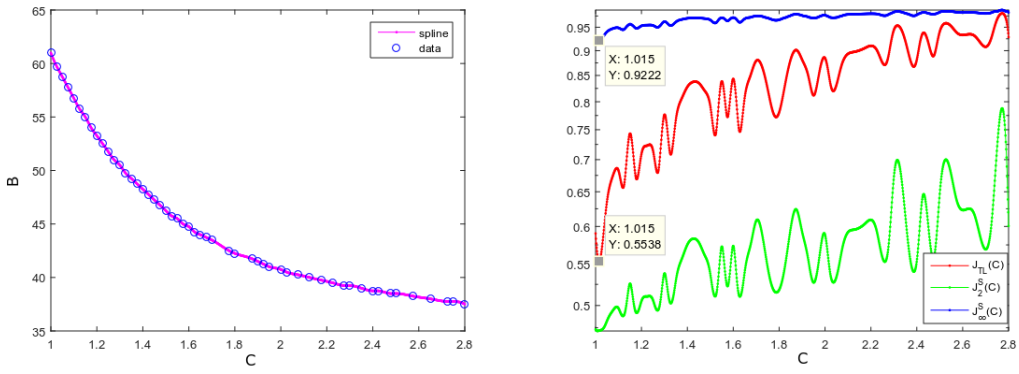

Figure 4.11: DIB-PIP: The set Ω_h^{FFT} in (4.25) and the interpolating spline $B = \mathcal{S}(C)$ (left); cost functions (4.26)–(4.29) (right) for the (C, B) values on the spline. Note that the value of $J_2^{\mathcal{S}}(C)$ has been divided by its maximum value to make to costs comparable.

$$J_2^{\mathcal{S}}(C) = J_{2norm}(C,B)_{|\mathcal{S}} = \|\Theta(C,B) - \widetilde{\Theta}_{exp}\|_2^2, \quad C \in [C_0, C_f], \tag{4.27}$$

and the time-lag of the numerical solution with respect to the experimental data in the first and in the last points of the interval given by

$$J_{TL}(C) = J_{TimeLag}(C,B)_{|\mathcal{S}} = |\Theta_1(C,B) - (\widetilde{\Theta}_{exp})_1|, \quad C \in [C_0, C_f], \tag{4.28}$$

$$J_{TLE}(C) = J_{TimeLagEnd}(C,B)_{|\mathcal{S}} = |\Theta_N(C,B) - (\widetilde{\Theta}_{exp})_N|, \quad C \in [C_0, C_f]. \tag{4.29}$$

All these costs are shown in Figure 4.11, where we can see that they grow as the C value increases. (Note that the value of $J_2^{\mathcal{S}}(C)$ has been divided by its maximum value to make the costs comparable.) All costs have an absolute minimum in $(C_1, B_1) = (C_1, \mathcal{S}(C_1)) = (1.015, 60.2017)$ where $(J_2^{\mathcal{S}}(C_1,B_1), J_\infty^{\mathcal{S}}(C_1,B_1), J_{TL}(C_1,B_1)) =$

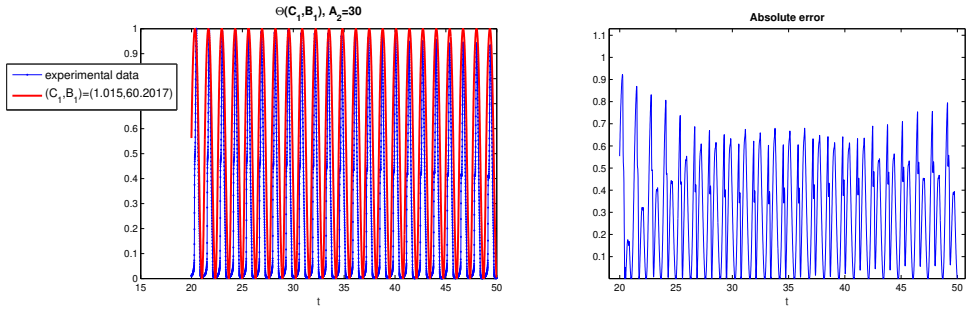

Figure 4.12: DIB-PIP: Left, experimental data $\widetilde{\Theta}_{exp}$ compared with the simulation $\Theta(C_1, B_1)$ for the Fourier-PIP optimal parameter set (C_1, B_1). Right: corresponding absolute error $Err_1 = |\Theta(C_1, B_1) - \widetilde{\Theta}_{exp}|$.

$(27.1495, 0.9222, 0.5538)$. Figure 4.12, left, shows the corresponding numerical solution $\Theta(C_1, B_1)$ compared with the experimental data. In Figure 4.12, right, we report also the time behaviour of the absolute error $Err_1 = |\Theta(C_1, B_1) - \widetilde{\Theta}_{exp}|$. We note that the same frequency is present and the largest error is due to the "shape" of oscillations.

In conclusion, the Fourier regularization for the DIB-PIP identifies only a unique optimal parameter set such that the maximum error wrt to the data is $max(Err_1) = 0.9222$ (see Figure 4.11, right). In principle, we could use (C_1, B_1) as the starting guess of an optimization algorithm for DIB-PIP that minimizes the classical 2-norm in (4.23), but this study would be beyond the scope of the present paper that is focused on the advantages of the Fourier approach. Moreover, we wish to show that the Fourier regularization can be naturally extended to the case in which more than two parameters have to be identified. In the next subsection, we shall thus describe this extension on the DIB-PIP to the case of $m = 3$ parameters.

4.3.1 DIB-PIP: Fourier regularization for $m = 3$ parameters

Considering the same experimental data in Figures 4.8-4.10, we wish to identify the parameters (C, B, A_2) in (4.20)–(4.22). In particular, we shall relax the constraint of setting A_2 fixed to $A_2 = 30$. The parameter space thus becomes the following subset of the Hopf region $\Omega = [C_0, C_f] \times [B_0, B_f] \times [A_2^0, A_2^f] = [1, 1.6] \times [51, 61] \times [1, 40]$ (the intervals for C and B are smaller than before). We apply the Fourier regularization as follows. We consider the set Ω_h, the discretiza-

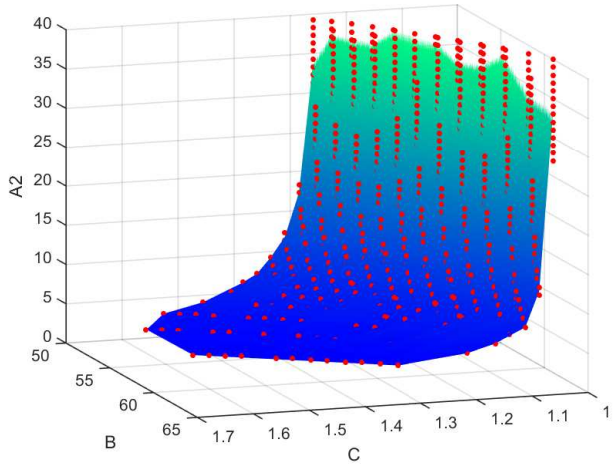

Figure 4.13: DIB-PIP: The set Ω_h^{FFT} (red points) and its interpolating surface $A_2 = \Phi(C, B)$.

tion of Ω with stepsizes $h_C = 0.025$, $h_B = 0.25$ and $h_{A_2} = 1$, and we compute the Fourier cost on Ω_h:

$$J_{FFT}(C, B, A_2) = \frac{|f_1(C, B, A_2) - \widetilde{f}_1|}{\widetilde{f}_1} \tag{4.30}$$

As in the previous Section, we obtain now

$$\Omega_h^{FFT} = \{(C_i, B_j, A_{2,k}) \in \ \Omega_h \mid J_{FFT}(C_i, B_j, A_{2,k}) = 0\} \subset \mathbb{R}^3.$$

Figure 4.13 shows the points (triplets) $P_l = (X_l, Y_l, Z_l) \in \Omega_h^{FFT},\ l = 1, \ldots, N_P$ (red symbols) and a 3D interpolating surface $A_2 = \Phi(C, B)$, that we have approximated by using the command *griddata* in MATLAB. Φ identifies an iso-frequency surface, that is a sub-manifold of codimension one in $\mathbb{R}^3$, such that data and simulations have the same frequency for triplets of parameters, that is points, on Φ.

To solve the Fourier-PIP we have to evaluate the usual costs on the iso-frequency manifold Φ. To extend the approach explained in the case of only two parameters, we proceed as described below. We extract the points of Ω_h^{FFT} at each A_2-level for fixed values $A_{2,k}, k = 1, \ldots, N_A$ and then we compute the parametric interpolating 3D curve in the plane (C, B, Z_k) for all $k = 1, \ldots, N_A$. Hence, we calculate in MATLAB [56] the interpolating splines: $B = \mathcal{S}(C; A_{2,k}) = \mathcal{S}_k(C)$, in the plane (C, B) at the level $A_{2,k}$, for $k = 1, \ldots, N_A$ (the curves are not shown). Hence, for all $k = 1, \ldots, N_A$, we can evaluate the usual costs on each parametric

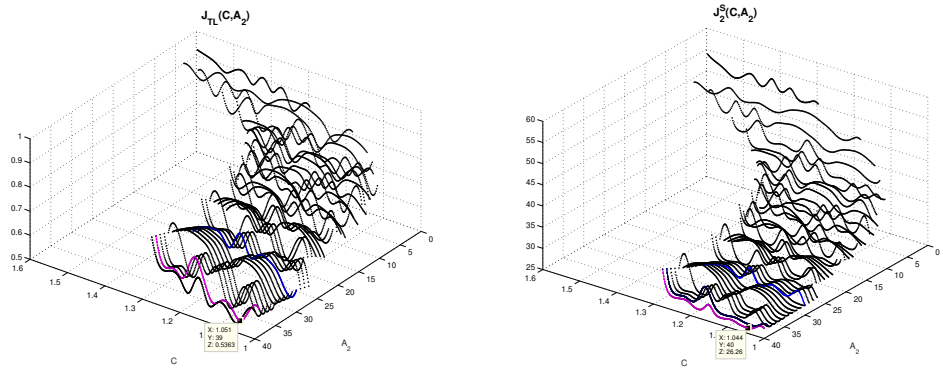

Figure 4.14: DIB-PIP for $m = 3$ parameters: cost functions (4.33) (left) and (4.32)(right) evaluated on the curves $B = \mathcal{S}(C, A_{2,k})$ for $k = 1, \ldots, N_A$, $N_A = 40$ (increasing values of A_2). In each subplot, the cost function $J^{\mathcal{S}_k}$ which contains the absolute minimum is emphasized in magenta. The blue one is that corresponding to the value $A_2 = 30$ studied in the previous Section.

curve $\mathcal{S}_k$ as follows :

$$J_\infty^{\mathcal{S}_k}(C) = J_\infty(C, B, A_2)_{|_{\mathcal{S}_k}} = \|\Theta(C, B, A_2) - \widetilde{\Theta}_{exp}\|_\infty \tag{4.31}$$

$$J_2^{\mathcal{S}_k}(C) = J_{2norm}(C, B, A_2)_{|_{\mathcal{S}_k}} = \|\Theta(C, B, A_2) - \widetilde{\Theta}_{exp}\|_2^2 \tag{4.32}$$

$$J_{TL}^{k}(C) = J_{TimeLag}(C, B, A_2)_{|_{\mathcal{S}_k}} = |\Theta_1(C, B, A_2) - (\widetilde{\Theta}_{exp})_1| \tag{4.33}$$

For simplicity of exposition, here we do not consider the time-lag error in the last point. We represent the above projected costs as functions of C and A_2 in Figure 4.14, for $k = 1, \ldots, N_A$ ($N_A = 40$), that is for the chosen discrete values of A_2 in Ω_h, (including the value $A_2 = 30$ considered in the 2-parameter case). We decided to not represent $J_\infty^{\mathcal{S}_k}$ because it is analogous to J_{TL}^k for all k.

$J_2^{\mathcal{S}}$ has a minimum in $\mathbf{p}' = (C, B, A_2) = (C, \mathcal{S}(C, A_2), A_2) = (1.0442, 58.7684, 40)$, where $(J_{TL}(\mathbf{p}'), J_\infty^{\mathcal{S}}(\mathbf{p}'), J_2^{\mathcal{S}}(\mathbf{p}')) = (0.5386, 0.9214, 26.26)$. In Figure 4.14, the cost function $J^{\mathcal{S}_k}$ which contains the minimum is shown in magenta, while the cost for $A_2 = 30$ is reported in blue.

J_{TL} (and $J_\infty^{\mathcal{S}}$) has a different low minimum in $\mathbf{p}'' = (C, B, A_2) = (C, \mathcal{S}(C, A_2), A_2) = (1.0511, 58.4498, 39)$, where $(J_{TL}(\mathbf{p}''), J_\infty^{\mathcal{S}}(\mathbf{p}''), J_2^{\mathcal{S}}(\mathbf{p}'')) = (0.5363, 0.9196, 26.34)$. We see that the parameter sets are different, but the cost values (residuals) are very similar. By calculating, for example, $\min\{J_{TL}(\mathbf{p}'), J_{TL}(\mathbf{p}'')\}$ we can identify $\mathbf{p}^* = \mathbf{p}''$ as the PIP-Fourier optimal solution on three parameters that has

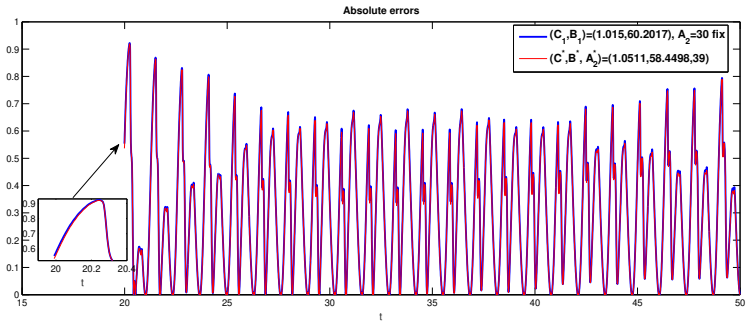

Figure 4.15: DIB-PIP: Absolute errors wrt to data of the numerical solutions $\Theta(C_1, B_1), A_2 = 30$ and $\Theta(\mathbf{p}^*)$ obtained as PIP-Fourier optimal solutions in the case of two and three parameters, respectively.

the minimum time-lag error. Note that the optimal values of (C^*, B^*) are different form those identified by PIP-Fourier on two parameters and, in particular, $A_2^* \neq 30$ the value that was kept fixed in the two-parameter optimization.

Therefore, we would like to compare the numerical solutions of the DIB-ODE model (4.20)–(4.22) corresponding to the optimal triplet $\mathbf{p}^*$ and the optimal couple $(C_1, B_1) = (1.015, 60.2017), A_2 = 30$. To this aim, in Figure 4.15 we compare the corresponding absolute errors wrt to the data. Along the time interval the errors for $\Theta(\mathbf{p}^*)$ are sligthly lower than those for $\Theta(C_1, B_1), A_2 = 30$ fixed, even if the maximum errors (0.9196 and 0.9222) are very similar (see the zoom inset on the left). $\mathbf{p}^*$ could be also used as starting guess in an optimization algorithm to solve the original DIB-PIP in (4.19)-(4.20), but - as anticipated above - this is beyond the scope of this paper.

Moreover, we believe that the PIP Fourier regularization presented in this subsection could be applied recursively by choosing as third value to be identified by another parameter in the DIB-ODE model (4.20) different from A_2, so that better data approximation could be obtained.

4.4 Application: Dynamics of zinc-air battery anodes

Electricity demand is growing systematically and fossil fuel production is not sustainable. Renewable approaches, such as solar and wind power, could replace fossil fuels, but it is crucial the availability of safe and efficient systems for the accumulation in the different power requirements of settlements, transport and

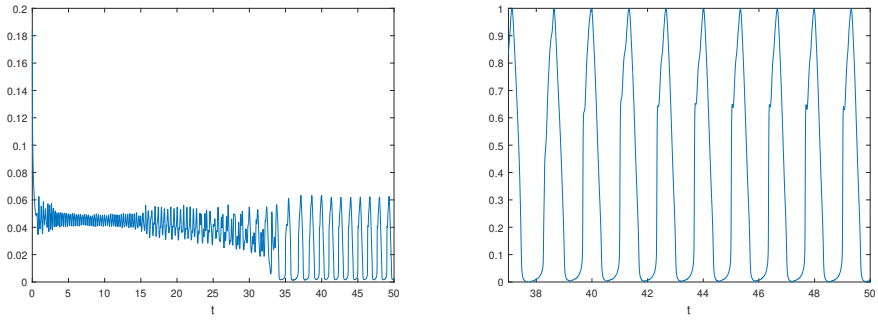

Figure 4.16: On the left a new set of experimental data is shown. On the right we can observe the piece of the normalized data for $t \in [37, 50]$.

industry. Research is actively engaged in identifying new strategies and in this context electrochemical technologies play a key role. Therefore the metal-air zinc-air batteries, in particular, constitute strategic alternatives that need the deepening of the relevant technologies. In [8] we present a dynamic study of the behavior of anodes Zn in aqueous solution of 6M KOH (potassium hydroxide), based on electrochemical techniques combined with measurements of spectral electromagnation visible in situ. Electrochemical measurements demonstrate a wide variety of dynamic scenarios, comprising active-passive transitions and different oscillating regimes, as shown in Figure 4.8 or in Figure 4.16, that shown another set of experimental data. Dynamic processes have been rationalized within the DIB mathematical model of the electrometallurgical phase formation process, based as we have seen on a system of two equations ordinary physical differentials respectively for the morphology and for the degree of coating with pseudopassive film. The source terms of the model contain simple relative information electrocinetic and adsorption electrochemical, formulated in terms of installments phenomenological equation. In particular, it is possible to follow the oscillating regimes of current and reflectivity with the model, reproducing the details of the structure process dynamics and identifying the physical parameters. The parameters are located between the Hopf line and the transcritical line in the absence of oscillation, in the Hopf region in the case of harmonic oscillations and finally in the set $\Omega_h^{\mathcal{K}}$, defined in the following Section, in the case of the so-called "relaxation oscillations". As explained in [46], the values of parameters B and C are physically traceable to the operating conditions of the electrochemical process.

