Combining state and parameter estimation problems for reaction-diffusion models

Flavien Alonzo, Ben Mansour Dia, Mazen Saad

Preprint: Combining state and parameter estimation problems for reaction-diffusion models:
Application to the Keller-Segel system and to the pattern formation system.

Flavien Alonzo, Ben Mansour Dia, Mazen Saad

March 1, 2023

Abstract

Inverse problems aim to find the causes of outcoming features knowing the consequences of a model by calibrating the model's parameters to fit data. In this paper, we present a method that solves simultaneously the inverse problem and the state estimation problem associated with reaction-diffusion models and showcase it on two examples: a Keller-Segel system used for the chemotaxis, and a Turing system producing stable spatial patterns. The method is defined as an optimization problem that minimizes the misfit formulated with three different types of errors: on the modelling choices, on the initial state assumption, and on the difference between data and the forward predictive model output. The resolution of those inverse problems rely on the rewriting of the variational systems and leads to solving two systems of partial differential equations combined with the cancel out of a vectorial function that describes the optimality of the coefficients. We solve those inverse problems numerically to calibrate both vectors of state and parameters when given sparse data in time. We use strategical numerical schemes to solve efficiently the resulting coupled system rather than using a classic Newton algorithm. Numerical experiments have been performed with synthetic data to evaluate the efficiency of the proposed method, but also to describe the influence of hyperparameters on the inverse problem.

Keywords: Numerical Analysis, Inverse problems, Reaction-diffusion systems, Numerical simulation

1 Introduction

The combination of the state and the parameter estimations refers to the simultaneous calibration of the state variables of a model (the unknowns of a model) and of the coefficients in the input of the same model. This type of problem is widely common in a variety of fields, e.g., in chemistry [Dochain, 2003], in vehicle system dynamics [Wenzel et al., 2006, Song et al., 2020] or in Oceanography [Simon et al., 2015, Rafiee et al., 2011] to mention few. The usual strategy to address a combined problem of state and parameter estimations is to use Kalman Filters [Chui and Chen, 2017]. This approach is classic in Data Assimilation and Optimal Control and was designed initially with ordinary differential equations (ODE), which are, in general, systems of differential equations with the variability on the time only. It starts with KFs for linear dynamics, but improvements in the design of the filters led to a better handling of nonlinear dynamics. For example, we refer to Unscented Kalman Filters (UKF) [Wan and van der Menwe, 2000] and Ensemble Kalman Filters (EnKF) [Evensen, 2003].

While a lot of approaches in Data Assimilation rely on systems of ODEs, some of them are developed based on systems of partial differential equations (PDE) , see for instance [Iglesias et al., 2013]. Reaction and diffusion terms are present in many systems of PDEs governing real-world phenomenon. This is the case in Biology, see [Murray, 2002, Jones et al., 2009], where the main interest of this paper frames and where the need of solving problems using the data assimilation comes from the personalized medicine. In practice, we are interested in using data from an individual to guide decisions for the prevention, the diagnosis, and the treatment of a disease [NIH, 2022]. There are some research works, in the mathematical setting, on joint state and parameter estimations based on PDE systems modelling

tumour growth, see [Swanson et al., 2000, Konukoglu et al., 2010, Rochoux et al., 2018, Hogea et al., 2008, Grenier et al., 2014].

In practice, counting the use of real data requires to build a methodology that works with sparse data over time. Indeed, in that context, we do not expect to be able to have data when needed. For instance, we cannot impose a Magnetic Resonance Imaging (MRI) on a patient everyday for a large period. A common way to solve this issue, is to interpolate data over time to get the hypothesis of infinite data, which is easier to work with. The remaining issue is then to perform data interpolation as proposed [Collin et al., 2021].

Another problem for solving inverse problems based on PDE systems, is the computational complexity of algorithms to obtain the output of the forward model. Indeed, the calibration the parameters usually demands several evaluations of the forward model for accuracy check. One evaluation of the forward model is already expensive for PDEs-based inverse problems, it is then effective to minimize or at least control the total number of evaluations of the forward model. Various approaches can be indicated to contribute to alleviating the total number of costly approximations of the forward model. Here we refer to [Grenier et al., 2018], that shows an improved method of population parametrization combining the use of a classic algorithm in the field (Stochastic Approximation of Expectation-Maximization) and kriging methods to restrict the number of needs of solving the forward predictive model.

In this work, we develop a method that solve inverse problems based on reaction-diffusion models, without building data interpolation. The method calibrates a state vector ($\boldsymbol{\Psi}$) and a set of parameters ($\boldsymbol{\eta}$) when the data are given as noisy measurements of the predictive forward model. Another strength of the proposed method is that it counts also for the epistemic uncertainty by covering errors in the modelling choices while in the most cases of approaches addressing an optimal control problem, the formulation assumes that the forward modelling does not generate error. Yet, simplifications in modelling complex system, such as the patterns formation, in characterizing some coefficients by assuming its well-knowns can lead to slightly incorrect forward models. For those reasons, we address a formulation that allows errors in the modelling, errors in the given of the initial state, additionally of the observation errors.

The rest of this paper is organized as follows. In Section 2, we introduce the reaction-diffusion systems, on which we detail the inverse problems, that are also clearly stated. Section 3 describes the analytical approach developed to address the inverse problems associated to the Keller-Segel system and to the patterns formation system. In Section 4, we perform the numerical experiments for the inverse problems to showcase the efficiency of the presented approach for the two systems. We also study the hyperparameters of the method in terms of variability of the errors of the state variables and of the adjoint state estimation. We summarize the discussion and give some prospective research lines for the future in Section 5.

2 Inverse problem formulation

2.1 Position of the problem

In this work, we are interested in performing combined state and parameter estimation problems for Reaction-Diffusion systems. More specifically, we will be focused on Keller-Segel type systems and around spatial pattern formation with reaction diffusion mechanisms. For sake of simplicity, the analysis in the present paper treats two species systems, but stays valid for systems with a large number of species. In the next section, we will introduce the two systems that we are interested in here.

2.1.1 Keller-Segel System

In Biology, Keller-Segel type systems are widely used to model chemotaxis, that describes the movement of cells induced by a chemical signal. There is a large variety of such Keller-Segel type system, see [Arumugam and Tyagi, 2020], and with several applications in Biology. For example, modelling taxis toward Hydrogen gas by methanogen [Brileya et al., 2013], studying phenotypic heterogeneity in chemotactic sensitivity for E. Coli [Salek et al., 2019], or aerotaxis around Bacillus subtilis [Menolascina et al., 2017]. In this work, the Keller-Segel system models the chemotaxis between a cell population,

denoted by u, and a chemical signal, namely c,

$$\begin{cases} \frac{\partial u}{\partial t} - \nabla \cdot (D_1 \nabla u) + \nabla.(D_1 \chi u \nabla c) = g_1(u, c, \boldsymbol{\theta}) & ,\forall (t, \boldsymbol{x}) \in I \times \Omega, \\ \frac{\partial c}{\partial t} - \nabla \cdot (D_2 \nabla c) = g_2(u, c, \boldsymbol{\theta}) & ,\forall (t, \boldsymbol{x}) \in I \times \Omega, \\ u(t = 0, \boldsymbol{x}) = u_0(\boldsymbol{x}), \qquad c(t = 0, \boldsymbol{x}) = c_0(\boldsymbol{x}) & ,\forall \boldsymbol{x} \in \Omega, \\ D_1 \nabla u \cdot \vec{n} - D_1 \chi u \nabla c \cdot \vec{n} = 0, \qquad D_2 \nabla c \cdot \vec{n} = 0 & ,\forall (t, \boldsymbol{x}) \in I \times \partial\Omega. \end{cases} \tag{1}$$

We choose u to be the density of tumour cells, between 0 and 1, and c to be the concentration of the chemical signal, we fixed it to be dioxygen (O_2) for example. In that sense, we model the interaction of tumour cells and O_2 in a the time-space domain $I \times \Omega$. This model is a simplified version of the one introduced in [Alonzo et al., 2021], that is used in practise for *Glioblastoma Multiforme* modelling. Tumour cells diffuse naturally with a symmetric diffusion matrix D_1, are attracted by the gradient of O_2 with a chemotaxis coefficient χ, and O_2 diffuses naturally with a symmetric diffusion matrix D_2. We also consider reaction terms for both quantities that are given by the functions g_1 and g_2

$$\begin{cases} g_1(u, c, \boldsymbol{\theta}) = \rho u(1 - u) - \delta u \\ g_2(u, c, \boldsymbol{\theta}) = \alpha u - \beta c - \gamma u c \end{cases}, \text{ with } \boldsymbol{\theta} = (\rho, \delta, \alpha, \beta, \gamma) > 0. \tag{2}$$

The function g_1 models a logistic growth with a rate ρ, plus an apoptosis rate of δ. The function g_2 is given by the O_2 production rate α by the tumour cells, the O_2 degradation rate β, and the O_2 consumption rate γ by the tumour cells.

System (1) is associated with an initial condition u_0 and c_0, for u and c respectively, and we add an homogeneous zero-flux exchange on the boundary, which is a Neumann boundary condition.

In practice, the diffusion and sensitivity coefficients D_1, D_2, and χ are well determined in general, whereas the reaction coefficients ρ, δ, α, β and γ, defined in Equation (2), are not known. So we intend to calibrate $\boldsymbol{\theta} = (\rho, \delta, \alpha, \beta, \gamma)$ using on line measurements.