4.4.1 Relaxation oscillations

As we can observe in the Figures 4.8-4.16, the exhibited oscillations seem to be a kind of relaxation oscillation. The relaxation oscillation is a specific type of oscillation and an oscillator that exhibits these kind of oscillations is called relaxation oscillator. A relaxation oscillator is an oscillator based upon the behavior of a physical system's return to equilibrium after being disturbed. That is, a dynamical system within the oscillator continuously dissipates its internal energy. Normally, the system would return to its natural equilibrium; however, each time the system reaches some threshold sufficiently close to its equilibrium, a mechanism disturbs it with additional energy. Hence, the oscillator's behavior is characterized by long periods of dissipation followed by short impulses [94]. In the case of the relaxation oscillation the limit cycles shows a "sharp shape"; in particular it changes its concavity along one period: the limit cycle is not elliptic as is usual sinusoidal oscillations. Hence we look for a change of concavity in the limit circle for the numerical solutions of (4.20) with the parameters $(C, B) \in \Omega_h$. Let $\alpha(t)$ be the parametric equation of the limit cycle as a curve in the $\eta - \theta$ plane:

$$\alpha(t) = (\eta(t), \theta(t)). \tag{4.34}$$

To compute the algebraic curvature, as defined in (B.2), we need the second derivative of $\eta(t)$ and $\theta(t)$. Let us compute $\eta''(t)$ and $\theta''(t)$, in terms of $\eta(t)$, $\theta(t)$, $\eta'(t)$ and $\theta'(t)$ (we omit the dependence on time t to simplify the notation):

$$\begin{cases} \eta' = A_1(1-\theta)\eta - A_2\eta^3 - B(\theta - \alpha)), \\ \theta' = C(1+k_2\eta)(1-\theta)(1-\gamma(1-\theta)) - D(\theta(1-\gamma\theta) + k_3\eta\theta(1+\gamma\theta)), \end{cases} \tag{4.35}$$

$$\Downarrow$$

$$\begin{cases} \eta'' = A_1(-\theta')\eta + A_1(1-\theta)\eta' - 3A_2\eta^2\eta' - B(\theta'), \\ \theta'' = C(1+k_2\eta')(1-\theta)(1-\gamma(1-\theta)) + C(1+k_2\eta)(-\theta')(1-\gamma(1-\theta)) + \\ \quad + C(1+k_2\eta)(1-\theta)(\gamma\theta') - D[\theta'(1+\gamma\theta) + \theta(\gamma\theta') + k_3\eta'\theta(1+\gamma\theta) + k_3\eta\theta'(1+\gamma\theta) + \\ \quad + k_3\eta\theta(\gamma\theta')]. \end{cases} \tag{4.36}$$

We do not have the explicit expression of $\eta(t)$ and $\theta(t)$, but we can obtain them numerically. Let $\boldsymbol{H}(C,B) = [H_0, ..., H_N]$, $\boldsymbol{\Theta}(C,B) = [\Theta_0, ..., \Theta_N] \in \mathbb{R}^{N+1}$ be the numerical approximations of $\eta(t)$ and $\theta(t)$ with a timestep $h = \frac{T-t_0}{N}$ for $(C,B) \in \Omega_h$ and all the others parameters fixed as in Section 4.3. Therefore we can compute the vectors $\boldsymbol{H}'(C,B)$, $\boldsymbol{\Theta}'(C,B)$, $\boldsymbol{H}''(C,B)$ and $\boldsymbol{\Theta}''(C,B)$ as the approximations of η', θ', η'' and θ'' respectively from the analytic expressions in (4.35)-(4.36). Then, from (B.2) in Appendix B, it results:

$$\mathcal{K}(C,B) = \frac{\boldsymbol{H}'(C,B)\boldsymbol{\Theta}''(C,B) - \boldsymbol{H}'(C,B)\boldsymbol{\Theta}''(C,B)}{\left(\sqrt{\boldsymbol{H}'^2(C,B) + \boldsymbol{\Theta}'^2(C,B)}\right)^3}. \tag{4.37}$$

The limit cycle changes its trend for the values in Ω: we can search if there are some parameters for which we have a type of "relaxation oscillation". For this purpose we extend the domain Ω for $B < 21$ and $B > 80$. For abuse of notation let us denote $\Omega = [0, 2.8] \times [0, 100]$ and Ω_h the corresponding discrete set computed by using $h_C = 0.025$ and $h_B = 0.25$. Let us now compute $\mathcal{K}(C,B)$ for all $(C,B) \in \Omega_h$ and look for the pairs for which $\mathcal{K}$ has an inflection point. Let $\Omega_{\mathcal{K}} \subset \Omega_h$ be the set that contains these pairs, that is:

$$\Omega_{\mathcal{K}} = \{(C_i, B_j) \in \Omega_h | \mathcal{K}(C_i, B_j) \text{ has an inflection point}\}. \tag{4.38}$$

The set $\Omega_{\mathcal{K}}$ is shown in Figure 4.17, compared with the set Ω_h^{FFT} of iso-frequency, defined in Section 4.3 in (4.25) for the experimental data in Figure 4.8.

Let us observe that the iso-frequency manifold is disjointed from the set $\Omega_h^{\mathcal{K}}$: that is, the relaxation oscillations in the C-B plane have different frequency wrt the experimental data. For this reason, in order to compare the "shape" of oscillation, we decide to consider the last three periods of simulations and experimental data; the piece of experimental data is shown in Figure 4.18 on the left. Since the frequency of the simulations and data is different, we rescale both in the time interval $[0, 1]$ and interpolate them in the same grid time in order to have vector of the same size. Let $\underline{\boldsymbol{\Theta}}(C,B) = [\underline{\Theta}_1, ..., \underline{\Theta}_{\overline{N}}] \in \mathbb{R}^{\overline{N}}$ and $\underline{\widetilde{\boldsymbol{\Theta}}}_{exp} = [\underline{\widetilde{\Theta}}_1, ..., \underline{\widetilde{\Theta}}_{\overline{N}}] \in \mathbb{R}^{\overline{N}}$ be the simulation and experimental pieces respectively, interpolated on the same time grid in $[0, 1]$. Let us compute the two costs defined as follows:

$$J_{\infty}(C,B)_{|\Omega_{\mathcal{K}}} = \|\underline{\boldsymbol{\Theta}}(C,B) - \underline{\widetilde{\boldsymbol{\Theta}}}_{exp}\|_{\infty} \tag{4.39}$$

$$J_{2norm}(C,B)_{|\Omega_{\mathcal{K}}} = \|\underline{\boldsymbol{\Theta}}(C,B) - \underline{\widetilde{\boldsymbol{\Theta}}}_{exp}\|_2^2. \tag{4.40}$$

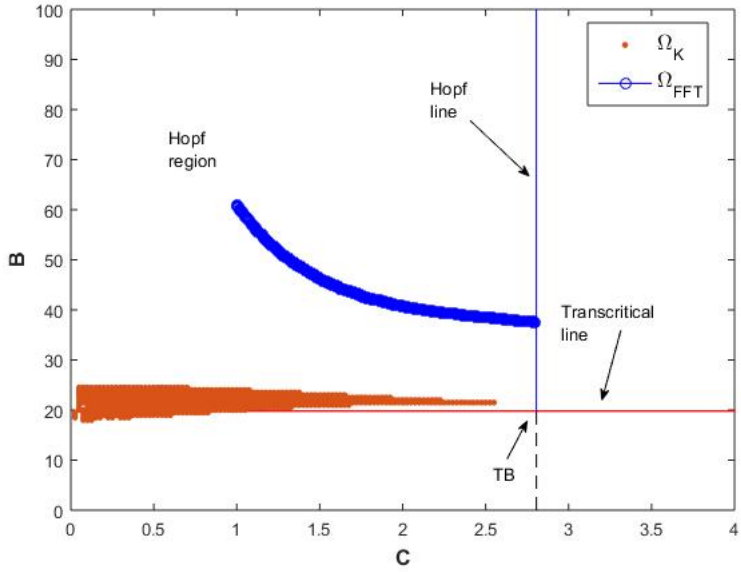

Figure 4.17: Set $\Omega_h^{\mathcal{K}}$ defined in (4.38), and set Ω_h^{FFT} in (4.25).

On the right of Figure 4.18 the minimum of (4.39) is shown for $(C, B) = (1.05, 20)$ and $(C, B) = (0.45, 20)$ for the two experimental data respectively, compared with the experimental pieces.

As we have seen, the search for parameters in the relaxation zone allowed us to make a first optimization on the oscillations "form", not considering their frequency. In this way we found the parameters in the C-B plane that approximate the shape of the periods as much as possible. It is worth noting that the experimental data exhibit a particular shape that the simulation can not fit by using the DIB model as it is.

4.5 Summary

The Fourier regularization method presented in this Chapter is a useful tool for PIP in ODE modelling of oscillating data. Our approach exhibits two key capabilities:

(i) it is able to find an iso-frequency manifold $\mathcal{S}$ of co-dimension one in the m dimensional parameter space Ω where target oscillatory data and simulations have the same frequency;

(ii) along this manifold the phase (time-lag) error and/or usual cost functions (e.g. least squares) can be straightforwardly minimized.

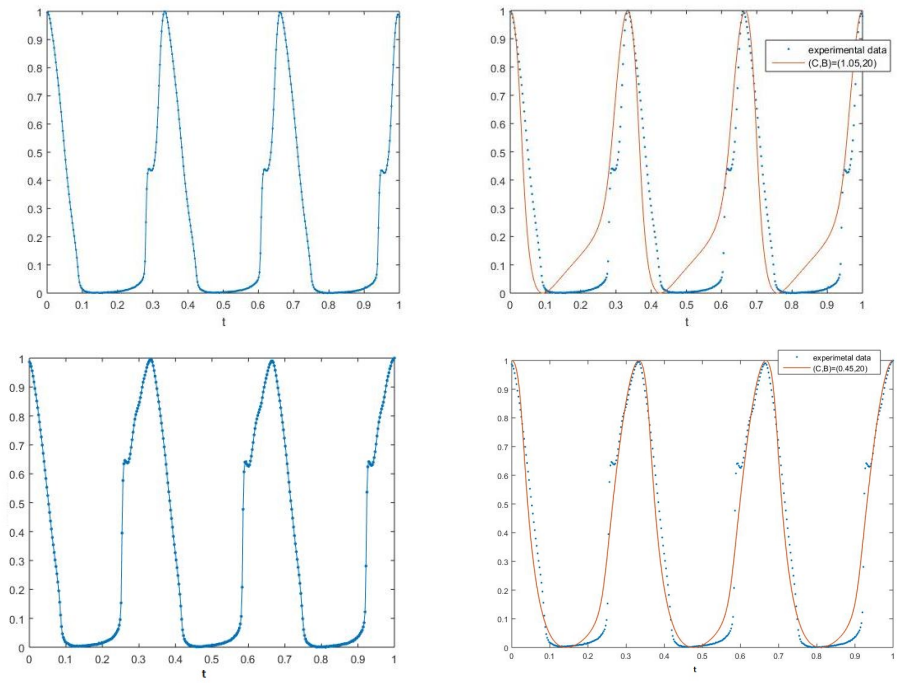

Figure 4.18: On the left the experimental pieces, on the right the comparison with the minimum of (4.39) for the experimental data in Figure 4.8 or in Figure 4.16 respectively.

Our Fourier regularization approach can thus be implemented as a two-steps algorithm for parameter identification or localization. In Sections 4.2 and 4.3, we have solved PIP for the Schnackenberg and the electrodeposition (DIB) ODE models to identify the two physically crucial parameters $\mathbf{p} = (p_1, p_2)$ in the case of simulated and experimental data. In both cases, we have shown that $\mathcal{S}$ is a parametric curve in the plane $p_1 - p_2$ obtained by (piecewise) interpolation. Then we have found optimal Fourier-PIP solutions by evaluating the appropriate cost functions along these curves. In Subsection 4.4.1, we have shown how to extend the Fourier regularization to the problem with $m = 3$ parameters on true data for the DIB-PIP model: in this case the manifold is an iso-frequency surface in $\mathbb{R}^3$. Then we have proposed a computational approach handling the above steps (i)-(ii) in an algorithmic way that projects the cost functions involved on a sequence of 3D spline curves. This strategy could be generalized in a recursive way to improve ODE-PIP data fitting including more parameters.

It is worth noting that the accuracy of the numerical method used for the approximation of the ODEs influences the Fourier-PIP approach, since it is crucial

to recognize the sub-manifold $\mathcal{S}$ in the parameters space. For this reason, here we applied the RK4, that has good dispersion order properties. We believe that our results show the effectiveness of our new regularization approach in comparision with the classical PIP approach based on least-squares cost function. Moreover, for problems in which a higher accuracy is required in the approximation, Fourier regularization can be regarded as a dedicated tool for the localization of starting guesses in classical numerical optimization algorithms.

In Section 4.4 we described a real application of the identification for the DIB model. Furthermore, in Subsection 4.4.1 we have fit the experimental data by looking for oscillations of relaxation type in the parameters space. Since the relaxation oscillations are not located in the iso-frequency manifold $\mathcal{S}$, we have extracted some periods of the experimental oscillation and found the simulation with the most similar trend.

Recent experiments allowed us to have the dynamics of both the variables η and θ of the model. This would imply that we could make a joint identification and formulate a fit of the experimental limit cycle for the ODE-PIP.

Chapter 5

PDE-PIP

5.1 Formulation for PDEs

We are also interested in defining the Parameter Identification Problem for Partial Differential Equations (PDE-PIP). In particular we will consider the case of time-dependent reaction-diffusion PDEs, whose solutions display a wide range of behaviors, including the self-organized pattern like stripes, spirals..., the so-called Turing patterns [89]. Parameter estimation in Turing system is an active field of research and an increasing amount of paper deals with this topic: see for example [15, 32, 93, 81].

Then the differential model in (1.2) (the constraint of the PDE-PIP) will assume the following dimensional form:

$$\begin{cases} u_t = d_1 \Delta u + f_1(u, v, \mathbf{p}), \\ v_t = d_2 \Delta v + f_2(u, v, \mathbf{p}), \\ u(x, y, t_0) = u_0, v(x, y, t_0) = v_0 \quad t \in [t_0, T_f],\ (x, y) \in \mathcal{D} \subset \mathbb{R}^2 \end{cases} \tag{5.1}$$

with appropriate boundary conditions, where Δ is the Laplace operator, d_1 and d_2 are the diffusion coefficients, f_1 and f_2 contains the nonlinear reaction terms and $\mathbf{p}$ represents the parameters set of the model. In case of PDE the target is provided as a map on a given discretization of the space domain, at a fixed time T. It will be a digital image given by a matrix of values which represents a desired configuration of the system. Hence in PDE-PIP we are not interested in the time evolution of the pattern deriving from the time integration of (5.1), but we consider the solution of (5.1) at a fixed time T_f and compare it with the

target map. The final time T_f must be specified a priori. Since we assume that the data map corresponds to a steady-state solution then it is sufficient to specify a final time that is long enough for the PDE steady state to be reached. We will discard possible optimal solutions that are not steady-state PDE solutions.

In general, we can formulate the PDE-PIP as follows: given an experimental map $\widetilde{\mathbf{M}} \in \mathbb{R}^{N_x \times N_y}$, the time integration interval $[0, T_f]$ and the initial conditions (u_0, v_0), we look for a suitable parameter set $\mathbf{p}^* \in \mathrm{R}^r$ for the model (5.1) such that $v(x, y, T_f) \approx \widetilde{\mathbf{M}}$, where v is the solution of the reaction-diffusion PDEs system, that is:

$$J(\mathbf{p}^*) = \min_{\mathbf{p}} J(\mathbf{p}) \tag{5.2}$$

where $J(\mathbf{p})$ is an suitable cost function that depend on the experimental map $\widetilde{\mathbf{M}}$, the solution of the system (5.1) v and implicitly on the parameters, and measures a certain "distance" between $\widetilde{\mathbf{M}}$ and v:

$$J(\mathbf{p}) = \int_{\mathcal{D}} (v(x, y, T_f) - \widetilde{\mathbf{M}})^2 dxdy \tag{5.3}$$

To ensure the uniqueness of the solution, often a regularization term is added [38], as we discussed in the PIP-ODE case and we show below in the description of the discretization issues for the PIP-PDE. The Direct (and Indirect) approach can be reformulated in terms of PDEs by suitable changes. In particular, the numerical method $\mathcal{M}$ to define the constraint in (1.5) is suitably substitutes by a numerical method for PDEs. Then PDEs system (5.1) is solved numerically on a spatial mesh-grid (x_i, y_j) $i = 1, ..., N_x,\ j = 1, ..., N_y$; we write the numerical solution at discrete times $t_n = nh_t$ as $\mathbf{V}_{ij}^n \approx v(x_i, y_i, t_n)$. Then the PDE-PIP in discrete formulation reads as follows: given an experimental map $\widetilde{\mathbf{M}} \in \mathbb{R}^{N_x \times N_y}$, the initial conditions $\mathbf{U}_0,\ \mathbf{V}_0 \in \mathbb{R}^{N_x \times N_y}$, and the final integration time $T_f > 0$, find the parameters $\mathbf{p}^* \in \mathbb{R}^r$ that minimize the cost functional

$$J(\mathbf{p}) = \|\mathbf{V}_{T_f}(\mathbf{p}) - \widetilde{\mathbf{M}}\|_W^2, \tag{5.4}$$

subject to the discrete model

$$[\mathbf{U}_{T_f}, \mathbf{V}_{T_f}] = \mathcal{P}(\mathbf{U}_0, \mathbf{V}_0, \mathbf{p}) \tag{5.5}$$

where $\mathcal{P}$ indicates the numerical method yielding the discrete version of the system (5.1) and W is a suitable weighting matrix. The initial condition are taken as a

perturbation of the equilibrium:

$$\begin{aligned} \mathbf{U}_0 &= \mathbf{U}_e + c_e X_u \\ \mathbf{V}_0 &= \mathbf{V}_e + c_e X_v \end{aligned} \tag{5.6}$$

where X_u and X_v are random matrices and c_e is a small scalar factor.

5.2 Discretization issues for PDE-PIP

The formulation of the discrete PDE-PIP by using the Direct approach introduces similar issues as the ODE case; in fact for the numerical resolution of the PIP we need:

- a PDEs solver
- a (weighted) norm for the cost function
- an optimization algorithm for the minimization.

PDEs solver: The PDEs solver $\mathcal{P}$ has to cope with problems that can be computationally very expensive. First of all, we are interested in the stationary pattern reached by the model for long integration time T_f; furthermore, the structure of some type of patterns will not emerge if the domain is small, as we will show next in details. Therefore we have to solve the PDEs on a large spatial domain, which implies the need to consider a very fine spatial grid, for long times. For these reason it is crucial to have efficient methods in terms on computational time and accuracy for the numerical approximation of the PDE. In the following Section 5.3 we report the results obtained in [21] and we study the resolution of the RD-PDEs introducing the *matrix formulation* of the problem that allows us to solve the differential model by using efficient methods for the matrix equations, based on the spectral decomposition of the coefficient matrices.

Cost function: The choice of a suitable norm as cost function is crucial in the construction of the discrete problem and in the optimization process. For example, if we had information about the map, a weighted norm can be chosen: as explained for example in [81], the weighting matrix W could allow the model to fit some parts better than others. In fact, the weighting matrix is often diagonal,

in which case it gives different relative emphasis to different components of the map. Furthermore, if the correlations between errors in different parts of the data map are known, then we can include these correlations in the off-diagonal elements of W. In [81] the authors present a first example of the fitting of an experimental electrochemical morphochemical distribution with the DIB model, introduced in [10, 47], and they investigated the use of a weighting matrix based on the data values themselves, that is the choice of W as a diagonal matrix with entries dependent on the values of the map $\widetilde{\mathbf{M}}$. By giving more importance to the pixels of the map where the feature of the pattern are present, they aimed to emphasize the fit to the patterns evident in the data.

Optimization algorithm: The numerical minimization of the cost function, as well as the ODE case, requires the use of an optimization algorithm to approach the minimum. Similarly to the previous case, we remember that many papers are focused on the choice of the optimization algorithm also in the PDE case. As example, in [93] the authors present an iterative algorithm for solving a parameter identification problem relative to a system of diffusion, convection and reaction equations, which solves a nonlinear least squares problem by means of a sequence of constrained optimization problems. In [32] an algorithm for the parameter estimation in the Turing system was introduced, by applying the optimal control theory; it is one of the first work that address parameter estimation for the Turing reaction-diffusion model. A further work on estimating parameters in Turing models is [81], where the authors proposed a two-step algorithm for the optimization. It is well known that the Turing instability produces different kind of pattern in the parameter space. Therefore, they want to identify:

i. at first, the position in the parameter space of the given pattern class;

ii. then, the unique or an optimal solution in this class/ subregion.

To tackle the point (i), they proposed to use the classical 2-norm cost function such that its lowest values can identify numerically the minimum value (C_0, B_0) into a sub-region of the Turing space where qualitatively similar solutions are present. They named this procedure PIP(i). To solve the above point (ii), they added a classical Tikhonov regularization term (see [38]) to the cost function (5.4),

centered on values (C_0, B_0) found by PIP(i). Thus, the original cost function was replaced by:

$$J(C,B) = \|\mathbf{V}_{T_f}(\mathbf{p}) - \widetilde{\mathbf{M}}\|_W^2 + \gamma_B\|B - B_0\| + \gamma_C\|C - C_0\| \tag{5.7}$$

where γ_B and γ_C are weighting suitable chosen parameters. In solving this optimization problem, called PIP(ii), they used the Polack-Ribiere flavor of a conjugate gradient method [63]. As the authors in [81] clearly explained, this gradient-based method requires the gradient of the objective function with respect to the parameters on each iteration, that clearly depend on the numerical method $\mathcal{P}$ chosen for the numerical approximation of the PDEs. In practice, it can be derived from the source code of the nonlinear model, in a process known as automatic differentiation (e.g. [34]).

5.3 Matrix-oriented methods for the approximation of RD-PDEs

In this work, we are interested in solutions of (5.1) due to the *diffusion-driven or Turing instability*. In this case where the diffusion is present, the spatially homogeneous solution of (5.1) (u_e, v_e) such that $f_1(u_e, v_e) = f_2(u_e, v_e) = 0$ (stable in absence of diffusion) can force a spatial instability and asymptotically tends towards a so-called *Turing pattern* characterized by interesting spatial structures like spots, worms, labyrinths, etc. Moreover, in [61, 62], the authors prove that the transient dynamics is important for pattern formation. In particular, the concept of *reactivity* describing the short-term transient behavior, is necessary for Turing instabilities. Let be $\mathbf{w} = (u, v)$ and $J = J(u_e, v_e) = \begin{bmatrix} f_{1,u} & f_{1,v} \\ f_{2,u} & f_{2,v} \end{bmatrix}_{\mathbf{w}_e}$ the Jacobian of the linearized ODE system associated to (5.1) evaluated at the spatially homogeneous solution $\mathbf{w}_e = (u_e, v_e)$. For the Turing theory, the spatially homogeneous solution $\mathbf{w}_e = (u_e, v_e)$ is stable if the eigenvalues of J all have negative real parts, but $\mathbf{w}_e$ it is not necessarily stable for the RD-PDE. In [61], the authors defined $\mathbf{w}_e$ as *reactive equilibrium* if the largest eigenvalue of the symmetric part of J is positive:

$$\lambda_1(H(J)) > 0, \qquad H(J) = (J + J^T)/2.$$

If the initial conditions in (5.1) are a small (random) perturbation to $\mathbf{w}_e$, the RD-PDE solution in the initial transient, say $\mathbf{v}(x, y, t)$, is governed by the linearization

$$\mathbf{v}_t = D\,\Delta\mathbf{v} + J\mathbf{v}, \quad D = diag(d_1, d_2),$$

that applying the Fourier transform $\widetilde{\mathbf{v}}(\mathbf{k}, t) = \int_{-\infty}^{\infty} e^{i(k_x x + k_y y)t}\mathbf{v}(x, y, t)dxdy$ becomes the linear ODE system

$$\widetilde{\mathbf{v}}' = \widetilde{J}\widetilde{\mathbf{v}}, \qquad \widetilde{J} = J - \|\mathbf{k}\|_2^2 D, \quad .$$

where $\mathbf{k} = (k_x, k_y)$ and $\|\mathbf{k}\|_2^2 = k_x^2 + k_y^2 = (\pi\nu_x/L_x)^2 + (\pi\nu_y/L_y)^2$ account for the spatial frequencies ν_x, ν_y. Let $\lambda_1(\widetilde{J})$ be the eigenvalue of $\widetilde{J}$ with largest real part; Turing theory implies that if $Re(\lambda_1(\widetilde{J})) > 0$ for some values of $\mathbf{k}$, the perturbations with this spatial frequency will grow and produce spatial patterns, then $\mathbf{w}_e$ is destabilized by diffusion. The Turing conditions on the model parameters identify a range of spatial modes such that pattern formation arises for $\|\mathbf{k}\|_2^2 \in [\mathbf{k}_1^2, \mathbf{k}_2^2]$ (see e.g. [3]). In [62], the authors show that the largest eigenvalue of $H(J)$ must be positive for there to be an eigenvalue of $\widetilde{J}$ with positive real part. Reactivity is therefore a prerequisite for pattern formation via Turing instability. It would be desiderable that numerical methods for approximation of Turing patterns account not only for the asymptotic stability, but also for reactivity features during the initial transient regime.

To sum up, the numerical approximation of Turing pattern solutions is challenging for the following reasons: (i) longtime integration is needed to identify the final pattern as asymptotic solution of the PDE system; (ii) the time solver would account for reactivity at short times; (iii) a large domain $\mathcal{D}$ of integration is required to carefully identify the spatial structures of the Turing pattern and then an accurate spatial discretization with large meshsizes N_x, N_y is needed.
For these reasons, stemming on the appealing computational savings reported in [21] for the test case of the semilinear Heat equation, here we show the matrix-oriented approach to a nonlinear RD-PDEs models. We will apply the well-known ADI method often used in literature (e.g. [80]) and matrix-methods studied below. We will focus our numerical tests highlighting the above points (i) and (ii) for the Schackenberg model, well studied in literature as a prototype with Turing solutions [3]. To deal with the point (iii), we propose to apply these approaches to solve the morpho-chemical DIB model. All these results are reported in [21].

In order to present the results in this Section for simplicity we refer to the R-D equation with zero Neumann boundary condition of the type:

$$\begin{cases} u_t = d\Delta u + f(u), \\ u(x,y,0) = u_0(x,y), \\ (n \cdot \nabla u)|_{\partial\mathcal{D}} = 0 \quad with \quad (x,y) \in \mathcal{D} \subset \mathbb{R}^2, \quad t \in]0,T], \end{cases} \tag{5.8}$$

where Δ is the Laplace operator and n denotes the (typically exterior) normal to the boundary $\partial\mathcal{D}$. Then we show the analogous formulation for the RD system in (5.1).

It is well known that the Method of Lines (MOL) based on classical semi-discretizations in space (e.g. finite differences, finite elements) rewrites (5.8) as an ODE system. For the numerical treatment, we consider a finite difference approximation for spatial derivatives based on the Extended Central Difference Formulas (ECDF$_p$) [1, 80]. These schemes consider the approximation of the Neumann BCs with the same order of schemes used in the interior domain, so that no reduction of order arises near the boundaries. In particular, we apply ECDF of order $p = 2$ as follows. Let us discretize the domain $\mathcal{D} = [0, \ell_x] \times [0, \ell_y]$ with N_x and N_y interior points, giving step sizes $h_x = \ell_x/(N_x + 1)$ and $h_y = \ell_y/(N_y + 1)$.

Let $T_x \in \mathbb{R}^{N_x \times N_x}$ and $T_y \in \mathbb{R}^{N_y \times N_y}$ be the usual tridiagonal matrices corresponding to the approximation of the second order derivatives by central differences (order $p = 2$), along the x and y directions, and zero Neumann BCs approximation. More precisely, $T_x = diag(1, -2, 1) + B$, and similarly for T_y, with corresponding dimensions, where the BCs term (see [80, 77]) is given by

$$B = \frac{2}{3} \begin{bmatrix} 2 & -\frac{1}{2} & \cdots & 0 & 0 \\ 0 & 0 & \cdots & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & & -\frac{1}{2} & 2 \end{bmatrix}. \tag{5.9}$$

Therefore, the semi-discretization of (5.8) in vector form is given by

$$\dot{\mathbf{u}} = A\mathbf{u} + f(\mathbf{u}) \quad \mathbf{u}(0) = \mathbf{u}_0, \tag{5.10}$$

with

$$A = d\widetilde{\Delta} \tag{5.11}$$

and

$$\widetilde{\Delta} = \frac{1}{h_x^2}(I_y \otimes T_x) + \frac{1}{h_y^2}(T_y \otimes I_x) \in \mathbb{R}^{N_x N_y \times N_x N_y} \tag{5.12}$$

where $\otimes$ is the Kronecker operator. Let us observe that at each time $t \in [0, T]$ it is possible to explicitly employ the matrix $U(t) \in \mathbb{R}^{N_x \times N_y}$ containing the same components of $\mathbf{u}(t)$, with $U_{i,j}(t) \approx u(x_i, y_j, t)$, that is, the rows and columns of U explicitly reflect the space grid discretization of the given problem. We shall consider that the vector $\mathbf{u}$ corresponds to the `vec` operation of the matrix U, where each column of U is stuck one after the other. In a finite difference discretization this corresponds to a lexicographic order of the nodes in the rectangular grid. With this notation, for A in (5.11) we recall the property that $A\mathbf{u} = \text{vec}(T_1 U + UT_2)$. Then (5.8) can be written as the following differential matrix equation

$$\begin{cases} \dot{U} = T_1 U + U T_2 + F(U), \\ U(0) = U_0 \end{cases} \tag{5.13}$$

where

$$T_1 = \frac{d}{h_x^2} T_x, \quad T_2 = \frac{d}{h_y^2} T_y^T, \tag{5.14}$$

whit F being nonlinear vector function $f(\mathbf{u})$ evaluated componentwise, and $\text{vec}(U_0) = \mathbf{u}_0$ is the initial condition. This matrix form provides a quite different perspective at the time discretization level than classical approaches, allowing to significantly reduce the memory and computational requirements.

5.3.1 Classical vector methods and their matrix formulation

Vector form For the time stepping of (5.10) we can consider the following methods, where for the sake of simplicity we consider a constant timestep $h_t > 0$ and the time grid $t_n = nh_t$, $n = 0, 1, \ldots, N_t$ so that $(\mathbf{u}_n)_{ij} \approx u(x_i, y_j, t_n)$ in each point (x_i, y_j) of the discretized space:

1. *IMEX methods.*

   i) First order Euler: We discretized in time as $\mathbf{u}_{n+1} - \mathbf{u}_n = h_t(A\mathbf{u}_{n+1} + f(\mathbf{u}_n))$, so that

$$(I - h_t A)\mathbf{u}_{n+1} = \mathbf{u}_n + h_t f(\mathbf{u}_n), \quad n = 0, \ldots, N_t, \tag{5.15}$$

where $\mathbf{u}_0$ is given by the initial condition in (5.8); the linear part is treated implicitly, while the reaction (nonlinear) part f is treated explicitly [70, 2, 30].

ii) Second order SBDF. The widely used IMEX 2-SBDF method [70, 2] applied to (5.10) yields

$$3\mathbf{u}_{n+2}-4\mathbf{u}_{n+1}+\mathbf{u}_n = 2h_tA\mathbf{u}_{n+2}+2h_t(2f(\mathbf{u}_{n+1})-f(\mathbf{u}_n)), \quad n=0,1,\dots \tag{5.16}$$

As usual, $\mathbf{u}_0$ is known, while a step of the first order IMEX-Euler scheme can be used to determine $\mathbf{u}_1$ ([70, 2]).

2. *Exponential integrator.* Exponential first order Euler method [40]:

$$\mathbf{u}_{n+1} = e^{h_tA}\mathbf{u}_n + h_t\varphi_1(h_tA)f(\mathbf{u}_n) \tag{5.17}$$

where e^{h_tA} is the matrix exponential, and $\varphi_1(z) = (e^z-1)/z$ is the first "phi" function [40].

3. *ADI method* [58]. We consider the two-stage explicit time stepping when $\Delta u = u_{xx} + u_{yy}$ is the Laplace operator:

$$\begin{aligned} \frac{u_{ij}^{n+\frac{1}{2}} - u_{ij}^n}{h_t/2} &= (u_{xx})_{ij}^{n+\frac{1}{2}} + (u_{yy})_{ij}^n + f(u_{ij}^n) \\ \frac{u_{ij}^{n+1} - u_{ij}^{n+\frac{1}{2}}}{h_t/2} &= (u_{xx})_{ij}^{n+\frac{1}{2}} + (u_{yy})_{ij}^{n+1} + f(u_{ij}^n), \end{aligned} \tag{5.18}$$

Let $U_n \approx U(t_n) \in \mathbb{R}^{N_x\times N_y}$. After discretization we obtain:

$$\begin{aligned} \left(I - \frac{h_t}{2}T_1\right)U_{n+\frac{1}{2}} &= \left(I + \frac{h_t}{2}T_1\right)U_n + \frac{h_t}{2}F(U_n) \\ U_{n+1}\left(I - \frac{h_t}{2}T_2^T\right) &= U_{n+\frac{1}{2}}\left(I + \frac{h_t}{2}T_2^T\right) + \frac{h_t}{2}F(U_n). \end{aligned} \tag{5.19}$$

We remark that the ADI method naturally treats the approximation in matrix terms, therefore it is the closest to our approach.

Matrix form In this Section we reformulate the time steppings in matrix terms, by exploiting the Kronecker sum in (5.12). We then provide implementation details to make the new algorithms more efficient. We shall see that the matrix-oriented approach leads to the evaluation of matrix functions and to the solution of linear matrix equations with small matrices, instead of the solution of very large

vector linear systems. We stress that the matrix formulation does not affect the convergence and stability properties of the underlying time discretization method [21]. Rather, it exploits the structure of the linear part of the operator to make the computation more affordable. For instance, high accuracy in space is of interest in certain PDE models with Turing solution patterns, which are characterized by the presence of labyrinths, stripes, spots and worms structures. In the following we derive the iteration associated with the single differential equation (5.8). A completely analogous iteration will be obtained for the system (5.1). The matrix-oriented versions of the IMEX methods rely on the Kronecker form of A and on its property, transforming the vector linear system into a matrix linear equation to be solved, of much smaller size. The matrix-oriented formulation of the IMEX-methods is the natural transposition of the vector formulation in matrix form, taking into account the equalities already seen and summarized below:

$$\begin{aligned} \mathbf{u} &= \mathrm{vec}(U) \\ A\mathbf{u} &= \mathrm{vec}(T_1 U + U T_2). \end{aligned}$$

Let us consider the discretized times $t_n = nh_t,\ n = 0, \dots N_t$ with timestep $h_t > 0$.

1. *IMEX methods.*

   i) First order Euler: Adapting the on-step discretization scheme leading to (5.15), to the differential matrix form (5.13), yields

   $$U_{n+1} - U_n = h_t(T_1 U_{n+1} + U_{n+1} T_2) + h_t F(U_n),$$

   which, after reordering, gives the following linear matrix equation, called the *Sylvester* equation,

   $$\begin{aligned} (I - h_t T_1)U_{n+1} + U_{n+1}(-h_t T_2) = U_n + h_t F(U_n), \\ n = 0, \dots, N_t - 1. \end{aligned} \tag{5.20}$$

   Therefore, to obtain the next iterate U_{n+1} the approach requires the solution of a Sylvester equation at each time step, with coefficient matrices $(I - h_t T_1)$, $(-h_t T_2)$ and right-hand side $U_n + h_t F(U_n)$. The numerical solution of this equation is described in Section 5.3.2.

ii) Second order 2SBDF. For the matrix form, given the initial condition U_0, and a further approximation U_1 – obtained for instance by the IMEX Euler method – at each time step t_{n+2} the method determines the following matrix equation

$$3U_{n+2} - 4U_{n+1} + U_n = 2h_t \left(T_1 U_{n+2} + U_{n+2} T_2 + 2F(U_{n+1}) - F(U_n)\right),$$

which, after reordering, leads once again to the solution of a Sylvester equation, this time in the unknown matrix U_{n+2},

$$\left(3I - 2h_t T_1\right) U_{n+2} + U_{n+2} \left(-2h_t T_2\right)$$

$$= 4U_{n+1} - U_n + 2h_t(2F(U_{n+1}) - F(U_n)), \quad n = 0, \ldots, N_t - 2.$$

The coefficient matrices are $3I - 2h_t T_1$, $(-2h_t T_2)$ and the right-hand side is $4U_{n+1} - U_n + 2h_t(2F(U_{n+1}) - F(U_n))$.