2.1.2 Pattern Formation System

Some reaction-diffusion systems are known to produce stable spatial patterns during morphogenesis, see the pioneer work [Turing, 1952]. Because spatial patterns are omnipresent in Life, studies around Turing patterns arise in a lot of fields. The general process of patterns developing through diffusive systems will not be detailed in this work, and we refer to [Murray, 2003] for a detailed description of the process. To illustrate our method, we are interested in a dimensionless empirical substrate-inhibition system from [Murray, 2003] modelling the pattern formation. The general form of such system is given by

$$\begin{cases} \frac{\partial A}{\partial t} - \nabla^2 A = \mu f_1(A, B) & ,\forall (t, \boldsymbol{x}) \in I \times \Omega \\ \frac{\partial B}{\partial t} - d\nabla^2 B = \mu f_2(A, B) & ,\forall (t, \boldsymbol{x}) \in I \times \Omega \\ A(t = 0, \boldsymbol{x}) = A_0(\boldsymbol{x}), \qquad B(t = 0, \boldsymbol{x}) = B_0(\boldsymbol{x}) & ,\forall \boldsymbol{x} \in \Omega \\ \nabla A \cdot \vec{n} = 0, \qquad \nabla B \cdot \vec{n} = 0 & ,\forall (t, \boldsymbol{x}) \in I \times \partial\Omega. \end{cases} \tag{3}$$

Here A and B are the concentration of the substrate oxygen and of the enzyme uricase respectively. The isotropic diffusion in oxygen is normalized to 1, whereas, the isotropic diffusion in uricase is given by the coefficient $d > 1$. The reaction terms are given by the functions f_1 and f_2, that have the general form

$$\begin{cases} f_1(A, B) &= k_1 - A - h(A, B), \\ f_2(A, B) &= k_3(k_2 - B) - h(A, B), \end{cases} \text{with } h(A, B) = \frac{k_4 AB}{1 + A + k_5 A^2}, \tag{4}$$

scaled with a coefficient $\mu > 0$. The patterns exhibited by System (3) depend highly on the values of d, μ and $(k_i)_{1 \leq i \leq 5}$. In System (3), we suppose that the coefficients d and μ are unknown, and we aim to learn

$$\boldsymbol{\vartheta} = (d, \mu), \tag{5}$$

given measurements on A and B. This implies that the coefficients $(k_i)_{1 \leq i \leq 5}$ are available to us.

2.2 Measurement of the forward function $\mathscr{H}$

In our work, a measure is related to a time of inquisition t_i in I, a set of m values interpreted as the measurement data. The state vector $\mathbf{\Psi}$ stands for the solution (u, c) in System (1) and the solution (A, B) in system (3).

Formally, a measure is a vector in $\mathbb{R}^{\text{size}(\mathbf{\Psi})m} = \mathbb{R}^{2m}$ that gives the quantities measured of $\mathbf{\Psi}$ at time t_i. A set of N_{mea} measurements is a matrix $D_{mea} \in \mathbb{R}^{N_{mea} \times \text{size}(\mathbf{\Psi})m}$ corresponding to a number of N_{mea} time point values of measurements of $\mathbf{\Psi}$ at times $(t_1, t_2, ..., t_{N_{mea}})$.

The function $\mathscr{H}$ is the link between the state vector and the observation, i.e., $\mathscr{H}(\mathbf{\Psi}(t_i, \cdot))$ is the quantity measured with $\mathbf{\Psi}(t_i, \cdot)$.

For example, if a state vector would contain the height (h) and the weight (w) of an individual but that we were measuring its Body Mass Index (BMI), the measurement function would be $\mathscr{H}(h, w) = \frac{w}{h^2}$. Then, two measures of B.M.I at two different times t_1 and t_2 could be $d_{t_1} = 26.7$ and $d_{t_2} = 25.3$.

Here, we only consider direct measurements, i.e., $\mathscr{H}(\mathbf{\Psi}(t_i, \cdot)) \mapsto (\mathbf{\Psi}(t_i, \boldsymbol{x_j}))_{1 \leq j \leq m}$, which is in $\mathbb{R}^{2m}$ due to the fact that $\mathbf{\Psi}(t_i, \boldsymbol{x_j}) \in \mathbb{R}^2$, but we could also consider any other measurement functions that is enough smooth to use.

2.3 Formulation of the inverse problem

The idea is to find the parameter of interest $\boldsymbol{\theta}$ for System (1), and $\boldsymbol{\vartheta}$ for System (3), when given a set of measurements. In that regard, we need to define a criterion that describes how well the coefficients fit the system. Considering that errors come from the estimation of the initial state, from the accuracy of the analytical model (epistemic uncertainty), and from the discrepancy between the real measurement and the model predictions. The error when estimating the initial state could be a combination of the other two errors. We aim to minimize those three errors.

The classical way to introduce errors in a model, is to add stochastic terms standing for the errors. In our case, we add a stochastic term $\boldsymbol{q_{t,x}} = \left(q^{(1)}_{t,\boldsymbol{x}}, q^{(2)}_{t,\boldsymbol{x}}\right)^\top$ following a centered Gaussian process

$$\mathscr{N}(\boldsymbol{\mu_q}, K_{qq}) \text{ with } \begin{cases} \boldsymbol{\mu_q} : (t, \boldsymbol{x}) \in I \times \Omega & \mapsto \mathbb{E}(\boldsymbol{q_{t,x}}) = (0,0)^\top, \\ K_{qq} : (t, s, \boldsymbol{x}, \boldsymbol{y}) \in I^2 \times \Omega^2 & \mapsto \text{Cov}(\boldsymbol{q_{t,x}}, \boldsymbol{q_{s,y}}) \in \mathbb{R}^{2\times 2}, \end{cases}$$

in the analytical term, where $\mathbb{E}$ denotes the expectancy and Cov the covariance matrix. Then, we add $\boldsymbol{a_x} = \left(a^{(1)}_{\boldsymbol{x}}, a^{(2)}_{\boldsymbol{x}}\right)^\top$ following a centred Gaussian process

$$\mathscr{N}(\boldsymbol{\mu_a}, K_{aa}) \text{ with } \begin{cases} \boldsymbol{\mu_a} : \boldsymbol{x} \in \Omega & \mapsto \mathbb{E}(\boldsymbol{a_x}) = (0,0)^\top, \\ K_{aa} : (\boldsymbol{x}, \boldsymbol{y}) \in \Omega^2 & \mapsto \text{Cov}(\boldsymbol{a_x}, \boldsymbol{a_y}) \in \mathbb{R}^{2\times 2}, \end{cases}$$

in the initial condition and $\boldsymbol{\epsilon}$ following a centered Gaussian law on $\mathbb{R}^{2m}$

$$\mathscr{N}(\boldsymbol{\mu_\epsilon}, \Sigma_{\epsilon\epsilon}) \text{ with } \begin{cases} \boldsymbol{\mu_\epsilon} = \mathbb{E}(\boldsymbol{\epsilon}) = 0_{\mathbb{R}^{2m}} \\ \Sigma_{\epsilon\epsilon} = \mathbb{V}(\boldsymbol{\epsilon}) \in \mathbb{R}^{2m \times 2m} \end{cases}$$

in the measurement process. Where $\mathbb{V}$ denotes the variance of $\boldsymbol{\epsilon}$. More precisely, we consider that we have an error $\boldsymbol{\epsilon}$, independent of the measurement time, associated with the observation $\boldsymbol{d_i} \in \mathbb{R}^{2m}$ following the relation

$$\mathscr{H}(\mathbf{\Psi}(t_i, \cdot)) = \boldsymbol{d_i} + \boldsymbol{\epsilon} \tag{6}$$

In that case, System (1) becomes

$$\begin{cases} du - \nabla \cdot (D_1 \nabla u)dt + \nabla.(D_1 \chi u \nabla c)dt = g_1(u, c, \boldsymbol{\theta})dt + q^{(1)}_{t,\boldsymbol{x}}dt & ,\forall (t, \boldsymbol{x}) \in I \times \Omega, \\ dc - \nabla \cdot (D_2 \nabla c)dt = g_2(u, c, \boldsymbol{\theta})dt + q^{(2)}_{t,\boldsymbol{x}}dt & ,\forall (t, \boldsymbol{x}) \in I \times \Omega, \\ u(t = 0, \boldsymbol{x}) = u_0(\boldsymbol{x}) + a^{(1)}_{\boldsymbol{x}}, \qquad c(t = 0, \boldsymbol{x}) = c_0(\boldsymbol{x}) + a^{(2)}_{\boldsymbol{x}} & ,\forall \boldsymbol{x} \in \Omega, \\ D_1 \nabla u \cdot \vec{n} - D_1 \chi u \nabla c \cdot \vec{n} = 0, \qquad D_2 \nabla c \cdot \vec{n} = 0 & ,\forall (t, \boldsymbol{x}) \in I \times \partial\Omega, \end{cases} \tag{7}$$

and identically System (3) becomes

$$\begin{cases} dA - \nabla^2 A dt = \mu f_1(A,B)dt + q^{(1)}_{t,\boldsymbol{x}} dt & ,\forall (t,\boldsymbol{x}) \in I \times \Omega, \\ dB - d\nabla^2 B dt = \mu f_2(A,B)dt + q^{(2)}_{t,\boldsymbol{x}} dt & ,\forall (t,\boldsymbol{x}) \in I \times \Omega, \\ A(t=0,\boldsymbol{x}) = A_0(\boldsymbol{x}) + a^{(1)}_{\boldsymbol{x}}, \qquad B(t=0,\boldsymbol{x}) = B_0(\boldsymbol{x}) + a^{(2)}_{\boldsymbol{x}} & ,\forall \boldsymbol{x} \in \Omega, \\ \nabla A \cdot \vec{n} = 0, \qquad \nabla B \cdot \vec{n} = 0 & ,\forall (t,\boldsymbol{x}) \in I \times \partial\Omega. \end{cases} \tag{8}$$