2. *Exponential integrator.* A matrix-oriented version of the exponential Euler approach can exploit (5.12) in the computation of both the exponential and the phi-function. In particular, the following property of the exponential matrix is crucial [4]

$$e^{h_t A} = e^{h_t (I \otimes T_1 + T_2^T \otimes I)} = e^{h_t T_2^T} \otimes e^{h_t T_1}.$$

Therefore, for $u = \mathrm{vec}(U)$ we have

$$e^{h_t A} u = \left(e^{h_t T_2^T} \otimes e^{h_t T_1}\right) u = \mathrm{vec}(e^{h_t T_1} U e^{h_t T_2}).$$

Moreover, the operation $v = h_t \varphi_1(h_t A) f = A^{-1}(e^{h_t A} f - f)$ can be performed by means of a two step procedure which, given F such that $f = \mathrm{vec}(F)$ delivers V such that $v = \mathrm{vec}(V)$:

- Compute $G = e^{h_t T_1} F e^{h_t T_2}$
- Solve $T_1 V + V T_2 = G - F$ for V

Therefore, the Exponential Euler method first computes the matrix exponential of multiples of T_1 and T_2 once for all. Then, at each time step the method obtains the approximation U_{n+1} by solving a Sylvester matrix equation. More precisely,

(a) Compute $E_1 = e^{h_t T_1}$, $E_2 = e^{h_t T_2^T}$

(b) For each n

$$\begin{aligned} \text{Solve} \qquad & T_1 V_n + V_n T_2 = E_1 F(U_n) E_2^T - F(U_n) \\ \text{Compute} \qquad & U_{n+1} = E_1 U_n E_2^T + V_n. \end{aligned} \tag{5.21}$$

Several implementation suggestions are given in the next Section. It is important to realize that to be able to solve (5.21) the two matrices T_1 and $-T_2$ must have disjoint spectra. Unfortunately, Neumann boundary conditions imply that both T_1 and T_2 are singular, leading to a zero common eigenvalue. To cope with this problem with employed the following differential matrix equation, mathematically equivalent to (5.13),

$$\dot{U} = (T_1 - \sigma I)U + UT_2 + (F(U) + \sigma U), \tag{5.22}$$

with $\sigma \in \mathbb{R}$, $\sigma \neq 0$, opportunely chosen as explained in the next Section. With this simple "relaxation" procedure the matrix $T_1 - \sigma I$ is no longer singular, and has no common eigenvalues with $-T_2$, at the small price of including an extra linear term to the nonlinear part of the equation. We note that adding and subtracting the term σU to the ODE may be beneficial – though not strictly necessary – also for the other methods; thus in [21] we have included the stability analysis for all considered time integration strategies based on the relaxed matrix equation (5.22).

The matrix equation above should be compared with the vector form, requiring the solution of a linear system of size $N_x N_y \times N_x N_y$ at each time step. It is important to realize that for a two-dimensional problem on a rectangular grid, the number of nodes required in each direction need not exceed a thousand, even in the case a fine grid is desired to capture possibly pathological behaviors. Hence, while the Sylvester equation above deals with, say, matrices of size 500×500, the vector form deals with matrices and working vectors of size $250\,000 \times 250\,000$. Arguably, these latter large matrices are very sparse and structured, so that strategies for sparse matrices can be exploited; nonetheless, the Sylvester equation framework allows one to employ explicit factorizations, also exploiting the fact that the matrices do not change with the time steps. Algorithmic details will be given in the following Section 5.3.2.

5.3.2 Implementation details

Whenever the matrix sizes are not too large, say up to a thousand, the previously described matrix methods can be made more efficient by computing a-priori a spectral decomposition of the coefficient matrices involving T_1 and T_2. In the following we shall assume that the two matrices are diagonalizable, so that their eigenvalue decompositions can be determined. Let them be $T_k = X_k \Lambda_k X_k^{-1}$, $k = 1, 2$, with X_k nonsingular and $\Lambda_k = \mathrm{diag}(\lambda_1^{(k)}, \lambda_2^{(k)}, \ldots)$ diagonal. Let us first consider the IMEX Euler iteration in (5.20). Compute the $N_x \times N_y$ matrix $L_{i,j} = 1/((1 - h_t\lambda_i^{(1)}) + (-h_t\lambda_j^{(2)}))$. Hence, at each iteration n we can proceed as follows

1. Compute $\widehat{U}_n = X_1^{-1} Q(U_n) X_2$;
2. Compute $U_{n+1} = X_1 (L \circ \widehat{U}_n) X_2^{-1}$

where $Q(U_n) = U_n + h_t F(U_n)$ and $\circ$ is the Hadamard (element by element) product. The second step performs the solution of the Sylvester equation by determining the solution entries one at the time, in the eigenvector bases, and then the result is projected back onto the original space to get U_{n+1} [82]. Proceeding in the same manner, the corresponding version for IMEX-2SBDF can be derived. Letting this time $L_{i,j} = 1/((3 - 2h_t\lambda_i^{(1)}) + (-2h_t\lambda_j^{(2)}))$ and at each iteration n we have:

1. Compute $\widehat{U}_n = X_1^{-1} Q(U_n, U_{n+1}) X_2$;
2. Compute $U_{n+1} = X_1 (L \circ \widehat{U}_n) X_2^{-1}$.

where $Q(U_n, U_{n+1}) = 4U_{n+1} - U_n + 2h_t(2F(U_{n+1}) - F(U_n))$. In the following numerical experiments we will call these methods: **reduced IMEX-Euler (rEuler)** and **reduced 2SBDF (rSBDF)**. Whenever the PDE problem is linear, that is $f(u) = \alpha u + \beta$, the computation further simplifies, since all time steps can be performed in the eigenvector basis, and only at the final time of integration the approximate solution is interpolated back to the physical basis.

In a similar way, the exponential Euler integrator described in Section can be rewritten as

1. Compute $\widehat{e}_i = \mathrm{diag}(e^{h_t\lambda_1^{(i)}}, e^{ht\lambda_2^{(i)}}, \ldots)$, $i = 1, 2$; $\widehat{E} = \widehat{e}_1 \widehat{e}_2^*$ and $\widehat{L}_{i,j} = (h_t\lambda_i^{(1)} + h_t\lambda_j^{(2)})^{-1}$, with $\widehat{E}, \widehat{L} \in \mathbb{C}^{N_x \times N_y}$.

2. For each n,

Compute $\widehat{F}_n = X_1^{-1} F(U_n) X_2$ (Project $F(U_n)$ on the eigenbases)

Compute $G = \widehat{E} \circ \widehat{F}_n - \widehat{F}_n$ (Apply `exp` and form the Sylvester eqn rhs)

Compute $V = \widehat{L} \circ G$ (Solve the Sylvester eqn)

Compute $U_{n+1} = X_1(\widehat{E} \circ (X_1^{-1} U_n X_2) + V) X_2^{-1}$ (Compute the next iterate)

In the following numerical experiments we will call this method **reduced Exp (rExp)**. It is worth noting that, if the "relaxation" approach corresponding to (5.22) is considered, all above procedures in points (1)-(2) can be extended simply by considering $F_\sigma(U) = F(U) + \sigma U$ and the spectral decomposition of $T_1(\sigma) = T_1 - \sigma I$, for a fixed value of σ.

5.3.3 RD-PDE systems: matrix approach

In this Section, we present the application of the matrix approach to the reaction-diffusion model with non-linear reaction-terms and zero Neumann boundary condtions, given in (5.1). By using the Method of Lines for the space discretization, the matrix formulation of (5.1) yields a system of ODE matrix equations as follows:

$$\begin{cases} U' = d_1(T_1 U + U T_2) + F_1(U, V) \\ V' = d_2(T_1 V + V T_2) + F_2(U, V) \\ U(0) = U_0, V(0) = V_0. \end{cases} \tag{5.23}$$

The matrix form of the classical ODE methods can be derived for (5.23), such that at each timestep t_n, the solution of the following Sylvester matrix equations is required:

$$\begin{cases} S_1 U_{n+1} + U_{n+1} S_2 = Q_1^n, \\ R_1 V_{n+1} + V_{n+1} R_2 = Q_2^n, \quad n = 0, \ldots, N_t - 1 \qquad U_0, V_0 \text{ given} \end{cases} \tag{5.24}$$

where $Q_j^n = Q_j^n(U_n, V_n), j = 1, 2$ in the case of a one step method (like IMEX Euler method in the previous Sections) and $Q_j^n = Q_j^n(U_{n-1}, V_{n-1}, U_n, V_n)$ (U_0, U_1

given) for a two-step scheme. Recalling the procedure of Section 5.3.1, for IMEX-Euler we have

$$
\begin{aligned}
&S_1 = I - h_t d_1 T_1,\ S_2 = -h_t d_1 T_2,\\
&R_1 = I - h_t d_2 T_1,\ R_2 = -h_t d_2 T_2,\\
&Q_1^n = U_n + h_t F_1(U_n, V_n), \quad Q_2^n = V_n + h_t F_2(U_n, V_n),
\end{aligned}
$$

while for IMEX-2SBDF we have

$$
\begin{aligned}
&S_1 = 3I - 2h_t d_1 T_1,\ S_2 = -2h_t d_1 T_2,\\
&R_1 = 3I - 2h_t d_2 T_1,\ R_2 = -2h_t d_2 T_2,\\
&Q_1^n = 4U_n - U_{n-1} + 2h_t(F_1(U_n, V_n) - F_1(U_{n-1}, V_{n-1})),\\
&Q_2^n = 4V_n - V_{n-1} + 2h_t(F_2(U_n, V_n) - F_2(U_{n-1}, V_{n-1})).
\end{aligned}
$$

Also the matrix-oriented version of the exponential approach can be derived for the RD systems. In particular, letting once again $E_{1,1} = e^{h_t d_1 T_1}$, $E_{1,2} = e^{h_t d_1 T_2^T}$, and $E_{2,1} = e^{h_t d_2 T_1}$, $E_{2,2} = e^{h_t d_2 T_2^T}$ we obtain

$$
\begin{aligned}
U_{n+1} = E_{11} U_n E_{12}^T + Y_n, \quad &\text{where} \quad (d_1 T_1 - \sigma I) Y_n + Y_n (d_1 T_2) = \\
&\qquad E_{11} \widetilde{F}_1(U_n, V_n) E_{12}^T \\
V_{n+1} = E_{21} V_n E_{22}^T + Z_n, \quad &\text{where} \quad (d_2 T_1 - \sigma I) Z_n + Z_n (d_2 T_2) = \\
&\qquad E_{21} \widetilde{F}_2(U_n, V_n) E_{22}^T,
\end{aligned}
\tag{5.25}
$$

where σ is as described in Section 5.3.2, while $\widetilde{F}_1(U_n, V_n) = F_1(U_n, V_n) + \sigma U_n$ and $\widetilde{F}_2(U_n, V_n) = F_2(U_n, V_n) + \sigma V_n$. In particular, the approach requires the solution of two Sylvester equations per step, which is the same cost as for the IMEX procedure, together with matrix-matrix multiplications with the exponentials. As already discussed for the single equation case, these costs can be significantly reduced by working in the eigenvector basis of T_1 and T_2.

Schnakenberg model The RD-PDEs for the Schnakenberg non-dimensional model are given by

$$
\begin{cases}
u_t = \Delta u + \gamma(a - u + u^2 v), \quad (x, y) \in \mathcal{D} \subset \mathbb{R}^2,\ t \in]0, T_f] \\
v_t = d\Delta v + \gamma(b - u^2 v), \\
(n \nabla u)_{|\partial\mathcal{D}} = (n \nabla v)_{|\partial\mathcal{D}} = 0 \\
u(x, y, 0) = u_0(x, y), v(x, y, 0) = v_0(x, y)
\end{cases}
\tag{5.26}
$$

As a great number of recent papers show (see e.g. [50, 68]), this model receives great attention because it has a very simple nonlinear structure and its patterns are qualitatively similar to classical ones found in biological experiments. The parameters model a, b, d, γ are positive constants and a unique stable equilibrium exists which undergoes the Turing instability, given by $u_e = a + b$, $v_e = \frac{b}{(a+b)^2}$. We consider the literature choice:

$$\mathcal{D} = [0,1] \times [0,1], \quad d = 10, \gamma = 1000, a = 0.1, b = 0.9$$

yielding a cos-like spotty pattern $\approx \cos(\nu_x \pi x)\cos(\nu_y \pi y)$ with the selected modes $(\nu_x, \nu_y) = (3,5), (5,3)$ [3]. (see Figure 5.1). We consider the initial conditions $u_0(x,y) = u_e + 10^{-5} rand(x,y)$, $v_0(x,y) = v_e + 10^{-5} rand(x,y)$ where *rand* is the default MATLAB [56] function, where we fix the seed of the generator (rng('default')) at starting of each simulation. To study the time dynamics in our simulations we will calculate the space mean value

$$\langle U_n \rangle = mean(U_n) \approx \langle u(t_n) \rangle = \frac{1}{|\mathcal{D}|} \int_{\mathcal{D}} u(x,y,t_n)\, dx\, dy \qquad t_n = n\, h_t,\ n = 0, \ldots, N_t \tag{5.27}$$

that if a stationary pattern is attained would tend to a constant value for $t \to T_f = N_t\, h_t$. We report also the behaviour of the increment $\delta_n = \|U_{n+1} - U_n\|_F$ (Frobenius norm) that will tend to zero (a certain tolerance) if the steady state is reached. These two indicators will be useful also to describe the numerical behaviors of the methods in the initial transient and then to study their reactivity features. We present two tests as follows when $T_f = 2$, $N_y = N_x$.

Test (a). Let us fix $N_x = 100$ and variable h_t = 0.5e-4, 1e-4, 2e-4, 3e-4. In the simulations reported in Figure 5.2, it is possible to see that all methods share the same qualitative behavior and we can distinguish two time regimes $I_1 = [0, \tau]$ and $I_2 =]\tau, T_f]$. In I_1 the reactivity holds: the solution oscillating departs from the spatially homogeneous pattern due to the superimposed (small random) perturbations and becomes unstable, in I_2 the solution starts to stabilize towards the steady Turing pattern. Numerically the value of τ can be approximated by τ_n the time of the maximum of the increment δ_n. Let us discuss in more details the characteristics of the different methods.

I_1-reactivity zone: Figure 5.2, top subfigure, for the increments δ_n shows that

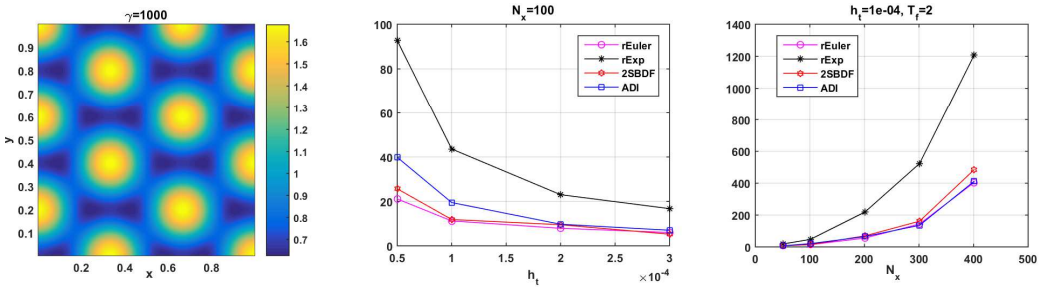

Figure 5.1: Schnackenberg model. Left plot: Turing pattern solution for $\gamma = 1000$ ($N_x = 400$). Center plot: CPU times (sec) for Test (a), $N_x = 100$ variation of h_t. Right plot: CPU times (sec) for Test (b), $h_t = 10^{-4}$, increasing values of $N_x = 50, 100, 200, 300, 400$.

there exists an initial phase of oscillations which size depends on the method and for $h_t \to 0$ this period tends to a certain time value τ_0. Then for $\tau_0 \leq t_n \leq \tau$ the solution *must* be unstable as the necessary condition for Turing instability requires. Looking at Figure 5.2, bottom subfigure, for $< U_n >$ the passage between I_1 and I_2 is related to the steep part of the curve $\langle U(t) \rangle$ that connects the very short-term and the final states of the system and the value of τ can be related to the inflection point of this curve. In our numerical experiments it seems that both $\tau_0 = \tau_0(h_t^p)$ and $\tau = \tau(h_t^p)$. In fact, as also the zoom insets show, for $h_t \to 0$ rEuler and ADI have the same behavior, rExp has curves $< U_n >$ with different slopes depending from h_t, rSBDF identifies the best approximation of τ_0, τ also for larger value of h_t (as expected because it is a 2nd order method). In [52, 53] similar studies were done, with also different numerical methods including the fractional step θ-methods [35].

I_2-stabilizing zone: for $h_t < h_t^{cr}$, fixed N_x, for all methods the asymptotic pattern is reached. Here for $h_t^{cr} \simeq$ 3e-4 rEuler and ADI do not attain any pattern (the oscillations of $< U_n >$ are shown only in the zoom insets of Figure 5.2), while rExp attains the final pattern after a fully oscillating transient behaviour (see the (red) oscillations in upper (top) subplots in Figure 5.2). We could say this h_t is a critical value for "reactive stability " of rExp. In Figure 5.1, central plot, we report the computational cost for all methods: note that we are solving here an increasing number of Sylvester matrix equations in (5.23) of the same dimension. rEuler and rSBDF have almost the same cost and are cheaper than the other

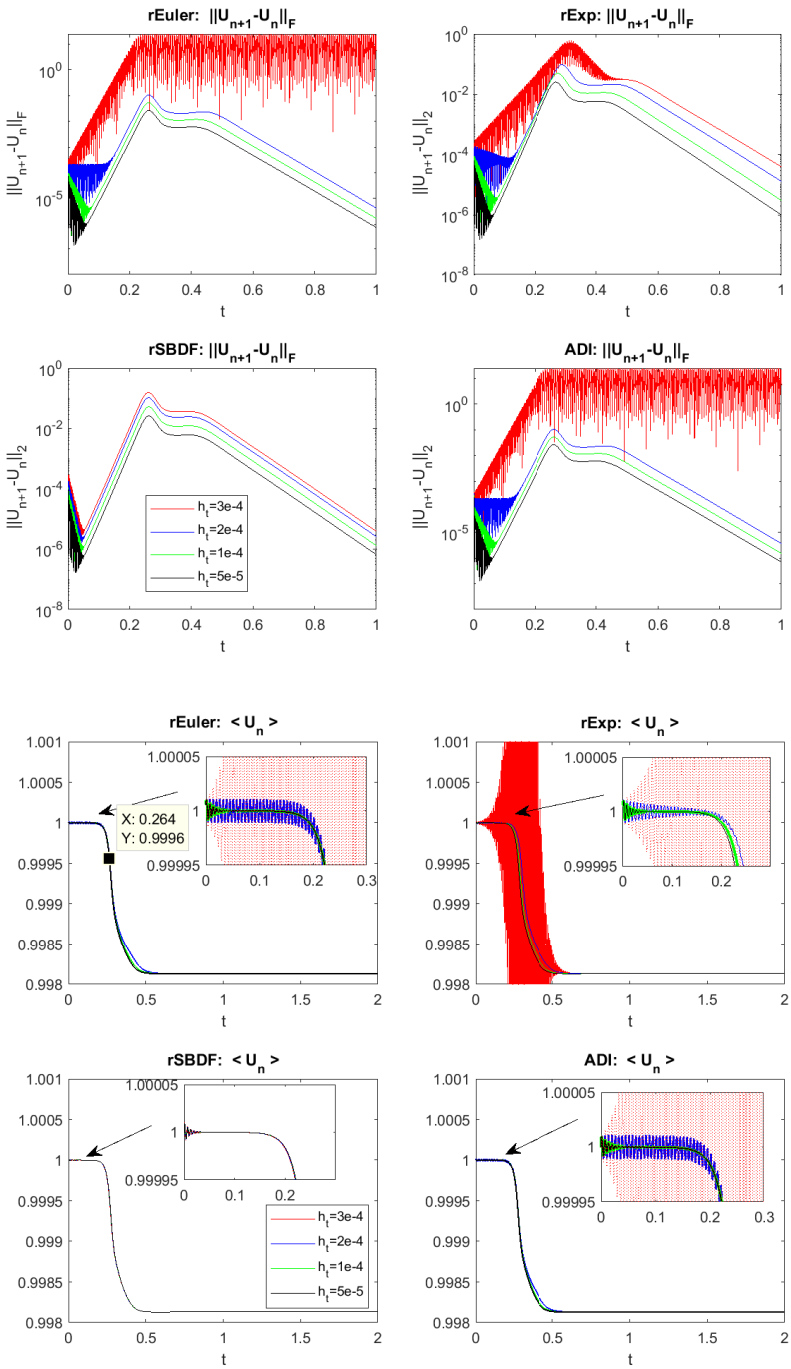

Figure 5.2: Schnackenberg model- Test (a). For all methods with $N_x = 100$ and varying $h_t = 0.5$e-4, 1e-4, 2e-4,3e-4 we show the time behaviours of the increments $\delta_n = \|U_{n+1} - U_n\|_F$ (top subfigure) and of the space mean values $< U_n >$ (bottom subfigure).

methods. rExp is more expensive than ADI. It is worth noting that the IMEX schemes in vector form are typically more expensive than ADI method (see e.g. [77]).

Test (b): we fix $h_t =$ 1e-4 and let vary $N_x = 50, 100, 200, 400$, such that the matrix methods solve the same number of Sylvester equations of increasing sizes. All methods have the same time dynamics in I_1 and I_2, then we show in Figure 5.3 the increments and the mean values found by rSBDF. In Figure 5.3, right plot, we note that: the final value of $< U_n >$, say $\sigma = \sigma(N_x)$, changes with N_x as expected. Moreover, looking at the left plot, $\tau = \tau(h_t, N_x)$ seems to be an increasing function of N_x. In some sense, this can be expected because we are indeed solving discrete problems with different initial conditions (that include different, even if small, random perturbations). This sensitivity wrt to the choice of the initial conditions is well known in literature about pattern formation (see e.g. [54]). This is the main reason why we do not propose to apply a low rank approximation method for (5.23), like the KPIK proposed in [57] to solve matrix differential equations of Lyapunov/Riccati type. In fact, it can be shown that projecting the Turing solution on a low-rank manifold especially during the transient time dynamics can induce the selection of specific Fourier modes in the final pattern. This topic will be object of future investigations. In the right plot of Figure 5.1 we report the computational costs of all methods. We recall that by varying N_x Sylvester matrix equations in (5.23) of increasing size are solved by the reduced spectral approach. rEuler, ADI and rSBDF have almost the same cost and are cheaper than rExp. For largest spatial dimensions rEuler becomes the more economic. It is worth noting, that for the larger values of N_x this test could not be performed by classical vector-oriented version of the same schemes due to the prohibitive computational load.

DIB model In this Section, we show the importance of the matrix-oriented approach to carefully approximate the spatial structure of Turing patterns on fine meshgrids and large domains with reasonable computational cost and accessible amount of required memory, otherwise not attainable by the classical vector-oriented approach. We consider the RD-PDE model studied in [47] describing an electrodeposition process for metal growth, where the kinetics in (5.1) are given

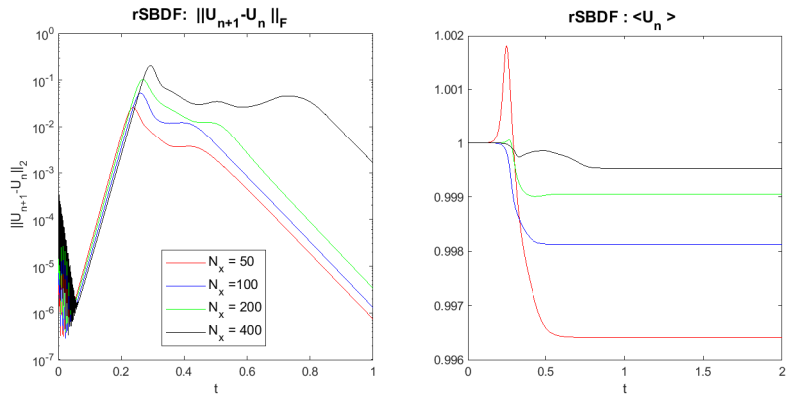

Figure 5.3: Schnackenberg model-Test (b). For h_t = 1e-4 and increasing values of $N_x = 50, 100, 200, 400$ we show the time behaviour of the increment $\delta_n = \|U_{n+1} - U_n\|_F$ (left plot) and of the space mean value $< U_n >$ (right plot). These results are those of the rSBDF method, all other schemes exhibit essentially the same trends (see comments in the main text).

by

$$\begin{aligned} f_1(u,v) &= \rho\left(A_1(1-v)u - A_2\,u^3 - B(v-\alpha)\right), \\ f_2(u,v) &= \rho\left(C(1+k_2u)(1-v)[1-\gamma(1-v)] - Dv(1+k_3u)(1+\gamma v)\right). \end{aligned} \tag{5.28}$$

$u(x,y,t)$ represents the morphology of the metal deposit, $v(x,y,t)$ its surface percentual chemical composition, the nonlinear source terms account for generation and loss of relevant material during the process. In particular in [46], this model has been proposed to study pattern formation during the charge-discharge process of batteries. In [46] it has been also proved that for a given parameter choice of the RD-PDE model there exists an *intrinsic pattern type* that only can emerge if an *effective domain size* of integration is considered given by $\mathcal{A} = \rho|\mathcal{D}|$, where $|\mathcal{D}|$ = area($\mathcal{D}$). Hence, if the scaling factor in (5.28) is $\rho = 1$, a large domain $\mathcal{D}$ must be chosen to "see" the Turing pattern. For this reason, the number of meshpoints N_x, N_y that is the size of the Sylvester equations (5.23) must be sufficiently large. In Figure 5.4 we report two typical situations: the left plot refers to a too small domain to be able to identify the morphological class, which is instead clearly visible in the right plot, determined with a much larger domain. Note that the fine grid allows us to clearly recognize the spot-worms pattern in its full granularity that is not otherwise obtained on the rough grid with $N_x = 50$, as shown in the lower plot. These solutions have been obtained by solving (5.1)-(5.28) on

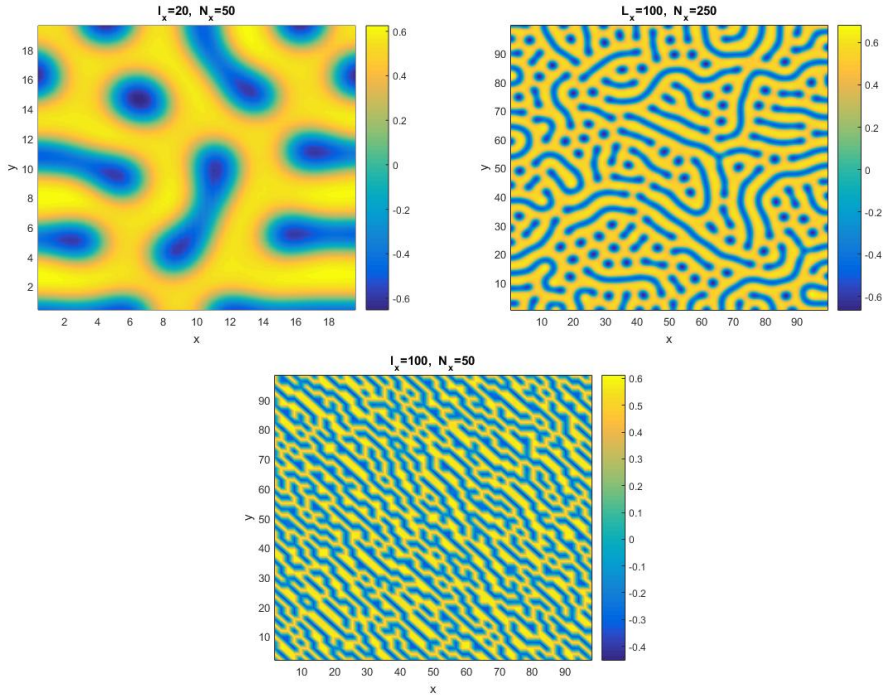

Figure 5.4: Spot-worms Turing pattern of DIB Model. Left: $\Omega = [0, 20] \times [0, 20]$ and $N_x = 50 (h_x = 0.4)$. Right: $\Omega = [0, 100] \times [0, 100]$ and $N_x = 250 (h_x = 0.4)$. Below: $\Omega = [0, 100] \times [0, 100]$ and $N_x = 50 (h_x = 2)$

a square domain $\Omega = [0, \ell_x] \times [0, \ell_x]$ and with the following parameter choice for which a spot-worms pattern is expected ([47]): $d_1 = 1, d_2 = d = 20, \rho = 1, A_1 = 10; A_2 = 5; k_2 = 2.5; k_3 = 1.5; \alpha = 0.5; \gamma = 0.2; D = 2.4545, B = 28, C = 8$.

We apply again ADI and the matrix methods rEuler, rExp, rSBDF until $T_f = 100$, with $h_t =$ 1e-2. In Figure 5.5 we show the dynamics of the increment $\|U_{n+1} - U_n\|_F$ and of the mean value $< U_n >$ for the simulations corresponding to the full spot and worms in Figure 5.4 (upper right). For the chosen h_t the methods exhibit different reactivity and stabilizing properties. The rSBDF method seems to display the best performance.

In the Table 5.1 we reports the computational time of all numerical methods for obtaining the patterns in Figure 5.4, that is for the two cases (i) $N_x = 50, \ell_x = 20$, (left) and (ii) $N_x = 250, \ell_x = 100$ (right), including that of the vector formulation (LU with pivoting) only for the IMEX Euler method. Note that the cost in the case $N_x = 50, \ell_x = 100$ (Figure 5.4- lower) is the same as for the case (i), for this reason it is not reported.

As it can observed, for $N_x = 50$, that is when the pattern is not well identified,

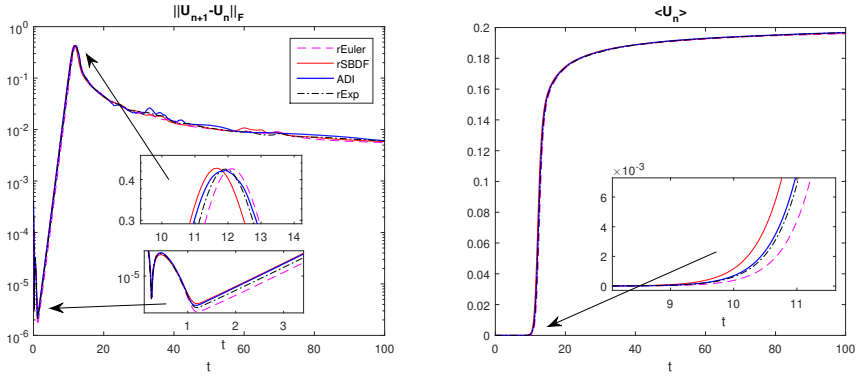

Figure 5.5: Spot-worms Turing pattern of DIB Model. Time dynamics of the increment $\|U_{n+1} - U_n\|_F$ (left plot) and of the mean value $< U_n >$ (right plot) for all methods in the case $\Omega = [0, 100] \times [0, 100]$, $T_f = 100$, $h_t = 10e - 2$, $N_x = 250$.

Methods	$\ell_x = 20, N_x = 50$	$\ell_x = 100, N_x = 250$
IMEX Euler (vector form)	5.6 s	191.9 s
rEuler	3.2 s	66.2 s
rSBDF	5.5 s	98.3 s
rExp	5.6 s	121.2 s
ADI	4.2 s	69.2 s

Table 5.1: DIB model: computational time for all methods to obtain the patterns in Figure 5.4 $N_x = 50, \ell_x = 20$ (left plot), $N_x = 250, \ell_x = 100$(right plot).

all methods display similar computational performances. In the other case, the vector form significantly suffers from dealing with much larger dimensional data, with respect to the matrix-oriented schemes. rEuler exhibits the best computational times for both dimensions. The other matrix-based schemes are almost equivalent, with ADI being slightly less expensive. These preliminary experiments seem to indicate that the matrix formulation is a competitive methodology for the numerical solution of the RD-PDE systems when a fine spatial grid is necessary to capture the morphological features of the pattern.

5.4 Summary

In the previous Section we have shown that the classical semi-discretization in space of reaction-diffusion PDEs can be seen as a system of matrix ODE equa-

tions. By using the matrix formulation, the time dicretization by an ODE solver yields a sequence of Sylvester matrix equations to be solved at each iteration in time. Therefore we have considered the well-known IMEX schemes (Euler and 2SBDF) and the Exponential Euler method and we have solved the corresponding Sylvester equations in the spectral space, by defing the corresponding *reduced* schemes. Comparing the *reduced* schemes with the classical vector approach allow us to deal with significantly smaller discrete problems and to reduce the computational costs. We have shown that both features are important for the numerical approximation of Turing pattern solutions. For the Schnackenberg model, we have solved the two complementary cases when (Test a) an increasing number N_t of Sylvester equations of the same size N_x are solved and when (Test b) we fix the same number N_t of problems of Sylvester equations of increasing size N_x. We found that rEuler is the more economic solver, but rSBDF is a good compromise between accuracy (second order) and cost. rSBDF is also the best scheme to track carefully the reactivity phase of Turing systems at short times. For the DIB-morphochemical model we have shown that a matrix-oriented approach is mostly important when a large finely discretized space domain $\otimes$ is required to well identify the spatial structures of the Turing pattern. In the literature, to avoid the huge computational load of the vector-IMEX methods, this challenge has been often faced by the using ADI approach, but our comparisons show that the new *reduced* schemes can be a valid alternative to it.

Chapter 6

Application: PDE-PIP for a morphochemical model

In this Chapter we show a real application of the PDE-PIP by estimating the parameters for the DIB model for experimental data, studied for the first time in [10, 47]. We describe the differential model and illustrate the types of solutions expected for different values of the parameters as reported in [81]; then we formulate the identification problem (DIB-PIP) and show two different approach: the classical least square minimization realized in [81] and the first results in the extension of the Fourier approach to the PDE case.

6.1 DIB model: description and formulation of DIB-PIP

The morphochemical model for electrodeposition (DIB) was studied for the first time in [10, 47] and it is given by (5.1), where the coupling equations involves two variables with the following meaning: the adimensional $u(x, y, t)$ that describes the morphology, and $0 \leq v(x, y, t) \leq 1$ which account for the surface chemistry and expresses the instantaneous increment of the electrodeposit profile; $d_1 = 1$, $d_2 = d := \frac{d_1}{d_2}$ is the ratio of the diffusion coefficients and the nonlinear source terms are given by (5.28), and account for the generation (deposition) and loss (corrosion) of relevant material. The model is accompanied by zero Nuemann boundary conditions and the following initial condition: $u(x, y, t_0) = u_0(x, y)$, $v(x, y, t_0) = v_0(x, y)$ $(x, y) \in \mathcal{D}$. The (C, B)-bifurcation diagram associated to

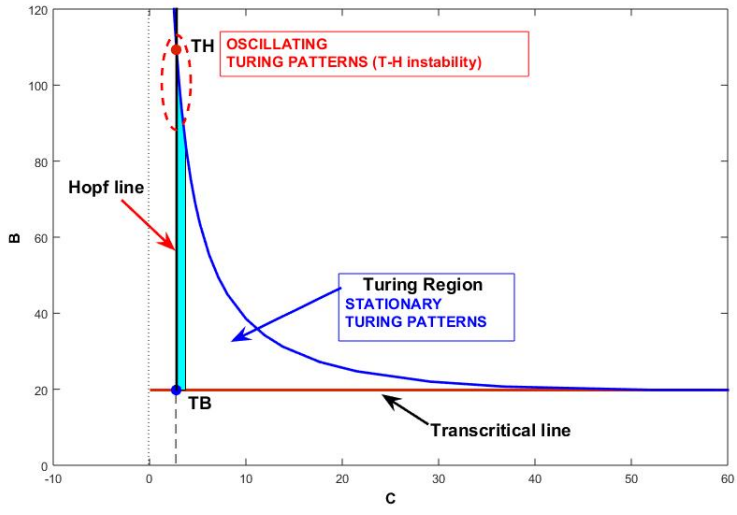

Figure 6.1: Bifurcation diagram in the parameter space (C, B) for the diffusion coefficient $d = 20$. The values for the other parameters are fixed as follows: $\pm\alpha = 0.5, \gamma = 0.2, k_2 = 2.5, k_3 = 1.5, A_1 = 10$. For this choice of the parameters the two bifurcation points TH and TB have coordinates $(C_{TH}, B_{TH}) = (2.8061, 109.13)$ and $(C_{TB}, B_{TB}) = (2.8061, 19.7979)$.

the RD model is reported in Figure 6.1, bounded by the following inequalities [47]:

$$C_H < C < dC_H, \quad B_{tr} < B < B_T \tag{6.1}$$

where

$$\begin{aligned} B_{tr} &= \frac{A_1(1-\alpha)F_2(\alpha,\gamma)}{(k_2-k_3)F_1(\alpha,\gamma)}, \\ C_H &= \frac{A_1(1-\alpha)}{F_2(\alpha,\gamma)}, \\ B_T &= \frac{d^2A_1^2(1-\alpha)^2 + CF_2(\alpha,\gamma)[2A_1d(1-\alpha) + CF_2(\alpha,\gamma)]}{4dC(k_2-k_3)F_1(\alpha,\gamma)}, \end{aligned} \tag{6.2}$$

with

$$\begin{aligned} F_1(\alpha,\gamma) &= (1-\alpha)(1-\gamma+\alpha\gamma), \\ F_2(\alpha,\gamma) &= \frac{2\alpha\gamma(1+\alpha\gamma-\gamma)+1-\gamma}{\alpha(1+\alpha\gamma)}. \end{aligned}$$

Let us observe that the Turing region $\mathcal{R}$, shown in Figure 6.1, does not depend on the values of A_2 and ρ.