We can define the inverse of the previous covariance matrices that will be used later on. We denote by K^{-1}_{qq},K^{-1}_{aa} and $\Sigma^{-1}_{\epsilon\epsilon}$ the inverse of the covariance matrix of K_{qq}, K_{aa} and $\Sigma_{\epsilon\epsilon}$ respectively, defined by

$$\forall (t_1,t_2) \in I^2, \forall (\boldsymbol{x_1},\boldsymbol{x_2}) \in \Omega^2 : \int_I \int_\Omega K^{-1}_{qq}(t_1,s,\boldsymbol{x_1},\boldsymbol{u}) K_{qq}(s,t_2,\boldsymbol{u},\boldsymbol{x_2}) \,\mathrm{du}\,\mathrm{ds} = \delta(\boldsymbol{x_1}-\boldsymbol{x_2})\delta(t_1-t_2)\mathbb{I}_2, \tag{9}$$

$$\forall (\boldsymbol{x_1},\boldsymbol{x_2}) \in \Omega^2 : \int_\Omega K^{-1}_{aa}(\boldsymbol{x_1},\boldsymbol{u}) K_{aa}(\boldsymbol{u},\boldsymbol{x_2}) \,\mathrm{du} = \delta(\boldsymbol{x_1}-\boldsymbol{x_2})\mathbb{I}_2, \tag{10}$$

$$\Sigma^{-1}_{\epsilon\epsilon}\Sigma_{\epsilon\epsilon} = \Sigma_{\epsilon\epsilon}\Sigma^{-1}_{\epsilon\epsilon} = \mathbb{I}_{2m}. \tag{11}$$

Under Gaussian assumptions, finding a maximum of the likelihood is equivalent to minimizing the weighted quadratic errors associated. So we intend to minimize the following payoff function

$$\mathscr{J}(\boldsymbol{\Psi},\boldsymbol{\eta}) \coloneqq e_1(\boldsymbol{\Psi},\boldsymbol{\eta}) + e_2(\boldsymbol{\Psi},\boldsymbol{\eta}) + e_3(\boldsymbol{\Psi},\boldsymbol{\eta}), \tag{12}$$

where e_1, e_2 and e_3 are defined in Equation (13), (16) and (17) respectively.

In Equation (12), $\boldsymbol{\Psi}$ is the state vector of a system i.e. $\boldsymbol{\Psi} = (\Psi_1,\Psi_2) = (u,c)$ for System (1) or $\boldsymbol{\Psi} = (\Psi_1,\Psi_2) = (A,B)$ for System (3) and $\boldsymbol{\eta}$ is the unknown coefficients associated i.e. $\boldsymbol{\eta} = \boldsymbol{\theta}$ for System (1) or $\boldsymbol{\eta} = \boldsymbol{\vartheta}$ for System (3). The quadratic error associated with the modelling uncertainty is defined as

$$e_1(\boldsymbol{\Psi},\boldsymbol{\eta}) \coloneqq \frac{1}{2}\int_I \int\int_\Omega \mathscr{E}(\boldsymbol{\Psi},\boldsymbol{\eta})^\top(t,\boldsymbol{x}) K^{-1}_{qq}(t,\boldsymbol{x},s,\boldsymbol{y}) \mathscr{E}(\boldsymbol{\Psi},\boldsymbol{\eta})(s,y) \,\mathrm{dx}\,\mathrm{dy}\,\mathrm{dt}\,\mathrm{ds}\,. \tag{13}$$

Operator $\mathscr{E}$ is the error of modelling, i.e., for System (7)

$$\mathscr{E}(\boldsymbol{\Psi},\boldsymbol{\eta})(t,\boldsymbol{x}) = \begin{pmatrix} \frac{\partial u}{\partial t} - \nabla\cdot(D_1\nabla u) + \nabla.(D_1\chi u \nabla c) - g_1(u,c,\boldsymbol{\theta}) \\ \frac{\partial c}{\partial t} - \nabla\cdot(D_2\nabla c) - g_2(u,c,\boldsymbol{\theta}) \end{pmatrix}, \tag{14}$$

and for System (8)

$$\mathscr{E}(\boldsymbol{\Psi},\boldsymbol{\eta})(t,\boldsymbol{x}) = \begin{pmatrix} \frac{\partial A}{\partial t} - \nabla^2 A - \mu f_1(A,B) \\ \frac{\partial B}{\partial t} - d\nabla^2 B - \mu f_2(A,B) \end{pmatrix}. \tag{15}$$

The quadratic error associated with the initial condition is

$$e_2(\boldsymbol{\Psi},\boldsymbol{\eta}) \coloneqq \frac{1}{2}\int_\Omega (\boldsymbol{\Psi_0}-\boldsymbol{\Phi_0})^\top(\boldsymbol{x}) K^{-1}_{aa}(\boldsymbol{x},\boldsymbol{y})(\boldsymbol{\Psi_0}-\boldsymbol{\Phi_0})(\boldsymbol{y}) \,\mathrm{dx}\,\mathrm{dy}\,. \tag{16}$$

The quantity $\boldsymbol{\Psi_0}$ is $\boldsymbol{\Psi}(t_0,\cdot)$ while $\boldsymbol{\Phi_0}$ is the guessed initial condition, i.e., $\boldsymbol{\Phi_0} = (u_0,c_0)$ for System (1) and $\boldsymbol{\Phi_0} = (A_0,B_0)$ for System (3). Finally,

$$e_3(\boldsymbol{\Psi},\boldsymbol{\eta}) \coloneqq \frac{1}{2}\sum_{i=1}^{N_{mea}} (\boldsymbol{d_i} - \mathscr{H}(\boldsymbol{\Psi}(t_i,\cdot)))^\top \Sigma^{-1}_{\epsilon\epsilon}(\boldsymbol{d_i} - \mathscr{H}(\boldsymbol{\Psi}(t_i,\cdot))), \tag{17}$$

the quadratic error associated with the difference between measurements and predictions, and is known to be the squared Mahalanobis distance.

We can rewrite the sum from Equation (17) in a vector form with $\boldsymbol{d} = \begin{pmatrix} \boldsymbol{d_1} & \boldsymbol{d_2} & \cdots & \boldsymbol{d_{N_{mea}}} \end{pmatrix}^\top$, $\mathscr{H}(\boldsymbol{\Psi}) = \begin{pmatrix} \mathscr{H}(\boldsymbol{\Psi}(t_1,\cdot)) & \mathscr{H}(\boldsymbol{\Psi}(t_2,\cdot)) & \cdots & \mathscr{H}(\boldsymbol{\Psi}(t_{N_{mea}},\cdot)) \end{pmatrix}^\top$ and finally $[K_{\epsilon\epsilon}^{-1}]$ is a block matrix in $\mathscr{M}_{N_{mea}\times 2m}(\mathbb{R})$ composed of N_{mea} times the matrix $K_{\epsilon\epsilon}^{-1}$ on the diagonal and zeros otherwise, i.e.,

$$[K_{\epsilon\epsilon}^{-1}] = \begin{bmatrix} K_{\epsilon\epsilon}^{-1} & 0_{2m\times 2m} & \cdots & 0_{2m\times 2m} \\ 0_{2m\times 2m} & K_{\epsilon\epsilon}^{-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{2m\times 2m} \\ 0_{2m\times 2m} & \cdots & 0_{2m\times 2m} & K_{\epsilon\epsilon}^{-1} \end{bmatrix}.$$

Lastly, we define the operators $\circ \coloneqq \int_I \int_\Omega \cdot \,\mathrm{dx}\,\mathrm{dt}$ and $\star \coloneqq \int_\Omega \cdot \,\mathrm{dx}$ to simplify notations. This enables us to rewrite the payoff function $\mathscr{J}$ from Equation (12) as follows

$$\mathscr{J}(\boldsymbol{\Psi},\boldsymbol{\eta}) = \underbrace{\frac{1}{2}\mathscr{E}^\top \circ K_{qq}^{-1} \circ \mathscr{E}}_{e_1(\boldsymbol{\Psi},\boldsymbol{\eta})} + \underbrace{\frac{1}{2}(\boldsymbol{\Psi_0}-\boldsymbol{\Phi_0})^\top \star K_{aa}^{-1} \star (\boldsymbol{\Psi_0}-\boldsymbol{\Phi_0})}_{e_2(\boldsymbol{\Psi},\boldsymbol{\eta})} + \underbrace{\frac{1}{2}(\boldsymbol{d}-\mathscr{H}(\boldsymbol{\Psi}))^\top [K_{\epsilon\epsilon}^{-1}](\boldsymbol{d}-\mathscr{H}(\boldsymbol{\Psi}))}_{e_3(\boldsymbol{\Psi},\boldsymbol{\eta})}. \tag{18}$$

We conclude this section by saying that the purpose of our work is to find $(\boldsymbol{\Psi},\boldsymbol{\eta})$ for each system that minimizes the function $\mathscr{J}$ from Equation (18).

3 Resolution of the inverse problem

Some methods already exist on minimizing the misfit function $\mathscr{J}$ at (18). for instance, we can cite the *representer method* from [Eknes and Evensen, 1997] or the use of genetic algorithmics in [Goldberg, 1989]. While those methods may be used for this problem, they are known to be extremely time-consuming and usually rely on the assumption of no model and initial estimation errors, i.e., which means that $e_1 = e_2 = 0$. The method that we will develop in this work do not rely on such assumptions, but is inspired from [Evensen, 2009], that solves similar systems but for a uniform scalar state model.

3.1 Weak solution of the inverse Problem

Before defining the weak formulation of System (7) and (8), we need to handle the new stochastic terms. In that regard, we define the adjoint variable associated with the state variable $\boldsymbol{\Psi}$ that will be used later on.

Definition 1 (Adjoint variable $\boldsymbol{\lambda}$)**.** *We define the variable* $\boldsymbol{\lambda} = (\lambda_1,\lambda_2)^\top$ *as*

$$\boldsymbol{\lambda} : (t,\boldsymbol{x}) \in I\times\Omega \mapsto K_{qq}^{-1}(t,\cdot,\boldsymbol{x},\cdot)\circ\mathscr{E}(\boldsymbol{\Psi},\boldsymbol{\eta}).$$

Let us remark that λ_1 is the adjoint variable of Ψ_1 (u or A) and λ_2 is the adjoint variable of Ψ_2 (c or B). We now show one of the uses of the adjoint variable $\boldsymbol{\lambda}$ through this lemma.

Lemma 1 (Link between $\boldsymbol{\lambda}$ and $\boldsymbol{\Psi}$)**.** *We have for all* $t\in I$ *and* $\boldsymbol{x}\in\Omega$

$$\begin{aligned} K_{qq}(t,\cdot,\boldsymbol{x},\cdot)\circ\boldsymbol{\lambda} &\coloneqq ((K_{qq}\lambda)_1,(K_{qq}\lambda)_2)^\top \\ &= \int_I\int_\Omega K_{qq}(t,s,\boldsymbol{x},\boldsymbol{y})\boldsymbol{\lambda}(s,\boldsymbol{y})\,\mathrm{dy}\,\mathrm{ds}, \\ &= \mathscr{E}(\boldsymbol{\Psi},\boldsymbol{\eta})(t,\boldsymbol{x}). \end{aligned}$$

Proof. We just need to replace $\boldsymbol{\lambda}$ by its expression

$$\begin{aligned} K_{qq}(t,\cdot,\boldsymbol{x},\cdot)\circ\boldsymbol{\lambda} &= \int_I\int_\Omega K_{qq}(t,s,\boldsymbol{x},\boldsymbol{y})\boldsymbol{\lambda}(s,\boldsymbol{y})\,\mathrm{dy}\,\mathrm{ds}, \\ &= \int_I\int_\Omega K_{qq}(t,s,\boldsymbol{x},\boldsymbol{y})K_{qq}^{-1}(s,\cdot,\boldsymbol{y},\cdot)\circ\mathscr{E}(\boldsymbol{\Psi},\boldsymbol{\eta})\,\mathrm{dy}\,\mathrm{ds}, \\ &= \int_I\int_\Omega\int_I\int_\Omega K_{qq}(t,s,\boldsymbol{x},\boldsymbol{y})K_{qq}^{-1}(s,r,\boldsymbol{y},\boldsymbol{u})\mathscr{E}(\boldsymbol{\Psi},\boldsymbol{\eta})(r,\boldsymbol{u})\,\mathrm{du}\,\mathrm{dr}\,\mathrm{dy}\,\mathrm{ds}, \end{aligned}$$

and then perform the integration over u and s to use the relation between the covariance matrix K_{qq} and its inverse K_{qq}^{-1} given in Equation (10),

$$\begin{aligned} K_{qq}(t,\cdot,\boldsymbol{x},\cdot)\circ\boldsymbol{\lambda} &= \int_I\int_\Omega \delta(t-r)\delta(\boldsymbol{x}-\boldsymbol{y})\mathbb{I}_2\mathscr{E}(\boldsymbol{\Psi},\boldsymbol{\eta})(r,\boldsymbol{y})\,\mathrm{d}y\,\mathrm{d}r, \\ &= \mathscr{E}(\boldsymbol{\Psi},\boldsymbol{\eta})(t,\boldsymbol{x}). \end{aligned}$$

□

Consequently, we can rewrite System (7) as follows

$$\begin{cases} \frac{\partial u}{\partial t} - \nabla\cdot(D_1\nabla u) + \nabla.(D_1\chi u\nabla c) = g_1(u,c,\boldsymbol{\theta}) + (K_{qq}\boldsymbol{\lambda})_1 & ,\forall(t,\boldsymbol{x})\in I\times\Omega \\ \frac{\partial c}{\partial t} - \nabla\cdot(D_2\nabla c) = g_2(u,c,\boldsymbol{\theta}) + (K_{qq}\boldsymbol{\lambda})_2 & ,\forall(t,\boldsymbol{x})\in I\times\Omega \\ u(t=0,\boldsymbol{x}) = u_0(\boldsymbol{x}) + a_{\boldsymbol{x}}^{(1)}, \qquad c(t=0,\boldsymbol{x}) = c_0(\boldsymbol{x}) + a_{\boldsymbol{x}}^{(2)} & ,\forall\boldsymbol{x}\in\Omega \\ D_1\nabla u\cdot\vec{n} - D_1\chi u\nabla c\cdot\vec{n} = 0, \qquad D_2\nabla c\cdot\vec{n} = 0 & ,\forall(t,\boldsymbol{x})\in I\times\partial\Omega, \end{cases} \tag{19}$$

and System (8) as follows

$$\begin{cases} \frac{\partial A}{\partial t} - \nabla^2 A = \mu f_1(A,B) + (K_{qq}\boldsymbol{\lambda})_1 & ,\forall(t,\boldsymbol{x})\in I\times\Omega \\ \frac{\partial B}{\partial t} - d\nabla^2 B = \mu f_2(A,B) + (K_{qq}\boldsymbol{\lambda})_2 & ,\forall(t,\boldsymbol{x})\in I\times\Omega \\ A(t=0,\boldsymbol{x}) = A_0(\boldsymbol{x}) + a_{\boldsymbol{x}}^{(1)}, \qquad B(t=0,\boldsymbol{x}) = B_0(\boldsymbol{x}) + a_{\boldsymbol{x}}^{(2)} & ,\forall\boldsymbol{x}\in\Omega \\ \nabla A\cdot\vec{n} = 0, \qquad \nabla B\cdot\vec{n} = 0 & ,\forall(t,\boldsymbol{x})\in I\times\partial\Omega. \end{cases} \tag{20}$$

With the notation $(K_{qq}\boldsymbol{\lambda})_1$ and $(K_{qq}\boldsymbol{\lambda})_2$ defined in Lemma 1.

Observe that System (19) and (20) do not need the Gaussian process $\boldsymbol{q_{t,x}}$ anymore. At this step, we still have the Gaussian process $\boldsymbol{a_x}$ explicitly in the initial conditions, yet, we will show later that it is not needed. We can now define the variational formulation of System (19) and (20) given the adjoint state $\boldsymbol{\lambda}$ and Lemma 1.

Definition 2 (Weak solution of System (19))**.** *Given $\boldsymbol{\lambda}$ such that $((K_{qq}\lambda)_1,(K_{qq}\lambda)_2)^\top\in(L^2(\Omega))^2$ and the trajectories $a_{\boldsymbol{x}}^{(1)},a_{\boldsymbol{x}}^{(2)}\in L^2(\Omega)$, the weak solution associated with System* (19) *is, for almost every $t\in I$, $u(t,\cdot)\in H^1(\Omega)\cap L^\infty(\Omega)$ and $c(t,\cdot)\in H^1(\Omega)$ with $g_1(u,c,\boldsymbol{\theta})$ and $g_2(u,c,\boldsymbol{\theta})\in L^2(\Omega)$ according to the following variational form*

$$\begin{cases} \dfrac{d}{dt}\displaystyle\int_\Omega uv\,\mathrm{d}x + \int_\Omega D_1\nabla u\cdot\nabla v\,\mathrm{d}x - \int_\Omega \chi D_1 u\nabla c\cdot\nabla v\,\mathrm{d}x = \int_\Omega g_1(u,c,\boldsymbol{\theta})v\,\mathrm{d}x + \int_\Omega (K_{qq}\boldsymbol{\lambda})_1 v\,\mathrm{d}x & ,\forall v\in H^1(\Omega) \\ \dfrac{d}{dt}\displaystyle\int_\Omega cw\,\mathrm{d}x + \int_\Omega D_2\nabla c\cdot\nabla w\,\mathrm{d}x = \int_\Omega g_2(u,c,\boldsymbol{\theta})w\,\mathrm{d}x + \int_\Omega (K_{qq}\boldsymbol{\lambda})_2 w\,\mathrm{d}x & ,\forall w\in H^1(\Omega) \\ u(t=0) = u_0 + a_{\boldsymbol{x}}^{(1)} & ,\forall\boldsymbol{x}\in\Omega \\ c(t=0) = c_0 + a_{\boldsymbol{x}}^{(2)} & ,\forall\boldsymbol{x}\in\Omega \end{cases} \tag{21}$$

The existence of weak solutions and investigating the optimal setting of the spaces are out of the scope of this paper. Mainly, we have chosen those spaces to ensure that every term in Equation (21) has a meaning, and we refer to [Osaki et al., 2002, Bendahmane et al., 2007, Blanchet et al., 2006] for the existence of weak solutions associated with Keller Segel model.