Information collected from simulations with parameters in $\mathcal{R}$ reported in [81] show interesting features, that we synthesize below. In Figure 6.2 a segmentation of the Turing region in significant parts is shown. Six sub-regions could be

identified, named $R_0, R_1, ..., R_5$, that are highlighted in Figure 6.2 with different colours. The authors in [81] considered a selection of parameter values (C, B) in each sub-region to describe the different kinds of patterns present and reported these choices by different symbols associating a small letter from a to m to each of them. Each letter identifies the corresponding (stationary) pattern shown on the right. Hence, by following the description reported in [81], for decreasing values of B the R_i, $i = 0, ..., 5$, the sub-regions are given by:

- R_0: is all the zone above the Turing boundary. Here the solutions tend to the homogeneous equilibrium equal to v_e and Turing instability disappears.

- R_1 mixed-spots-stripes: is the interior zone of the Turing region near its boundary. Here, the stationary solutions are mixed spot-stripe patterns. There is a predominance of spots near the boundary (see pattern a) and more stripes far from the boundary (see patterns b,c).

- R_2 labyrinths: this zone is full of solutions similar to labyrinth. Patterns d,e,f show labyrinths with different arrangements of their arms that tend to be longer and better aligned for increasing values of C.

- R_3 reversed spots and worms: this zone is between the labyrinth and holes regions. This is like a transition zone where for decreasing values of B the labyrinths are flattened and the arms are fragmented in reversed spots and worms that are in fact holes of these particular shapes. The worms become longer and predominant for increasing values of C (patterns g, h, i).

- R_4 reversed spots/holes: the simulated pattern is reported in the small picture on the left of the Hopf line (bottom). These spots are indeed holes on a flat surface. The number of holes increases for increasing values of C (patterns j, k)

- R_5: this region is just above the transcritical line. Here even if inside the Turing region, the stationary solution is not a Turing pattern, but the destabilization of the equilibrium v_e leads to another spatially homogeneous equilibrium. For (C, B) values on the boundary shared with the R_4 zone, we show the pattern m corresponding to an almost flat surface with few spots entering 'from' the border.

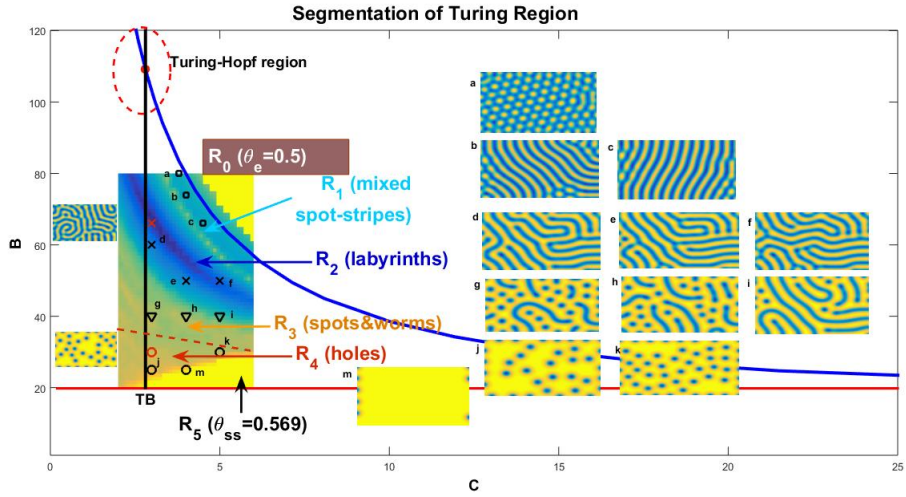

Figure 6.2: Segmentation of the Turing region: six subregions R_0, R_1, ..., R_5 from top to bottom are identified in the bifurcation diagram of Figure 6.1. In each subregion we report a selection of parameter pairs (C, B), indicated by a symbol and a letter. Each letter from a to m identifies the corresponding (stationary) pattern shown on the right. Taken from [81]

As we have seen, in the Turing zone there are sub-regions where parameters can describe several types of structured data. Furthermore, these sub-regions are found to be contiguous and topologically simply connected. In [81], the authors pointed out that: *(1) the segmentation has been obtained for* $A_2 = 1$ *and* $N_x = N_y = 70$ *meshpoints in all simulation snapshots; (2) each pattern snapshot has been obtained for a final time* T *such that the steady state of the PDE model has been reached; (3) for increasing values of* A_2 *the scenario in the lower part of the Turing region changes, that is regions* R_3, R_4, R_5 *disappear and the* R_2 *region of labyrinths of different shapes englobes them.* A theoretical analysis on the role of the parameter A_2 deserves further studies; a qualitative discussion on this point was already given in [47].

To define our PIP-DIB, we assume that all the parameters of the model are fixed except C and B, which are more meaningful parameters from the electrochemical point of view [47]. Given an experimental map, we wish to associate its a couple $(C, B) \in \mathcal{R}$, which is located in the sub-region that best identifies his structure. Then, the DIB-PIP can be expressed as follows: find $(C^*, B^*) \in \Omega$

such that

$$J(C^*, B^*) = \min_{(C,B)} J(C, B) \tag{6.3}$$

where Ω is a fixed subset of the Turing region $\mathcal{R}$, which contains the simulation with the same 'type' of spatial structure of the given map.

6.2 DIB-PIP: numerical results

In this Section we report two example of PDE-PIP with simulated and experimental data reported in [81], in order to present the first parameters identification for the DIB model and the optimization procedure introduced in [81]. For this purpose the authors fixed all the parameters except C and B, as described above, and they referred to the Turing region $\mathcal{R}$ shown in Figure 6.1. They chose the values of the parameter A_2 and of the final time T_f according to the typology of data, as reported in the discussion below and the scaling factor $\rho = 1$. In each case the data map $\widetilde{\mathbf{M}}$ was normalized, because they are interested in the shape of the experimental pattern independently from its numerical values, so the comparisons was made with the model solution $v(x, y, T_f)$ normalized between 0 and 1 which, with a slight abuse of notation, we continue to refer to as v, obtained by using the ADI method as PDEs solver.

Simulated data Let us consider a simulated pattern obtained in [81] for $A_2 = 1$, $(C, B) = (3, 66)$ and $T_f = 20$ of labyrinth-type, and the domain of integration fixed to be $\Omega = [0, 50] \times [0, 36]$. To show the convergence properties of the optimization algorithm, the authors solved the PIP(i) and PIP(ii) steps, defined in Section 5.2, in the case of this simulated labyrinth with noisy data. The synthetic observation was generated by added a random Gaussian noise shown in Figure 6.3 on the top left, where the standard deviation of the noise is taken to be 0.1, representing 10% of the maximum value of the field. The corresponding cost function (5.4), with $W = I$, in shown in Figure 6.3 on the top right. By the inspection of the cost they were able to find a sub-region of the parameter space, around the exact minimum, yielding labyrinth-like structures. The cost function for the noisy data shown has a very similar pattern to the perfect observation case, with one zone of low cost function values. They chose as initial guess $(C_0, B_0) = (2.9, 80)$,

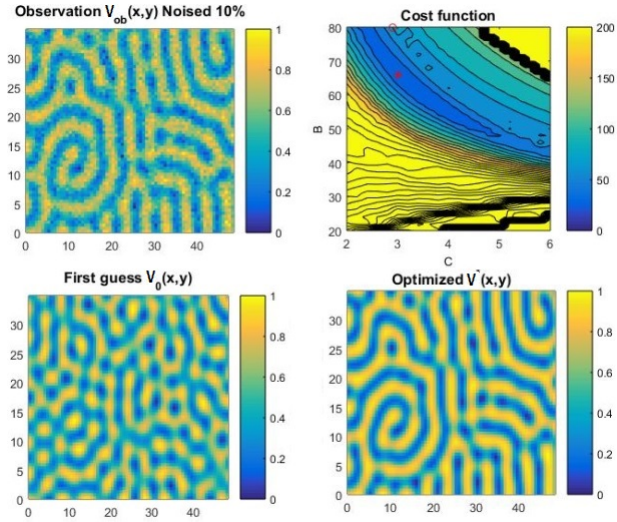

Figure 6.3: PDE-PIP-DIB: On the top: on the left the simulated labyrinth for $(C, B) = (3, 66)$ normalized patter with 10% noise; on the right the cost function. On the bottom the fist guess for the optimization and the result pattern of the minimization. Taken from [81].

a point far from the minimum value; we show the corresponding numerical solution of the model used as the first guess pattern in Figure 6.3 on the bottom left. The optimized pattern attained by the descent algorithm is shown in Figure 6.3 on the bottom right, and the relative error on the parameters are given by $rel_{errB} = |B^* - B|/|B| = 0.008$ and $rel_{errC} = |C^* - C|/|C| = 0.0175$. Since the zone of minimum values is well defined even with the noisy data, the authors deduced that the algorithm is able to find a solution that is much better than the initial starting point, in terms of both error norm and pattern produced.

Experimental data Let us consider an experimental map in Figure 6.4 on the top left reported in [81] and reprinted with permission from [45], whose structure is classified as mixed-spot-stripes. By fixing $A_2 = 30$ and $T_f = 20$, the authors constructed the cost function (5.4) in the region $\mathcal{R} = [2, 10] \times [20, 80]$ of the parameter space, shown in Figure 6.4 on the top right. We note that there are now two distinct and clearly defined zones of low cost-function values. Here it is evident that there exists a minimum near $(C_0, B_0) = (5, 20)$ far from the Turing curved boundary but in the lower part of the bifurcation diagram. The corresponding pattern $v^*(x, y, T_f)$ generated using these parameter values results

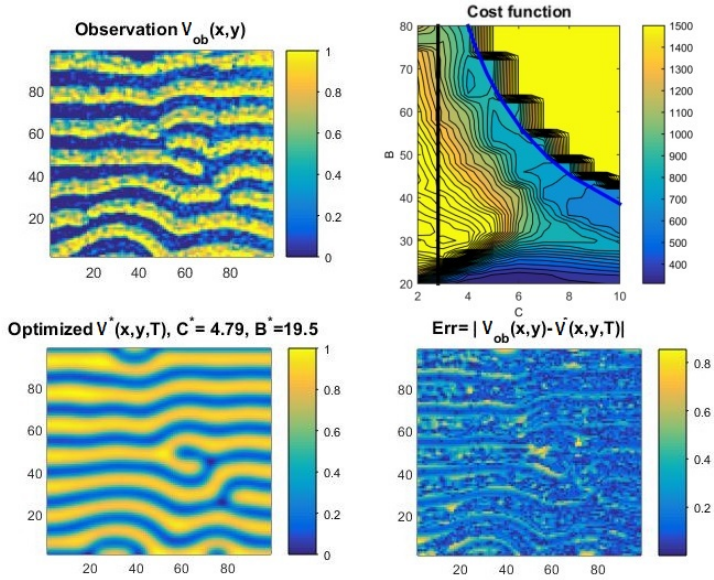

Figure 6.4: PDE-PIP-DIB: Top line: experimental stripes (left) and cost function in (5.4) (right). Bottom line: simulated solution for the minimun value $(C^*, B^*) = (4.7914, 19.5127)$ and absolute error map. Taken from [81].

in a relative error in the Frobenius norm of $rel^*_{err} = 0.438$. Then the authors in [81] solved PIP(ii) by using $(C_0, B_0) = (5, 20)$ as the first guess of the optimization algorithm without regularization; the algorithm converged to values of $(C^*, B^*) = (4.7914, 19.5127)$, with a small improvement in the solution. The corresponding pattern $v^*(x, y, T_f)$ is shown in Figure 6.4 on the bottom left, and on the bottom right the absolute error is represented.

6.3 Fourier approach for DIB-PIP

The target patterns we examined are experimental images which represent electrochemical distributions, that can be described as a linear combination of sines and cosines in space. For this reason we start to extend with appropriate changes the Fourier approach to the PDE case: we show that by using the classical 2-norm as cost function in the PIP(i) we could lose some important information deriving from the model. Therefore we compare the classical approach of fitting the model in a least-square sense, where the cost function is defined as usual as:

$$J_{N2}(C, B) = \|\mathbf{V}_{T_f}(C, B) - \widetilde{\mathbf{M}}\|_2 \tag{6.4}$$

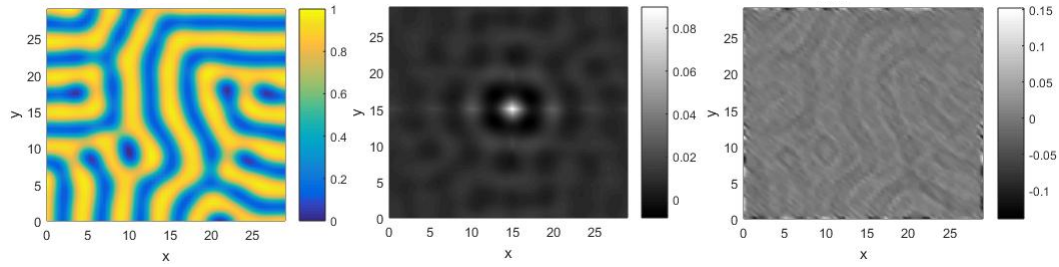

Figure 6.5: Example of image reconstruction by magnitude and phase spectra: on the left the original pattern obtained by the DIB model with the parameters setting as indicated in the main text, in the center the magnitude-only reconstruction, on the right the phase-only reconstruction.

where $\mathbf{V}_{T_f}(C, B)$ and $\widetilde{\mathbf{M}}$ are defined as in (5.4), with a new approach that develops in the Fourier space. In particular we define different cost functions which take into account the spectral properties of the pattern. To define the Fourier costs we need the FFT in 2-D of the final pattern $\mathbf{V}_{T_f}$ (see Appendix A for details). Hence, by using the MATLAB [56] function fft2, we define:

$$
\begin{aligned}
F &= \text{fft2}(\mathbf{V}_{T_f}), \\
P &= |F|, \\
\Phi &= tan^{-1}\frac{Im(M)}{Re(M)},
\end{aligned}
$$

where $F \in \mathbb{C}^{N_x \times N_y}$, $P \in \mathbb{R}^{N_x \times N_y}$ is the so-called *magnitude* and Φ the *phase* of FFT. Now we can not avoid considering the phase of FFT, because its essential information in the position of the image features. To give an example of the importance of the phase in the images reconstruction, let us consider the pattern shown in Figure 6.5 on the left, obtained as numerical solution of the DIB model get by setting $C = 3$, $B = 66$, $A_1 = 10$, $A_2 = 1$, $\alpha = 0.5$, $\gamma = 0.2$, $k_2 = 2.5$, $k_3 = 1.5$,$d = 20$, $\rho = 1$ and $T_f = 50$. The phase values determine the shift in the sinusoidal components of the pattern. With zero phase, all the sinusoidal are centered at the same position and by reconstructing the image without phase information we obtain a symmetric image whose structure has no correlation with the original pattern at all, as shown in Figure 6.5 in the center. Being centered at the same location means that the sinusoidal are a maximum at that location, and is why there is a big white patch in the middle of pattern. When we do a phase-only reconstruction, we set all the magnitude to one: it changes the shape

of the features but not their location, as shown in 6.5 on the right.

Therefore we define different Fourier cost functions, in order to consider dominant frequencies in both spatial directions, the magnitude and the phase information. In particular we define the FFT-cost, the two phase-costs and the module-cost respectively:

$$J_{FFT}(C,B) = \sqrt{(f_x(C,B) - \widetilde{f}_x)^2 + (f_y(C,B) - \widetilde{f}_y)^2} = \|\mathbf{f}(C,B) - \widetilde{\mathbf{f}}\|_2, \quad (6.5)$$

$$J_{\phi}(C,B) = \|\Phi(C,B) - \widetilde{\Phi}\|_2, \quad (6.6)$$

$$J_{\phi+\pi}(C,B) = \|\Phi_{\pi}(C,B) - \widetilde{\Phi}\|_2, \quad (6.7)$$

$$J_{P}(C,B) = \|P(C,B) - \widetilde{P}\|_2, \quad (6.8)$$

where $\widetilde{f}_x$ and $\widetilde{f}_y$ are the first dominant frequencies in x and y direction respectively of the target pattern $\widetilde{\mathbf{M}}$, $f_x(C,B)$ and $f_y(C,B)$ are the frequencies of $\mathbf{V}(C,B)$ for $(C,B) \in \Omega_h$, $\widetilde{\mathbf{f}} = [\widetilde{f}_x \ \widetilde{f}_y]'$ and $\mathbf{f}(C,B) = [f_x(C,B) \ f_y(C,B)]'$. Let us note that we compare also $\Phi_{\pi}(C,B) = \Phi(C,B) + \pi$ with the experimental phase: this because we admit the pattern where the maxima and minima are reversed as acceptable solution. These kinds of pattern are admissible solutions for the model and the presence of one of them is caused by the initial conditions, in particular the structure of X_u and X_v in 5.6. $J_{FFT}(C,B)$ is able to find the area in the parameters space where the patterns have the same first dominant frequencies in both directions. Let Ω_h^{FFT} be this set, defined as follows:

$$\Omega_h^{FFT} = \{(C,B) \in \Omega_h | J_{FFT}(C,B) = 0\}. \quad (6.9)$$

In the following numerical example we use $rSBDF$ (see Section 5.3) as PDEs solver.