Definition 3 (Weak solution of System (20))**.** *Given $\boldsymbol{\lambda}$ such that $((K_{qq}\lambda)_1,(K_{qq}\lambda)_2)^\top\in(L^2(\Omega))^2$ and the trajectories $a_{\boldsymbol{x}}^{(1)},a_{\boldsymbol{x}}^{(2)}\in L^2(\Omega)$, the weak solution associated with System* (20) *is, for almost every $t\in I$, $A(t,\cdot)\in H^1(\Omega)$ and $B(t,\cdot)\in H^1(\Omega)$ with $f_1(A,B)$, $f_2(A,B)\in L^2(\Omega)$ according to the following*

variational form

$$\begin{cases} \dfrac{d}{dt}\displaystyle\int_\Omega AV\,\mathrm{dx} + \int_\Omega \nabla A\cdot\nabla V\,\mathrm{dx} = \int_\Omega \mu f_1(A,B)V\,\mathrm{dx} + \int_\Omega (K_{qq}\boldsymbol{\lambda})_1 V\,\mathrm{dx} & ,\forall V\in H^1(\Omega) \\ \dfrac{d}{dt}\displaystyle\int_\Omega BW\,\mathrm{dx} + \int_\Omega d\nabla B\cdot\nabla W\,\mathrm{dx} = \int_\Omega \mu f_2(A,B)W\,\mathrm{dx} + \int_\Omega (K_{qq}\boldsymbol{\lambda})_2 W\,\mathrm{dx} & ,\forall W\in H^1(\Omega) \\ A(t=0) = A_0 + a^{(1)}_{\boldsymbol{x}} & ,\forall \boldsymbol{x}\in\Omega \\ B(t=0) = B_0 + a^{(2)}_{\boldsymbol{x}} & ,\forall \boldsymbol{x}\in\Omega \end{cases} \tag{22}$$

3.2 Minimization of $\mathscr{J}$

Let us recall that the main objective is to minimize the function $(\boldsymbol{\Psi},\boldsymbol{\eta})\mapsto \mathscr{J}(\boldsymbol{\Psi},\boldsymbol{\eta})$ given by

$$\mathscr{J}(\boldsymbol{\Psi},\boldsymbol{\eta}) = \frac{1}{2}\mathscr{E}^\top\circ K_{qq}^{-1}\circ\mathscr{E} + \frac{1}{2}(\boldsymbol{\Psi_0}-\boldsymbol{\Phi_0})^\top\star K_{aa}^{-1}\star(\boldsymbol{\Psi_0}-\boldsymbol{\Phi_0}) + \frac{1}{2}(\boldsymbol{d}-\mathscr{H}(\boldsymbol{\Psi}))^\top[\Sigma_{\epsilon\epsilon}^{-1}](\boldsymbol{d}-\mathscr{H}(\boldsymbol{\Psi}))$$

on the space $E \coloneqq (H^1(\Omega))^2\times(\mathbb{R}^+)^{\#\boldsymbol{\eta}}$. This optimization problem has some good properties because $\mathscr{J}$ is a sum of quadratic functions. We know that the function $\mathscr{J}$ has a global minima on E, but it may not be unique due to the nonlinearity of the model, and also $\mathscr{J}$ can possess local minima. A minimum $(\boldsymbol{\Psi}^*,\boldsymbol{\eta}^*)$ of $\mathscr{J}$ on E follows

$$d\mathscr{J}(\boldsymbol{\Psi}^*,\boldsymbol{\eta}^*)\cdot(\delta\boldsymbol{\Psi},\delta\boldsymbol{\eta}) = 0,\ \text{for all directions } (\delta\boldsymbol{\Psi},\delta\boldsymbol{\eta}) \text{ in } E, \tag{23}$$

where $\delta\boldsymbol{\Psi}$ and $\delta\boldsymbol{\eta}$ are the infinitesimal variations of $\boldsymbol{\Psi}$ and $\boldsymbol{\eta}$, respectively. Now, we need to calculate the differential of $\mathscr{J}$. We recall the chain rule for differential functions: the differential of a function h with variables $\boldsymbol{\Psi}$ and $\boldsymbol{\eta}$ for directions $\delta\boldsymbol{\Psi}$ and $\delta\boldsymbol{\eta}$ is given by

$$dh(\boldsymbol{\Psi},\boldsymbol{\eta})\cdot(\delta\boldsymbol{\Psi},\delta\boldsymbol{\eta}) = \frac{\partial h}{\partial\boldsymbol{\Psi}}(\boldsymbol{\Psi},\boldsymbol{\eta})\cdot\delta\boldsymbol{\Psi} + \frac{\partial h}{\partial\boldsymbol{\eta}}(\boldsymbol{\Psi},\boldsymbol{\eta})\cdot\delta\boldsymbol{\eta}. \tag{24}$$

Using the chain rule, the differential of each reaction function from Equation (2) is given by

$$\begin{cases} dg_1(u,c,\boldsymbol{\theta})\cdot(\delta u,\delta c,\delta\boldsymbol{\theta}) &= \begin{bmatrix}-2\rho u+\rho-\delta & 0\end{bmatrix}\begin{bmatrix}\delta u\\ \delta c\end{bmatrix} + \begin{bmatrix}u(1-u) & -u & 0 & 0 & 0\end{bmatrix}\delta\boldsymbol{\theta}, \\ dg_2(u,c,\boldsymbol{\theta})\cdot(\delta u,\delta c,\delta\boldsymbol{\theta}) &= \begin{bmatrix}\alpha-\gamma c & -\beta-\gamma u\end{bmatrix}\begin{bmatrix}\delta u\\ \delta c\end{bmatrix} + \begin{bmatrix}0 & 0 & u & -c & -uc\end{bmatrix}\delta\boldsymbol{\theta}, \end{cases} \tag{25}$$

for System (7), and the ones from Equation (4) are given by

$$\begin{cases} df_1(A,B)\cdot(\delta A,\delta B) &= -\begin{bmatrix}(1+\frac{\partial h}{\partial A}(A,B)) & \frac{\partial h}{\partial B}(A,B)\end{bmatrix}\begin{bmatrix}\delta A\\ \delta B\end{bmatrix}, \\ df_2(A,B)\cdot(\delta A,\delta B) &= -\begin{bmatrix}\frac{\partial h}{\partial A}(A,B) & (k_3+\frac{\partial h}{\partial B}(A,B))\end{bmatrix}\begin{bmatrix}\delta A\\ \delta B\end{bmatrix}, \end{cases} \tag{26}$$

for System (8).

The operator $\mathscr{E}$, defined in Equation (14), is quadratic for the Keller Segel system. Thus, the differential of $\mathscr{E}$ reads to

$$d\mathscr{E}(u,c,\boldsymbol{\theta})\cdot(\delta u,\delta c,\delta\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial\delta u}{\partial t} - \nabla\cdot(D_1\nabla\delta u) + \nabla\cdot(D_1\chi\delta u\nabla c) + \nabla\cdot(D_1\chi u\nabla\delta c) - dg_1(u,c,\boldsymbol{\theta})\cdot(\delta u,\delta c,\delta\boldsymbol{\theta}) \\ \frac{\partial\delta c}{\partial t} - \nabla\cdot(D_2\nabla\delta c) - dg_2(u,c,\boldsymbol{\theta})\cdot(\delta u,\delta c,\delta\boldsymbol{\theta}) \end{pmatrix}, \tag{27}$$

for System (7). For the pattern formation system, due to the linearity of the diffusion terms, one gets

$$d\mathscr{E}(A,B,\boldsymbol{\vartheta})\cdot(\delta A,\delta B,\delta\boldsymbol{\vartheta}) = \begin{pmatrix} \frac{\partial\delta A}{\partial t} - \nabla^2\delta A - \mu df_1(A,B)\cdot(\delta A,\delta B) - \delta\mu f_1(A,B) \\ \frac{\partial\delta B}{\partial t} - d\nabla^2\delta B - \delta d\nabla^2 B - \mu df_2(A,B)\cdot(\delta A,\delta B) - \delta\mu f_2(A,B) \end{pmatrix}. \tag{28}$$

If the measurement is direct, i.e., the function $\mathscr{H}$ is $\mathscr{H}(\boldsymbol{\Psi}(t,\cdot)) = (\boldsymbol{\Psi}(t,\boldsymbol{x_j}))_{1\leq j\leq m}$, then the differential of $\mathscr{H}$ reads

$$\begin{aligned} d\mathscr{H}(\boldsymbol{\Psi})\cdot(\delta\boldsymbol{\Psi}(t,\cdot)) &= (\delta\boldsymbol{\Psi}(t,\boldsymbol{x_j}))_{1\leq j\leq m}, \\ &= \int_I\int_\Omega \delta\boldsymbol{\Psi}(s,\boldsymbol{y}) \times \underbrace{(\delta(t-s)\delta(\boldsymbol{x_j}-\boldsymbol{y})_{1\leq j\leq m})}_{\coloneqq\mathscr{H}(\delta_t)}\,\mathrm{ds}\,\mathrm{dy}, \end{aligned}$$

where $\times$ denotes the usual Cartesian product. This enables us to rewrite the differential of $\mathscr{H}$ in an integral form with the notation

$$\delta\boldsymbol{\Psi}\circ\mathscr{H}(\delta_t) \coloneqq \int_I\int_\Omega \delta\boldsymbol{\Psi}(s,u)\times\mathscr{H}(\delta_t)(s,u)\,\mathrm{ds}\,\mathrm{du}\,. \tag{29}$$

Equation (29) is crucial for our methodology because to solve the inverse problem, we rely on the reformulation of the differential of $\mathscr{J}$ in the integral form.