6.3.1 Simulated data: *Square*-patterns

In this first case let us consider a simulated patterns for which we know the analytic approximation given by a particular choice of parameters of the DIB model. In [9], the authors used a specific nonlinear bifurcation technique to characterize the shape and amplitude of the pattern close to the boundary of the Turing bifurcation threshold of the physically relevant equilibrium. By fixing $A_1 = 54$, $A_2 = 50$, $B = 65$, $C = 12$, $k_2 = 9.42$, $k_3 = 1$, $\alpha = 0.2$, $\beta = 0.2$,

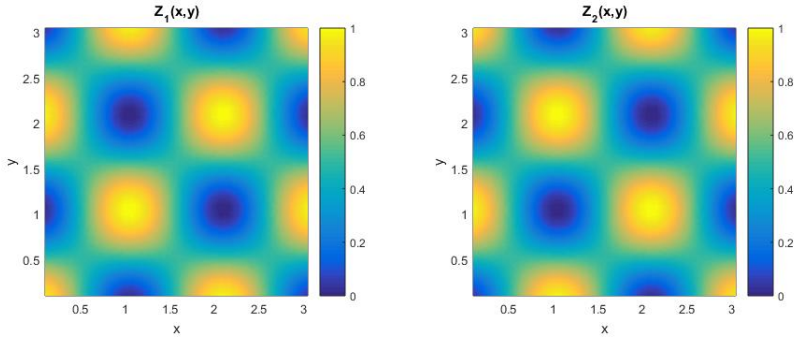

Figure 6.6: Square-patterns: On the left the map $\mathbf{Z}_1$, on the right the map $\mathbf{Z}_2$.

$D = 38.7692$ and $d = d_c(1+\epsilon^2)$, with $d_c = 6.9503$ and $\epsilon = 0.1$, the Turing pattern stationary solution $u(x, y, T_f)$ could be well approximated by:

$$w(x, y) = 0.082172 \cos(3x) \cos(3y) + O(\epsilon^2). \tag{6.10}$$

Let us consider two different target maps, chosen as follows:

$$\begin{aligned} &\mathbf{Z}_1(x, y) = 0.082172 \cos(3x) \cos(3y), \\ &\mathbf{Z}_2(x, y) = 0.082172 \cos(3x + \pi) \cos(3y), \quad (x, y) \in [0, \pi] \times [0, \pi], \end{aligned} \tag{6.11}$$

where $\mathbf{Z}_2$ has a shift of π along the x direction wrt respect to $\mathbf{Z}_1$: the shift of π reverses the pattern exchanging the maxima and the minima. We normalize both the patterns and represent them in Figure 6.6. Then we compute the cost functions defined in (6.4)-(6.8), for $(C, B) \in \Omega_h = [10, 14] \times [55, 75]$, by fixing $h_C = 0.2$ and $h_B = 0.5$ as discretization steps in the C and B directions respectively to obtain the discrete set, $T_f = 100$, $h_t = 1e-03$ for the time integration and X_u and X_v as random perturbation matrices. Let us indicate as J^1, J^2 the cost functions referred to $\mathbf{Z}_1$ and $\mathbf{Z}_2$ respectively. In Figure 6.7 we can observe and compare the J^i, $i = 1, 2$. The curves represent the boundary of the Turing region $\mathcal{R}$.

The 2-norms are clearly different: as we expect, $J_2^1(C, B)$ individuates a minimun in $(C, B) = (12, 65.5)$, but when we look for the reversed pattern the cost has a zone of maxima along the boundary of the Turing region; therefore we can deduce that the model does not admits as solution this kind of pattern, since we known that the technique used in [9] is able to approximate the solution on the boundary of the Turing region. The magnitude costs are essentially the same for $\mathbf{Z}_1$ and $\mathbf{Z}_2$; $J_P^i(C, B)$, $i = 1, 2$, individuates the minimum in $(C, B) = (12, 65.5)$,

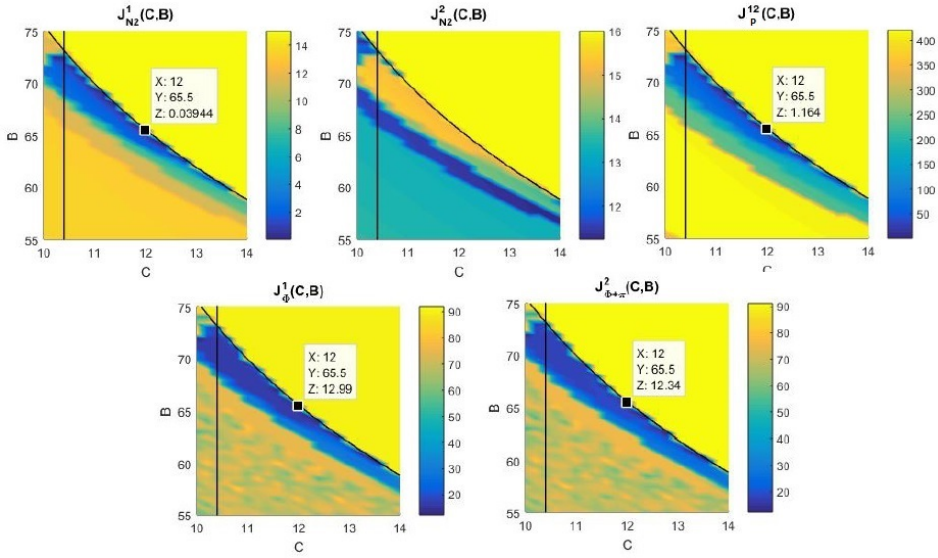

Figure 6.7: Square-patterns: On the top line the two 2-norms and the magnitude cost (which is the same for both pattern). On the bottom line the phase costs.

and seems that the comparison of the magnitudes includes more information with respect to the classical 2-norm: although the starting pattern is inverted, the magnitude is able to identify the pattern that has the same spatial structure. $J_P(C, B)$ contains also the information deriving from phase costs: $J^1_\Phi(C, B)$ individuates the solutions with the nearest phase of $\mathbf{Z}_1$ and $J^2_{\Phi+\pi}(C, B)$ individuates the solutions with the nearest phase of $\mathbf{Z}_2$. Both this information are in $J_P(C, B)$ that localize the minima in a smaller area.

This can be very useful in case of experimental map, in which case we do not know a priori if the PDEs model admits the pattern we are looking for, its reversed form or both. Then the study of the magnitude cost can give us more information than the study of the classical 2-norm. We will observe these advantages in the following Sections where we consider two experimental maps, and show the lose of information that can emerge from studying only the 2norm.

6.3.2 Experimental data

Stripes Let us consider the experimental pattern examined in Section 6.2, shown in Figure 6.8 on the top left and fix the parameters of the model as follows: $A_1 = 10$, $\alpha = 0.5$, $\gamma = 0.2$, $k_2 = 2.5$, $k_3 = 1.5$, $d = 20$ and $\rho = 1$. Let $\Omega = [2, 10] \times [20, 80]$ be the subset of the $C - B$ plane where we construct the cost

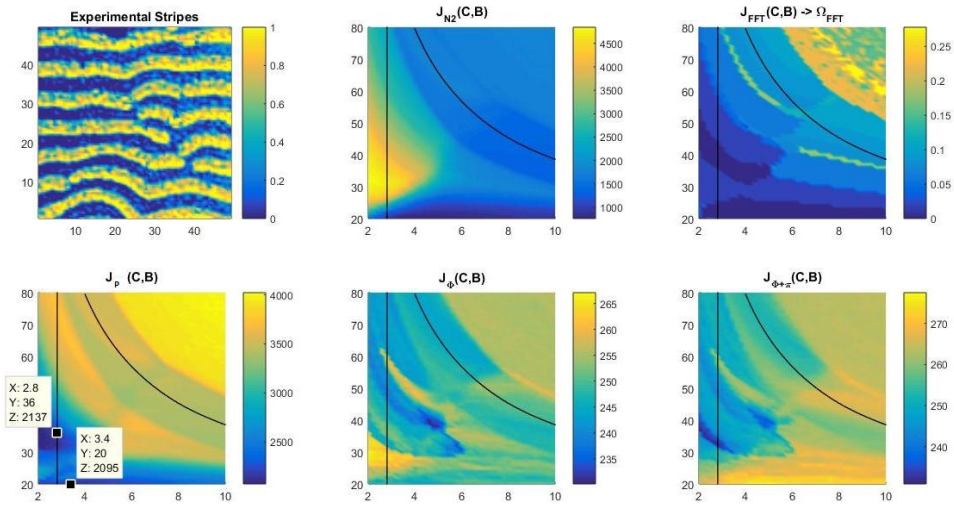

Figure 6.8: Experimetal stripes: On the top: the experimental pattern (left), the 2-norm (center) and the FFT-cost (right). In bottom row the Fourier costs for the magnitude and the phase.

functions, and $h_C = 0.2$, $h_B = 1$ the discretization steps in the C and B direction respectively to obtain the discrete set Ω_h. Let us fix the $A_2 = 30$, $T_f = 20$, and X_u and X_v as the normalize experimental map. In Figure 6.8 we observe the cost functions: the 2-norm defined as in (6.4), the magnitude cost (6.8), the FFT-cost in (6.5) and the phase-costs defined as in (6.6)-(6.7).

The FFT-cost identifies two areas included in the set Ω_h^{FFT}, where the simulations have the same first dominant frequencies of the experimental pattern, which correspond to the zones of maxima and minima in J_{N2}. Instead J_P has both zones of minima that are included in Ω_h^{FFT}: again the information that are in the magnitude does not compare by using in the classical 2-norm.

We can identify two minima of J_P in the intersection of the Turing region $\mathcal{R}$ and the set Ω_h^{FFT}. The value of parameters are shown with the datatips in the Figure 6.8; their values are: $(C, B) = (2.8, 36)$ (which is on the boundary of the Turing region) and $(C, B) = (3.4, 20)$ and the corresponding simulations are shown in Figure 6.9. The minimum in $(C, B) = (2.81, 36)$ of J_P seems to be very similar to the reversed experimental map: it is located in the area of maxima of the 2-norm and is also in the zone of minima of $J_{\Phi+\pi}$. If the model admits reversed-solution, we can not find them by using the 2-norm, but we realize their

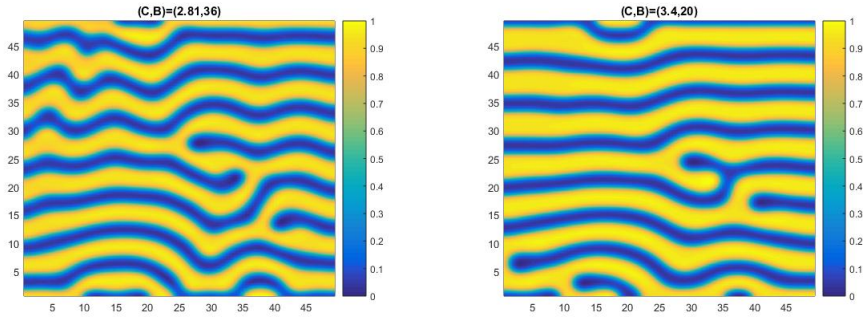

Figure 6.9: Experimetal stripes: minima of the magnitude cost indicated by the data-tips in Figure 6.8, obtained for $(C, B) = (2.81, 36)$ and $(C, B) = (3.4, 20)$

presence by looking the magnitude-cost. An alternative could be to use the 2-norm by computing the simulations with the "reversed" initial condition: but in this way we have to compute the simulation twice and store two different cost functions. J_P not only gives us additional information but also includes all the information that we can derived by studying the other cost function: it localizes the minima of J_{N2} in a smaller area and individuates the simulations that have both the same frequencies (in J_{FFT}) and the same phase (or reversed phase) of the experimental map (in J_Φ and $J_{\Phi+\pi}$).

Labyrinths Let us consider a new experimental map shown in Figure 6.10 on the top left. Let $\Omega = [2, 10] \times [20, 80]$ be the subset of the $C - B$ plane where we construct the cost functions, and $h_C = 0.2$, $h_B = 1$ the discretization steps in the C and B direction respectively to obtain the discrete set Ω_h. Let us fix the $A_2 = 30$, $T_f = 30$, and X_u and X_v as the normalize experimental map. In Figure 6.10 the cost functions are displayed: the 2-norm (6.4), the magnitude cost (6.8), the FFT-cost in (6.5) and the phase-costs (6.6)-(6.7).

From the magnitude cost we can deduce that also in this case the model admits both the pattern and its reversed form: the minima in the two different zone of J_P are shown by the datatips in the Figure 6.10. One of them, $(C, B) = (10, 20)$ is also in the minima areas of J_{N2} and J_Φ, the other one; $(C, B) = (2.8, 31)$ is in zone of maxima of the 2-norm and the minima of $J_{\Phi+\pi}$ and corresponds to the reversed pattern. The corresponding simulations are shown in Figure 6.11.

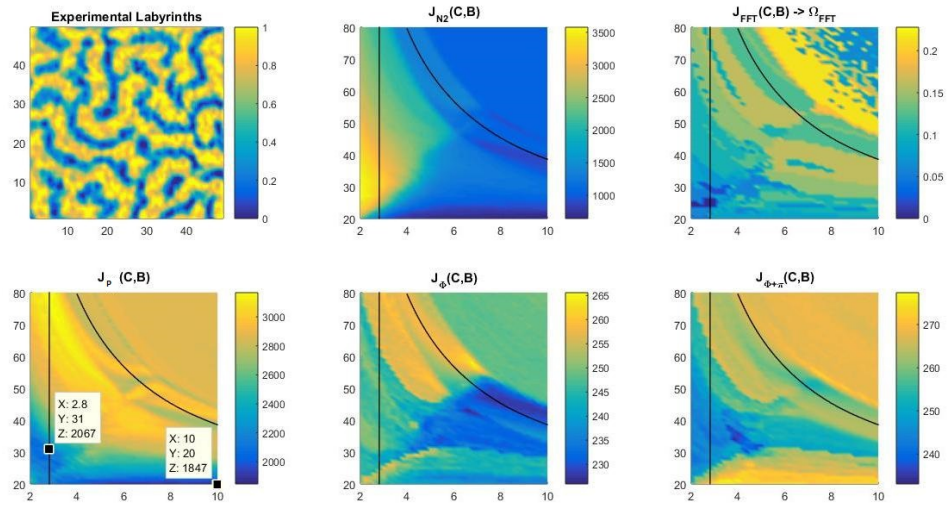

Figure 6.10: Experimetal labyrinths: On the top: the experimental pattern (left), the 2-norm (center) and the FFT-cost (right). In bottom row the Fourier costs for the magnitude and the phase.

6.4 Summary

The Fourier approach for the DIB-PIP described above proves to be an effective tool for the localization of the zone in the Turing region where the simulated patterns exhibit the same spatial feature of the experimental map. It can be used not only for the PIP(i), but the magnitude cost could be a more accurate tool and interesting choice for minimization in the PIP(ii) as an alternative to the classical 2-norm. In fact, by including also the phase information, the magnitude allows us to realize the presence of (admissible) reversed patterns in the model and find them in the Turing region.

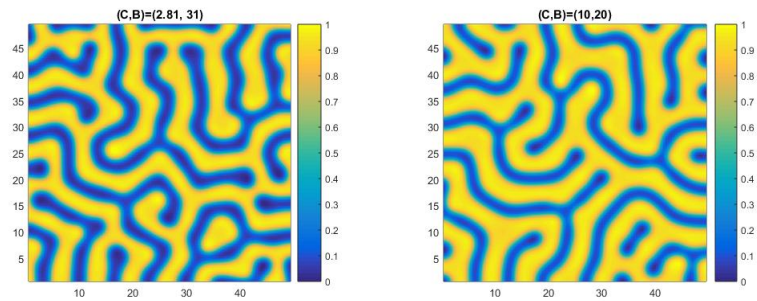

Figure 6.11: Experimetal labyrinths: minima of the magnitude cost indicated by the data-tips in Figure 6.10, obtained for $(C, B) = (2.81, 31)$ and $(C, B) = (10, 20)$

Chapter 7

Future work

In this thesis we studied different numerical aspects involved in the discretization of a PIP, by using both the Direct and the Indirect approaches. Due to the complexity of the examined problems, after analyzing the two approaches in detail, it was decided to consider only the Direct approach. A future extension could concern the use of the Indirect approach for the same problems with some suitable changes.

Then we introduced the so-called Fourier regularization approach for the ODE-PIP with oscillating data, which is able to identify the sub manifold of the parameter space that contains the solution with the same frequency of the experimental data. In the present work we have confined the application of our original method to the first dominant frequency, but future extensions will consider a more general cost function including higher frequencies to better account for the fine structure of the oscillating phenomena under investigation. This will further improve the results described, leading to a stronger tool for ODE control problems.

In Section 5.3 we solved numerically the RD-PDEs which represent the constraints for the PDE-PIP, by introducing the matrix-oriented formulation of the semi-discretized equations. Future work may involve the implementation of higher order methods in space and time for the problem in matrix form. For example, by using the Extended Central Difference Formulae (ECDFs) of high order [1], we can improve the spatial approximation of the second order derivatives, and this would change the structure of the coefficient matrices T_1 and T_2. Then, an high order IMEX-methods for the time integration will modify the Sylvester equations to solve at each h_t. The stability analysis, the convergence and the computa-

tional time will be object of future studies. Moreover, an interesting future work could concern the implementation of the so-called fractional step θ-method [35] in matrix form, studied in [53] for the RD-PDE system.

In the Chapter 6, after the description of the morphochemical model (DIB), we presented the first results in the extension of the Fourier approach to the PDE case. As we have seen, the use of the classical 2-norm may imply the loss of some important features of the data and consequently some admissible solutions in the parameters space might not be found. Instead, the magnitude cost, which requires the two-dimensional Fourier transform of the patterns, not only give us the information already present in the 2-norm, but it also contains the spectral information that is indispensable in identifying at the same time the pattern and its "reversed". Future work will concern the implementation of an optimization algorithm whose cost function can be the magnitude cost in (6.8). At last, a future further extension will consider the DIB-PIP where the solutions have an oscillating behavior both in space and time. In fact, in [47] the authors showed that near the Turing-Hopf (TH) bifurcation point (see Figure 6.1) a class of spatio-temporal patterns emerges, due the interaction between the formation of inhomogeneous stationary patterns caused by Turing instabilities with the homogeneous oscillations caused by a Hopf bifurcation. A first *map identification* was done in [78], where the authors looked for the time t^* belonging to the transient dynamics, such that the PDE solution approximates a given experimental map for a fixing set of parameters. An extension of this work could concern the minimization of a cost function that depends both from parameter $\mathbf{p}$ and the time: $J = J(t, \mathbf{p})$.

Appendix A

Fast Fourier Transform

The Fast Fourier Transform (FFT) is an algorithm that compute the Discrete Fourier Transform (DFT), and its inverse, of a sequence of values which represent a equally-spaced samples of a periodic signal. DFT is a one-to-one transform, which converts the signal from its domain to a representation in the frequency domain. Given the sequence of complex numbers $\{x_n\} = x_0, ..., x_{N-1}$ that represents the input signal, then the DFT is a sequence of complex numbers $\{X_k\} = X_0, ..., X_{N-1}$, defined as follows:

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi}{N} kn} \quad k = 0, ..., N-1 \tag{A.1}$$

In matrix form (A.1) can be formulated as:

$$X = \frac{1}{N} V^H x \tag{A.2}$$

where V is the Vandermonde matrix whose entries are $v_{kj} = \omega_n^{kj}$, with ω_n the complex nth root of unity, and V^H is its conjugate transpose. If the vector x is real, then the entries of the vector X are: X_0 real and $\overline{X}_j = X_{N-j}$ for $j = 1, ..., N-1$.

The inverse of the DFT (IDFT), is given by:

$$x_n = \sum_{k=0}^{N-1} X_k e^{\frac{i2\pi}{N} kn} \quad n = 0, ..., N-1. \tag{A.3}$$

Evaluating the definitions(A.1) and (A.3) directly requires $O(N^2)$ operations. The FFT takes advantage of the special properties of the complex roots of unity to compute DFT(x) and IDFT(x) in time $O(NlogN)$.

Fourier transform is defined not only for one-dimensional signal, but for functions of arbitrary dimensions. In particular, for a two dimensional periodic signal g (e.g. an image) of dimension $N \times M$, the 2D-DFT is defined as:

$$G(n,m) = \frac{1}{NM} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} g(u,v) e^{-i2\pi\left(\frac{nu}{N}+\frac{mv}{M}\right)} \; n = 0, ..., N-1, \; m = 0, ..., M-1 \tag{A.4}$$

where G is again a two-dimensional function of the same dimension $(N \times M)$ as the original signal. The 2D-IDFT is defines as:

$$g(u,v) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} G(n,m) e^{i2\pi\left(\frac{nu}{N}+\frac{mv}{M}\right)} \; u = 0, ..., N-1, \; v = 0, ..., M-1. \tag{A.5}$$

Given a matrix g of dimension $N \times M$, the algorithm corresponds to first performing the FFT of all the rows, grouping the resulting transformed rows together as another $N \times M$ matrix, and then performing the FFT on each of the columns of this second matrix, and similarly grouping the results into the final result matrix.

Appendix B

Algebraic curvature

Let us consider a parametric curve defined as follows:

$$\alpha(t) = (x(t), y(t)), \quad t \in \mathbb{R}_0^+. \tag{B.1}$$

The algebraic curvature is defined as follows:

$$\mathcal{K}(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(\sqrt{x'(t)^2 + y'(t)^2})^3} \tag{B.2}$$

The algebraic curvature gives us information about the direction of rotation of the tangent. $\mathcal{K}(t)$ can be either positive or negative, in particular it results: if $\mathcal{K}(t) > 0$ the curve traces out in a counterclockwise direction, if $\mathcal{K}(t) < 0$ the curve traces out in a clockwise direction. Hence when $\mathcal{K}(t)$ changes sign there is a so-called *inflection point*.

Acknowledgements

I would like to express my deep gratitude to Professor Ivonne Sgura, my academic supervisor, for her patient guidance and enthusiastic encouragement of my research and thesis. It would not have been possible to succeed at a work like this one without her great contribution and advice.

I would also like to thank Professor Benedetto Bozzini, for his assistance and support in the work we have done together.

I am particularly grateful to Professor Valeria Simoncini for her valuable and constructive suggestions and useful critiques; she gave me also warm encouragement.

I would also like to extend my thanks to Prof Giuseppe Notarstefano and his research team for their help and interest in my work.

Special thanks to my colleagues and friends for their precious support.

Finally, I wish to thank my parents for their unconditional love, help and encouragement throughout my study.

Bibliography

[1] P Amodio and I Sgura. High order finite difference schemes for the solution of second order bvps. *J. Comput. Appl. Math.*, 176(1):59–76, 2005.

[2] U M Ascher, S J Ruuth, and B T R Wetton. Implicit-explicit methods for time dependent pde's. *J. Numerical Analysis*, 32(3):797–823, 1995.

[3] C H L Beentjes. Pattern formation analysis in the schnakenberg model. *Technical Report, University of Oxford, Oxford, UK*, 2015.

[4] M Benzi and V Simoncini. Approximation of functions of large matrices with kronecker structure. *Journal Numerische Mathematik*, 135(1):1–26, 2017.

[5] J T Bett. Survey of numerical methods for trajectory optimization. *Journal of Guidance Control and Dynamics*, 21(2):193–207, 1998.

[6] A Björck. Numerical methods for least squares problems. *Discrete & Continuous Dynamical Systems-A*, 51, 1996.

[7] K N Blazakis, A Madzvamuse, and CC Reyes-Aldasoro. Whole cell tracking through the optimal control of geometric evolution laws. *J of Computational Physics*, 297:495–514, 2015.

[8] B Bozzini, M C D'Autilia, C Mele, and I Sgura. Dynamics of zinc-air battery anodes: an electrochemical and optical study complemented by mathematical modelling. *Memory n 37.009, Proceeding 37° AIM Congress*, 2018.

[9] B Bozzini and G Gambini. Weakly nonlinear analysis of turing patterns in a morphochemical model for metal growth. *Comp. & Math. App*, 70(8):1948–1969, 2015.

[10] B Bozzini, D Lacitignola, and I Sgura. Spatio-temporal organization in alloy electrodeposition: a morphochemical mathematical model and its experimental validation. *Journal of Solid State Electrochemistry*, 17(2):467–479, 2013.

[11] C G Broyden. The convergence of a class of double-rank minimization algorithms. *Journal Inst. Math. Applic.*, 6:76–90, 1970.

[12] R Bulirsch, E Nerz, HJ Pesch, and O von Stryk. Combining direct and indirect methods in optimal control: Range maximization of hang glider. *Optimal control*, pages 327–288, 1993.

[13] M Burger and W Mühlhuber. Iterative regularization of parameter identification problems by sequential quadratic programming methods. *Inverse Problem*, 18:943–969, 2002.

[14] J C Butcher. Implicit runge-kutta processes. *Math. Comp*, 18:50–64, 1964.

[15] E Campillo-Funollet, C Venkataraman, and A Madzvamuse. Bayesian parameter identification for turing systems on stationary and evolving domains. *Bulletin of mathematical biology*, 81(1):81–104, 2019.

[16] T F Coleman and Y Li. On the convergence of reflective newton methods for large-scale nonlinear minimization subject to bounds. *Mathematical Programming*, 67(2):189–224, 1994.

[17] T F Coleman and Y Li. An interior, trust region approach for nonlinear minimization subject to bounds. *Journal on Optimization*, 6:418–445, 1996.

[18] O Cots, J Gerguard, and D Goubinat. Direct and indirect methods in optimal control with state constraints and the climbing trajectory of an aircraft. *Optim. Control Appl. Meth*, pages 1–22, 2017.

[19] W Croft, C M Elliott, and G Ladds. Parameter identification problems in the modelling of cell motility. *J of Mathematical Biology*, 71:399–436, 2015.

[20] M C D'Autilia, I Sgura, and B Bozzini. Parameter identification in ode models with oscillatory dynamics: a fourier regularization approach. *Inverse Problem*, 33(12):124009, 2017.

[21] M C D'Autilia, I Sgura, and V Simoncini. Matrix-oriented discretization methods for reaction-diffusion pdes: comparisons and applications. *arXiv:1903.05030*, 2019.

[22] J E Dennis and R B Schnabel. Numerical methods for unconstrained optimization and nonlinear equations. *Englewood Cliffs, NJ Prentice-Hall*, 1983.

[23] P Deuflhard and S Röblitz. A guide to numerical modelling in systems biology. *Springer*, 12, 2015.

[24] T Dierkes, S Röblitz, M Wade, and P Deuflhard. Parameter identification in large scale kinetic networks with bioparkin. *CoRR*, abs/1303.4928, 2013.

[25] H Egger, T Kugler, and N Strogies. Parameter identification in semilinear hyperbolic system. *Inverse Problem*, 33, 2017.

[26] S P Eller, Y Seifu, and R H Smith. Fitting population dynamic models to time-series data by gradient matching. *Ecology*, 83:2256–2270, 2002.

[27] H Engl, M Hanke, and A Neubauer. Regularization of inverse problems. *Mathematics and Its Applications, Vol. 375*, 1996.

[28] F Feng, P Edström, and M Gullikssone. Levenberg-marquardt methods for parameter estimation problems in the radiative transfer equation. *Inverse Problem*, 23:879–891, 2007.

[29] R Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13:317–322, 1970.

[30] J Frank, W Hundsdorfer, and J G Verwer. On the stability of implicit-explicit linear multistep methods. *Appl. Numer. Math.*, 25:193–205, 1997.

[31] M R Garvie, P K Maini, and C Trenchea. A methodology for parameters identification in turing systems. 2009.

[32] M R Garvie, P K Maini, and C Trenchea. An efficient and robust numerical algorithm for estimating parameters in turing systems. *J. of Computational Physics*, 229:7058–7071, 2010.

[33] M R Garvie and C Trenchea. Identification of space-time distributed parameters in the gierer-meinhardt reaction-diffusion system. *J Appl Math.*, 74(1):147–166, 2014.

[34] R Giering and T Kaminski. Recipes for adjoint code construction. *ACM Trans Math Softw.*, 24:437–474, 1998.

[35] R Glowinski. Finite element methods for incompressible viscous flow. *Handbook of Numerical Analysis*, 9, 2003.

[36] E Hairer, C Lubich, and G Wanner. Geometric numerical integration illustrated by the störmer verlet method. *Acta Numerica*, 12:399–450, 2003.

[37] E Hairer, S P Nø rsett, and G Wanner. Solving ordinary differential equation i, nonstiff problems. *Springer, Berlin*, 2nd ed., 1993.

[38] P J Hansen. Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion. *SIAM*, 4, 1998.

[39] J Hauser. A projection operator approach to the optimization of trajectory functionals. *IFAC world congress*, 15, 2002.

[40] M Hochbruck and A Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010.

[41] C Hogea, C Davatzikos, and G Biros. An image-driven parameter estimation problem for a reaction-diffusion glioma growth model with mass effects. *J of Mathematical Biology*, 56:793–825, 2008.

[42] B Jin and P Maass. Sparsity regularization for parameter identification problems. *Inverse Problem*, 28, 2012.

[43] J Kaipio and E Somersalo. Statistical and computational inverse problems. *Springer, New York*, 2006.

[44] E A Kendall. An introduction to numerical analysis. *John Wiley & Sans*, 1988.

[45] I Krastev and M T M Koper. Pattern formation during the electrodeposition of a silver-antimony alloy. *Phys A.*, 2013:199–208, 1995.

[46] D Lacitignola, B Bozzini, M Frittelli, and I Sgura. Turing pattern formation on the sphere for a morphochemical reaction-diffusion model for electrodeposition. *Communications in Nonlinear Science and Numerical Simulation*, 48:484–508, 2017.

[47] D Lacitignola, B Bozzini, and I Sgura. Spatio-temporal organization in a morphological electrodeposition model: Hopf and turing instabilities and their interplay. *European Journal of Applied Mathematics*, 26:143–173, 2015.

[48] A S Lawlwss, M J P Cullen, M A Freitag, S Kindermann, and R Scheichl. Variational data assimilation for very large environmental problems. *Large Scale Inverse Problems: Computational Methods and Applications*, 13:55–90, 2013.

[49] S Lenhart and J T Workman. Optimal control applied to biological models. *Chapman & Hall/CRC Mathematical and Computational Biology Series*, 2007.

[50] P Liu, J Shi, Y Wang, and X Feng. Bifurcation analysis of reaction-diffusion schakenberg model. *J Math Chem*, 51:2001–2019, 2013.

[51] B Macdonald and D Husmeier. Gradient matching methods for computational inference in mechanistic models for systems biology: A review and comparative analysis. *Frontiers in bioengineering and biotechnology*, 3:180, 2015.

[52] A Madzvamuse. Time-stepping schemes for moving grid finite elements applied to reaction-diffusion systems on fixed and growing domains. *Journal of Computational Physics*, 214:239–263, 2006.

[53] A Madzvamuse and AHW Chung. Fully implicit time-stepping schemes and non-linear solvers for systems of reaction-diffusion equations. *Applied Mathematics and Computation*, 244:361–374, 2014.

[54] P Maini and H Othmer. Mathematical models for biological pattern formation. *The IMA Volumes in Mathematics and its Applications - Frontiers in application of Mathematics*, 2001.

[55] D W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11:431–441, 1963.

[56] MATLAB. *R2015a*. The MathWorks Inc., Natick, Massachusetts, 2015.

[57] H Mena, A Ostermann, L M Pfurtscheller, and C Piazzola. Numerical low-rank approximation of matrix differential equations. *J. Comp.Applied Mathematics*, 340:602–614, 2018.

[58] K W Morton and D F Mayers. Numerical solution of partial differential equations. *Cambridge University Press*, 2005.

[59] D P Moualeu-Ngangue, R Röblitz, R Ehrig, and P Deuflhard. Parameter identification in a tuberculosis model for cameroon. *PLoS ONE*, 10(4):e0120607, 2015.

[60] I M Navon. Practical and theoretical aspects of adjoint parameter estimation and identifiability in meteorology and oceanography. *Dynamics of Atmospheres and Oceans*, 27:55–79, 1988.

[61] M G Neubert and H Caswell. Alternatives to resilience for measuring the responses of ecological systems to perturbations. *Ecology*, 78:653, 1997.

[62] M G Neubert, H Caswell, and Murray J D. Alternatives to resilience for measuring the responses of ecological systems to perturbations. *Math. Biosciences*, 175:1–11, 2002.

[63] J Nocedal and S J Wright. Numerical optimization. *Springer-Verlag, New York*, 1999.

[64] L S Pontryagin, V Boltyanskii, R Gamkrelidze, and E Mishchenko. The mathematical theory of optimal control processes. *L. W. Neustadt, Interscience, New York*, 1962.

[65] A Quarteroni, R Sacco, and F Saleri. Numerical mathematics. *Springer-Verlag, New York*, 2000.

[66] A V Rao. A survey of numerical method for optimal control. *Advances in the Astronautical Sciences*, 135(1):497–528, 2009.

[67] K Ratkovic. Limitations in direct and indirect methods for solving optimal control problems in growth theory. *Industrija*, 44:19–46, 2016.

[68] M R Ricard and S Mischler. Turing instabilities and hopf bifurcation. *J Nonlinear Sci*, 19:476–496, 2009.

[69] S Röblitz, C Stötzel, P Deuflhard, H M Jones, D O Azulay, P H van der Graaf, and S W Martin. A mathematical model of human menstrual cycle for the administration of gnrh analogues. *Journal of theoretical Biology*, 321:8–27, 2013.

[70] J Ruuth. Implicit-explicit methods for reaction-diffusion problems in pattern formation. *J. Math. Biol.*, 34:148–176, 1995.

[71] A Saccon, J Hauser, and A P Aguiar. Optimal control on lie groups: The projection operator approach. *IEEE Transactions on Automatic Contro*, 58(9):2230–2245, 2013.

[72] J M Sanz-Serna. Symplectic runge-kutta schemes for adjoint equations, automatic differentiation, optimal control and more. *Automatic Differentiation, Optimal Control, and More*, 58(1):3–33, 2016.

[73] J M Sanz-Serna and M P Calvo. Numerical hamiltonian problems. *Chapman and Hall, London*, 1994.

[74] R Scherer and H Turke. Reflected and transposed runge-kutta methods. *BIT*, 23:262–266, 1983.

[75] J Schnakenberg. Simple chmical reaction system with limit cycle behaviour. *J Theor Biol*, 81:389–400, 1979.

[76] G Settanni and I Sgura. Devising efficient numerical methods for oscillating patterns in reaction–diffusion systems. *Journal of Computational and Applied Mathematics*, 292:674–693, 2016.

[77] G Settanni and I Sgura. Devising efficient numerical methods for oscillating patterns in reaction-diffusion system. *J. Comput. Appl. Math.*, 292:674–693, 2016.

[78] I Sgura and B Bozzini. Xrf map identification problems based on a pde electrodeposition model. *J. Phys. D: Appl. Phys.*, 50(15):154002, 2017.

[79] I Sgura, B Bozzini, and D Lacitignola. Numerical approximation of Turing patterns in electrodeposition by ADI methods. *Journal of Computational and Applied Mathematics*, 236(16):4132–4147, 2012.

[80] I Sgura, B Bozzini, and D Lacitignola. Numerical approximation of turing patterns in electrodeposition by adi methods. *J. Comput. Appl. Math.*, 236(16):4132–4147, 2012.

[81] I Sgura, A S Lawless, and B Bozzini. Parameter estimation for a morphochemical reaction-diffusion model of electrochemical pattern formation. *Inverse Problems in Science and Engineering*, pages 1–30, 2018.

[82] V Simoncini. Computational methods for linear matrix equations. *SIAM Review*, 58(3):377441, 2016.

[83] M Stoll, J W Pearson, , and Maini P K. Fast solvers for optimal control problems from pattern formation. *Journal of Computational Physics*, 304:27–45, 2016.

[84] A M Stuart. Inverse problems: a bayesian perspective. *Acta Numer.*, 19:451–559, 2010.

[85] S Subchan and R Żbikowski. Computational optimal control, tools and practice. *John Wiley & Sons Ltd*, 2009.

[86] A Tarantola. Inverse problems theory and methods for model parameter estimation. *SIAM*, 89, 2005.

[87] E Trélat. Contrôle optimal: théorie et applications. *Collection "Mathématiques Concrètes". Vuibert Paris (In French)*, 2005.

[88] E Trélat. Optimal control and applications to aerospace: some result and challenges. *Journal of Optimization Theory and Applications*, 154(3):713–758, 2012.

[89] 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.

[90] P J Van Der Houwen and B P Sommeijeir. Explicit runge-kutta(-nystrom) methods with reduced phase errors for computing oscillating solution. *Journal on Numerical Analysis*, 24:595–617, 1987.

[91] O von Stryk and R Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operation Research*, 37:357–373, 1992.

[92] M Wöbbekind, A Kemper, C Büskens, and M Schollmeyer. Nonlinear parameter identification for ordinary differential equations. *Proceedings in Applied Mathematics and Mechanics*, 13:457–458, 2013.

[93] F Zama. Numerical parameters estimation in models of pollutant transport with chemical reaction. *IFIP Conference on System Modeling and Optimization*, pages 574–556, 2011.

[94] T Zhou, W Dubitzky, O Wolkenhauer, K H Cho, and H Yokota. Relaxation oscillation. *Encyclopedia of Systems Biology. Springer, New York*, 2013.