Proposition 1 (Differential of $\mathscr{J}$)**.** *The differential of the misfit function $\mathscr{J}$ is*

$$\begin{aligned} d\mathscr{J}(\boldsymbol{\Psi},\boldsymbol{\eta})\cdot(\delta\boldsymbol{\Psi},\delta\boldsymbol{\eta}) &= d\mathscr{E}(\boldsymbol{\Psi},\boldsymbol{\eta})^\top\cdot(\delta\boldsymbol{\Psi},\delta\boldsymbol{\eta})\circ\boldsymbol{\lambda} \\ &\quad + \delta\boldsymbol{\Psi}^\top(t=0)\star K_{aa}^{-1}\star(\boldsymbol{\Psi_0}-\boldsymbol{\Phi_0}) - \delta\boldsymbol{\Psi}^\top\circ\mathscr{H}(\delta_t)[\Sigma_{\epsilon\epsilon}^{-1}](d-\mathscr{H}(\boldsymbol{\Psi})), \end{aligned} \tag{30}$$

where $\boldsymbol{\lambda}$ is associated to $\boldsymbol{\Psi}$ and $\boldsymbol{\eta}$ as stated in Definition 1.

Proof. From Equation (18), the differential of $\mathscr{J}$ is the sum of the differential of e_1, e_2 and e_3,

$$d\mathscr{J}(\boldsymbol{\Psi},\boldsymbol{\eta})\cdot(\delta\boldsymbol{\Psi},\delta\boldsymbol{\eta}) = de_1(\boldsymbol{\Psi},\boldsymbol{\eta})\cdot(\delta\boldsymbol{\Psi},\delta\boldsymbol{\eta}) + de_2(\boldsymbol{\Psi},\boldsymbol{\eta})\cdot(\delta\boldsymbol{\Psi},\delta\boldsymbol{\eta}) + de_3(\boldsymbol{\Psi},\boldsymbol{\eta})\cdot(\delta\boldsymbol{\Psi},\delta\boldsymbol{\eta}),$$

where the functions e_1, e_2 and e_3 are quadratic. Therefore the differential of the sum of $e_1(\boldsymbol{\Psi},\boldsymbol{\eta})$, $e_2(\boldsymbol{\Psi},\boldsymbol{\eta})$ and $e_3(\boldsymbol{\Psi},\boldsymbol{\eta})$ is given by

$$\underbrace{d\mathscr{E}(\boldsymbol{\Psi},\boldsymbol{\eta})^\top\cdot(\delta\boldsymbol{\Psi},\delta\boldsymbol{\eta})\circ K_{qq}^{-1}\circ\mathscr{E}(\boldsymbol{\Psi},\boldsymbol{\eta})}_{de_1(\boldsymbol{\Psi},\boldsymbol{\eta})\cdot(\delta\boldsymbol{\Psi},\delta\boldsymbol{\eta})} + \underbrace{\delta\boldsymbol{\Psi}_0^\top\star K_{aa}^{-1}\star(\boldsymbol{\Psi_0}-\boldsymbol{\Phi_0})}_{de_2(\boldsymbol{\Psi},\boldsymbol{\eta})\cdot(\delta\boldsymbol{\Psi},\delta\boldsymbol{\eta})} - \underbrace{d\mathscr{H}(\boldsymbol{\Psi})^\top\cdot(\delta\boldsymbol{\Psi})[\Sigma_{\epsilon\epsilon}^{-1}](\boldsymbol{d}-\mathscr{H}(\boldsymbol{\Psi}))}_{de_3(\boldsymbol{\Psi},\boldsymbol{\eta})\cdot(\delta\boldsymbol{\Psi},\delta\boldsymbol{\eta})}.$$

With $\boldsymbol{\lambda}$ associated with $\boldsymbol{\Psi}$ and $\boldsymbol{\eta}$ at Definition 1, the differential of e_1 can be rewritten using Lemma 1 as

$$de_1(\boldsymbol{\Psi},\boldsymbol{\eta})\cdot(\delta\boldsymbol{\Psi},\delta\boldsymbol{\eta}) = d\mathscr{E}(\boldsymbol{\Psi},\boldsymbol{\eta})^\top\cdot(\delta\boldsymbol{\Psi},\delta\boldsymbol{\eta})\circ\boldsymbol{\lambda}.$$

Finally, we use the notation introduced in Equation (29) in the differential of e_3 to conclude the proof of the proposition. □

Looking $\boldsymbol{\lambda}$ as the adjoint variable, we observe that it appears explicitly when finding the minimum of $\mathscr{J}$ on E. The main difficulty is that the writing of $\boldsymbol{\lambda}$ depends only on the state vector $\boldsymbol{\Psi}$. However, denoting $(\boldsymbol{\Psi}^*,\boldsymbol{\eta}^*)$ be a minimum of $\mathscr{J}$, we will explore another way to define $\boldsymbol{\lambda}$. The adjoint variable $\boldsymbol{\lambda}^*$ corresponding to $(\boldsymbol{\Psi}^*,\boldsymbol{\eta}^*)$, according to Definition 1, comes as the solution of a system of PDEs. The idea is to detail the adjoint system to get an easier expression of $\boldsymbol{\lambda}^*$.

Definition 4 (Weak solution of $\boldsymbol{\lambda}$ associated to System (19))**.** *Given $\nabla c\in(L^2(\Omega))^d$, $u\in L^\infty(\Omega)$, $\frac{\partial}{\partial u}g_1(u,c,\boldsymbol{\theta})\in L^\infty(\Omega)$, $\frac{\partial}{\partial c}g_1(u,c,\boldsymbol{\theta})\in L^\infty(\Omega)$, $\frac{\partial}{\partial u}g_2(u,c,\boldsymbol{\theta})\in L^\infty(\Omega)$, $\frac{\partial}{\partial c}g_2(u,c,\boldsymbol{\theta})\in L^\infty(\Omega)$ and $\mathscr{C}(u,c) = (\mathscr{C}(u,c)_1,\mathscr{C}(u,c)_2)^\top \coloneqq (1,1)^\top\circ\mathscr{H}(\delta)[\Sigma_{\epsilon\epsilon}^{-1}](d-\mathscr{H}(u,c))\in(L^2(\Omega))^2$. We define the following variational problem: for almost every $t\in I$, find $\boldsymbol{\lambda}(t,\cdot) = (\lambda_1(t,\cdot),\lambda_2(t,\cdot))^\top\in(H^1(\Omega))^2$ such*

that

$$\begin{cases} \dfrac{d}{dt}\displaystyle\int_\Omega \lambda_1 v\,\mathrm{dx} + \int_\Omega D_1\nabla\lambda_1\cdot\nabla v\,\mathrm{dx} - \int_\Omega D_1\chi\nabla c\cdot\nabla\lambda_1 v\,\mathrm{dx} & \\ \qquad = -\displaystyle\int_\Omega \frac{\partial g_1}{\partial u}\lambda_1 v\,\mathrm{dx} - \int_\Omega \frac{\partial g_1}{\partial c}\lambda_2 v\,\mathrm{dx} - \int_\Omega \mathscr{C}(u,c)_1 v\,\mathrm{dx} & ,\forall v\in H^1(\Omega)\cap L^\infty(\Omega) \\ \dfrac{d}{dt}\displaystyle\int_\Omega \lambda_2 w\,\mathrm{dx} + \int_\Omega D_2\nabla\lambda_2\cdot\nabla w\,\mathrm{dx} - \int_\Omega D_1\chi u\nabla\lambda_1\cdot\nabla w\,\mathrm{dx} & \\ \qquad = -\displaystyle\int_\Omega \frac{\partial g_2}{\partial u}\lambda_1 w\,\mathrm{dx} - \int_\Omega \frac{\partial g_2}{\partial c}\lambda_2 w\,\mathrm{dx} - \int_\Omega \mathscr{C}(u,c)_2 w\,\mathrm{dx} & ,\forall w\in H^1(\Omega) \\ \lambda_1(t=t_f,\boldsymbol{x})=0, \qquad \lambda_2(t=t_f,\boldsymbol{x})=0 & ,\forall \boldsymbol{x}\in\Omega. \end{cases}$$

Similarly in Definition 2, the function spaces used in Definition 4 are not optimal. The setting is so that every term in the above weak form has a sense. If $\boldsymbol{\lambda}(t,\cdot)=(\lambda_1(t,\cdot),\lambda_2(t,\cdot))^\top$ is a weak solution associated to System (19), according to Definition 4, then we have

$$\begin{cases} \frac{\partial\lambda_1}{\partial t} - \nabla\cdot(D_1\nabla\lambda_1) - D_1\chi\nabla c\cdot\nabla\lambda_1 = -\frac{\partial g_1}{\partial u}\lambda_1 - \frac{\partial g_1}{\partial c}\lambda_2 - \mathscr{C}(u,c)_1 & ,a.e.(t,\boldsymbol{x})\in I\times\Omega \\ \frac{\partial\lambda_2}{\partial t} - \nabla\cdot(D_2\nabla\lambda_2) + \nabla\cdot(D_1\chi u\nabla\lambda_1) = -\frac{\partial g_2}{\partial u}\lambda_1 - \frac{\partial g_2}{\partial c}\lambda_2 - \mathscr{C}(u,c)_2 & ,a.e.(t,\boldsymbol{x})\in I\times\Omega \\ \lambda_1(t=t_f,\boldsymbol{x})=\lambda_2(t=t_f,\boldsymbol{x})=0 & ,a.e.(\boldsymbol{x})\in\Omega \\ (D_1\nabla\lambda_1 - D_1\chi\lambda_1\nabla c)\cdot\vec{n} = (D_2\nabla\lambda_2 - D_1\chi u\nabla\lambda_1)\cdot\vec{n} = 0 & ,a.e.(t,\boldsymbol{x})\in I\times\partial\Omega. \end{cases} \tag{31}$$

Definition 5 (Weak solution of $\boldsymbol{\lambda}$ associated to System (20))**.** *Given $\frac{\partial}{\partial A}f_1(A,B)\in L^\infty(\Omega)$, $\frac{\partial}{\partial B}f_1(A,B)\in L^\infty(\Omega)$, $\frac{\partial}{\partial A}f_2(A,B)\in L^\infty(\Omega)$, $\frac{\partial}{\partial B}f_2(A,B)\in L^\infty(\Omega)$ and $\mathscr{C}(A,B)=(\mathscr{C}(A,B)_1,\mathscr{C}(A,B)_2)^\top = (1,1)^\top\circ\mathscr{H}(\delta)[\Sigma_{\epsilon\epsilon}^{-1}](d-\mathscr{H}(A,B))\in(L^2(\Omega))^2$. We define the following variational problem: for almost every $t\in I$, find $\boldsymbol{\lambda}(t,\cdot)=(\lambda_1(t,\cdot),\lambda_2(t,\cdot))^\top\in(H^1(\Omega))^2$ such that*

$$\begin{cases} \dfrac{d}{dt}\displaystyle\int_\Omega \lambda_1 V\,\mathrm{dx} + \int_\Omega \nabla\lambda_1\cdot\nabla V\,\mathrm{dx} & \\ \qquad = -\displaystyle\int_\Omega \mu\frac{\partial f_1}{\partial A}\lambda_1 V\,\mathrm{dx} - \int_\Omega \mu\frac{\partial f_1}{\partial B}\lambda_2 V\,\mathrm{dx} - \int_\Omega \mathscr{C}(A,B)_1 V\,\mathrm{dx} & ,\forall V\in H^1(\Omega) \\ \dfrac{d}{dt}\displaystyle\int_\Omega \lambda_2 W\,\mathrm{dx} + \int_\Omega d\nabla\lambda_2\cdot\nabla W\,\mathrm{dx} & \\ \qquad = -\displaystyle\int_\Omega \mu\frac{\partial f_2}{\partial A}\lambda_1 W\,\mathrm{dx} - \int_\Omega \mu\frac{\partial f_2}{\partial B}\lambda_2 W\,\mathrm{dx} - \int_\Omega \mathscr{C}(A,B)_2 W\,\mathrm{dx} & ,\forall W\in H^1(\Omega) \\ \lambda_1(t=t_f,\boldsymbol{x})=0, \qquad \lambda_2(t=t_f,\boldsymbol{x})=0 & ,\forall\boldsymbol{x}\in\Omega. \end{cases}$$

If $\boldsymbol{\lambda}(t,\cdot)=(\lambda_1(t,\cdot),\lambda_2(t,\cdot))^\top$ is a weak solution associated to System (20), according to Definition 5, then we have

$$\begin{cases} \frac{\partial\lambda_1}{\partial t} - \nabla^2\lambda_1 = -\mu\frac{\partial f_1}{\partial A}\lambda_1 - \mu\frac{\partial f_1}{\partial B}\lambda_2 - \mathscr{C}(A,B)_1 & ,a.e.(t,\boldsymbol{x})\in I\times\Omega \\ \frac{\partial\lambda_2}{\partial t} - d\nabla^2\lambda_2 = -\mu\frac{\partial f_2}{\partial A}\lambda_1 - \mu\frac{\partial f_2}{\partial B}\lambda_2 - \mathscr{C}(A,B)_2 & ,a.e.(t,\boldsymbol{x})\in I\times\Omega \\ \lambda_1(t=t_f,\boldsymbol{x})=\lambda_2(t=t_f,\boldsymbol{x})=0 & ,a.e.(\boldsymbol{x})\in\Omega \\ \nabla\lambda_1\cdot\vec{n}=\nabla\lambda_2\cdot\vec{n}=0 & ,a.e.(t,\boldsymbol{x})\in I\times\partial\Omega. \end{cases} \tag{32}$$

Proposition 2 (Solution of the inverse problem)**.** *Let $(\boldsymbol{\Psi}^*,\boldsymbol{\eta}^*)$ be a minimum of $\mathscr{J}$ and associate with $\boldsymbol{\lambda}^*$ that is given in Definition 1, a solution of System* (19) *(resp. System* (20)*) is a weak solution $(\boldsymbol{\Psi}^*,\boldsymbol{\eta}^*)$ as defined by Definition 4 (resp. Definition 5), and the triplet $(\boldsymbol{\eta}^*,\boldsymbol{\Psi}^*,\boldsymbol{\lambda}^*)$ cancels out the vector function $\mathscr{G}(\boldsymbol{\eta}^*,\boldsymbol{\Psi}^*,\boldsymbol{\lambda}^*)$ defined in Equation* (KS 3)*(resp. $\mathscr{F}(\boldsymbol{\eta}^*,\boldsymbol{\Psi}^*,\boldsymbol{\lambda}^*)$ defined in Equation*

(PF 3)*). This can be summarized to* $(u^*, c^*, \boldsymbol{\lambda}^*, \boldsymbol{\theta}^*)$ *being a solution of this coupled system*

$$\begin{cases} \frac{\partial u}{\partial t} - \nabla \cdot (D_1 \nabla u) + \nabla.(D_1 \chi u \nabla c) = g_1(u, c, \boldsymbol{\theta}) + (K_{qq}\boldsymbol{\lambda})_1 & ,\forall (t, \boldsymbol{x}) \in I \times \Omega, \\ \frac{\partial c}{\partial t} - \nabla \cdot (D_2 \nabla c) = g_2(u, c, \boldsymbol{\theta}) + (K_{qq}\boldsymbol{\lambda})_2 & ,\forall (t, \boldsymbol{x}) \in I \times \Omega, \\ u(t=0, \boldsymbol{x}) = u_0(\boldsymbol{x}) + (K_{aa}\boldsymbol{\lambda}^0)_1, c(t=0, \boldsymbol{x}) = c_0(\boldsymbol{x}) + (K_{aa}\boldsymbol{\lambda}^0)_2 & ,\forall \boldsymbol{x} \in \Omega, \\ D_1 \nabla u \cdot \vec{n} - D_1 \chi u \nabla c \cdot \vec{n} = 0, D_2 \nabla c \cdot \vec{n} = 0 & ,\forall (t, \boldsymbol{x}) \in I \times \partial\Omega, \end{cases} \tag{KS 1}$$

$$\begin{cases} \frac{\partial \lambda_1}{\partial t} - \nabla \cdot (D_1 \nabla \lambda_1) - D_1 \chi \nabla c \cdot \nabla \lambda_1 = -\frac{\partial g_1}{\partial u}\lambda_1 - \frac{\partial g_1}{\partial c}\lambda_2 - \mathscr{C}(u, c)_1 & ,a.e.(t, \boldsymbol{x}) \in I \times \Omega, \\ \frac{\partial \lambda_2}{\partial t} - \nabla \cdot (D_2 \nabla \lambda_2) + \nabla \cdot (D_1 \chi u \nabla \lambda_1) = -\frac{\partial g_2}{\partial u}\lambda_1 - \frac{\partial g_2}{\partial c}\lambda_2 - \mathscr{C}(u, c)_2 & ,a.e.(t, \boldsymbol{x}) \in I \times \Omega, \\ \boldsymbol{\lambda}(t = t_f, \boldsymbol{x}) = 0 & ,a.e. \boldsymbol{x} \in \Omega, \\ (D_1 \nabla \lambda_1 - D_1 \chi \lambda_1 \nabla c) \cdot \vec{n} = (D_2 \nabla \lambda_2 - D_1 \chi u \nabla \lambda_1) \cdot \vec{n} = 0 & ,a.e.(t, \boldsymbol{x}) \in I \times \partial\Omega, \end{cases} \tag{KS 2}$$

$$\mathscr{G}(\boldsymbol{\theta}, u, c, \boldsymbol{\lambda}) = \int_I \int_\Omega \begin{bmatrix} \frac{\partial g_1}{\partial \rho} & \frac{\partial g_1}{\partial \delta} & \frac{\partial g_1}{\partial \alpha} & \frac{\partial g_1}{\partial \beta} & \frac{\partial g_1}{\partial \gamma} \\ \frac{\partial g_2}{\partial \rho} & \frac{\partial g_2}{\partial \delta} & \frac{\partial g_2}{\partial \alpha} & \frac{\partial g_2}{\partial \beta} & \frac{\partial g_2}{\partial \gamma} \end{bmatrix}^\top \cdot \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \mathrm{dt}\,\mathrm{dx} = 0_{\mathbb{R}^5}, \tag{KS 3}$$

and to $(A^*, B^*, \boldsymbol{\lambda}^*, \boldsymbol{\vartheta}^*)$ *being a solution of this coupled system*

$$\begin{cases} \frac{\partial A}{\partial t} - \nabla^2 A = \mu f_1(A, B) + (K_{qq}\lambda)_1 & ,\forall (t, \boldsymbol{x}) \in I \times \Omega, \\ \frac{\partial B}{\partial t} - d\nabla^2 B = \mu f_2(A, B) + (K_{qq}\lambda)_2 & ,\forall (t, \boldsymbol{x}) \in I \times \Omega, \\ A(t=0, \boldsymbol{x}) = A_0(\boldsymbol{x}) + (K_{aa}\boldsymbol{\lambda}^0)_1, B(t=0, \boldsymbol{x}) = B_0(\boldsymbol{x}) + (K_{aa}\boldsymbol{\lambda}^0)_2 & ,\forall \boldsymbol{x} \in \Omega, \\ \nabla A \cdot \vec{n} = 0, \nabla B \cdot \vec{n} = 0 & ,\forall (t, \boldsymbol{x}) \in I \times \partial\Omega, \end{cases} \tag{PF 1}$$

$$\begin{cases} \frac{\partial \lambda_1}{\partial t} - \nabla^2 \lambda_1 = -\mu \frac{\partial f_1}{\partial A}\lambda_1 - \mu \frac{\partial f_1}{\partial B}\lambda_2 - \mathscr{C}(A, B)_1 & ,a.e.(t, \boldsymbol{x}) \in I \times \Omega, \\ \frac{\partial \lambda_2}{\partial t} - d\nabla^2 \lambda_2 = -\mu \frac{\partial f_2}{\partial A}\lambda_1 - \mu \frac{\partial f_2}{\partial B}\lambda_2 - \mathscr{C}(A, B)_2 & ,a.e.(t, \boldsymbol{x}) \in I \times \Omega, \\ \lambda_1(t = t_f, \boldsymbol{x}) = \lambda_2(t = t_f, \boldsymbol{x}) = 0 & ,a.e.(\boldsymbol{x}) \in \Omega, \\ \nabla \lambda_1 \cdot \vec{n} = \nabla \lambda_2 \cdot \vec{n} = 0 & ,a.e.(t, \boldsymbol{x}) \in I \times \partial\Omega, \end{cases} \tag{PF 2}$$

$$\mathscr{F}(\boldsymbol{\vartheta}, A, B, \boldsymbol{\lambda}) = \int_I \int_\Omega \begin{bmatrix} 0 & f_1(A, B) \\ \nabla^2 B & f_2(A, B) \end{bmatrix}^\top \cdot \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \mathrm{dt}\,\mathrm{dx} = 0_{\mathbb{R}^2}, \tag{PF 3}$$

where $\left((K_{aa}\boldsymbol{\lambda}^0)_1, (K_{aa}\boldsymbol{\lambda}^0)_2\right)^\top \coloneqq K_{aa}(\boldsymbol{x}, \cdot) \star \boldsymbol{\lambda}(t_0, \cdot)$.

Proof. We will expose the proof of the proposition for the Keller Segel system, the same reasoning leads to the result for the pattern formation system, will be omitted for sake of simplicity.

The result comes from writing the condition in Equation (23) with the differential of the misfit function $\mathscr{J}$ stated in Equation (30), which gives

$$\underbrace{d\mathscr{E}^\top(\boldsymbol{\Psi}^*, \boldsymbol{\eta}^*) \cdot (\delta\boldsymbol{\Psi}, \delta\boldsymbol{\eta}) \circ \boldsymbol{\lambda}^*}_{\mathscr{M}_1} + \underbrace{\delta\boldsymbol{\Psi}_{\mathbf{0}}^{*\top} \star K_{aa}^{-1} \star (\boldsymbol{\Psi}_0^* - \boldsymbol{\Phi}_{\mathbf{0}})}_{\mathscr{M}_2} - \underbrace{\delta\boldsymbol{\Psi}^{*\top} \circ \mathscr{H}(\delta)\Sigma_{\epsilon\epsilon}^{-1}(d - \mathscr{H}(\boldsymbol{\Psi}^*))}_{\mathscr{M}_3} = 0, \forall (\delta\boldsymbol{\Psi}, \boldsymbol{\eta}) \in E.$$

We extend the integrals over space and time and drop the asterisk notation for readability, to get

$$\begin{aligned} \mathscr{M}_{11} = \int_I \int_\Omega (&\frac{\partial \delta u}{\partial t} - \nabla \cdot (D_1 \nabla \delta u) + \nabla \cdot (D_1 \chi \delta u \nabla c) \\ &+ \nabla \cdot (D_1 \chi u \nabla \delta c) - dg_1(u, c, \boldsymbol{\theta}) \cdot (\delta u, \delta c, \delta\boldsymbol{\theta}))\lambda_1 \,\mathrm{dx}\,\mathrm{dt}, \\ \mathscr{M}_{12} = \int_I \int_\Omega &\left(\frac{\partial \delta c}{\partial t} - \nabla \cdot (D_2 \nabla \delta c) - dg_2(u, c, \boldsymbol{\theta}) \cdot (\delta u, \delta c, \delta\boldsymbol{\theta})\right) \lambda_2 \,\mathrm{dx}\,\mathrm{dt}, \end{aligned}$$

when developing $\mathscr{M}_1 = (\mathscr{M}_{11}, \mathscr{M}_{12})^\top$. Then, we get

$$\mathscr{M}_2 = \int_\Omega \delta\boldsymbol{\Psi}_{\mathbf{0}}^\top \left(K_{aa}^{-1} \star (\boldsymbol{\Psi}_{\mathbf{0}} - \boldsymbol{\Phi}_{\mathbf{0}})\right) \mathrm{dx},$$

and

$$\mathscr{M}_3 = \int_I \int_\Omega \delta\mathbf{\Psi}(t,\boldsymbol{x}) \times \mathscr{H}(\delta_t)[\Sigma_{\epsilon\epsilon}^{-1}](d - \mathscr{H}(\mathbf{\Psi}))\,\mathrm{d}t\,\mathrm{d}x\,.$$

The main idea now is to factorize each term by $\delta\mathbf{\Psi}$ or $\delta\boldsymbol{\eta}$ and remove every differential operators on $\delta\mathbf{\Psi}$ and $\delta\boldsymbol{\eta}$. This is done by performing integration by parts, in time and space when needed, and using the weak formulations of the state variable and the adjoint variable equations to cancel out some terms. This leads to independent conditions to come up with the system governing $\boldsymbol{\lambda}$. We have

$$\begin{aligned}
\mathscr{M}_{11} = &\underbrace{\int_I \int_\Omega \frac{\partial \delta u}{\partial t}\lambda_1\,\mathrm{d}x\,\mathrm{d}t}_{\mathscr{M}_{11}^1} - \underbrace{\int_I \int_\Omega \nabla\cdot(D_1\nabla\delta u)\lambda_1\,\mathrm{d}x\,\mathrm{d}t}_{\mathscr{M}_{11}^2} \\
&+ \underbrace{\int_I \int_\Omega \nabla\cdot(D_1\chi\delta u\nabla c)\lambda_1\,\mathrm{d}x\,\mathrm{d}t}_{\mathscr{M}_{11}^3} + \underbrace{\int_I \int_\Omega \nabla\cdot(D_1\chi u\nabla\delta c)\lambda_1\,\mathrm{d}x\,\mathrm{d}t}_{\mathscr{M}_{11}^4} \\
&- \underbrace{\int_I \int_\Omega \begin{bmatrix}-2\rho u + \rho - \delta & 0\end{bmatrix}\begin{bmatrix}\delta u \\ \delta c\end{bmatrix}\lambda_1\,\mathrm{d}x\,\mathrm{d}t}_{\mathscr{M}_{11}^5} - \underbrace{\int_I \int_\Omega \begin{bmatrix}u(1-u) & -u & 0 & 0 & 0\end{bmatrix}\delta\boldsymbol{\theta}\lambda_1\,\mathrm{d}x\,\mathrm{d}t}_{\mathscr{M}_{11}^6}.
\end{aligned}$$

The terms $\mathscr{M}_{11}^5$ and $\mathscr{M}_{11}^6$ are in the integral form that we expect, we need to modify the other ones due to the presence of derivatives on δu or δc.

We perform an integration by parts in time to handle $\mathscr{M}_{11}^1$ which gives

$$\mathscr{M}_{11}^1 = \int_\Omega \delta u(t_f)\lambda_1(t_f)\,\mathrm{d}x - \int_\Omega \delta u(t_0)\lambda_1(t_0)\,\mathrm{d}x - \int_I \int_\Omega \frac{\partial\lambda_1}{\partial t}\delta u\,\mathrm{d}x\,\mathrm{d}t\,.$$

Finally, we integrate two times by parts in space the terms $\mathscr{M}_{11}^2$, $\mathscr{M}_{11}^3$ and $\mathscr{M}_{11}^4$ to write

$$\begin{aligned}
\mathscr{M}_{11}^2 &= \int_I \int_\Omega \nabla\cdot(D_1\nabla\delta u\lambda_1)\,\mathrm{d}x\,\mathrm{d}t - \int_I \int_\Omega D_1\nabla\delta u\cdot\nabla\lambda_1\,\mathrm{d}x\,\mathrm{d}t, \\
&= \int_I \int_{\partial\Omega} D_1\nabla\delta u\cdot\vec{n}\lambda_1\,\mathrm{d}x\,\mathrm{d}t + \int_I \int_\Omega \nabla\cdot(D_1\nabla\lambda_1)\delta u\,\mathrm{d}x\,\mathrm{d}t - \int_I \int_{\partial\Omega} D_1\nabla\lambda_1\cdot\vec{n}\delta u\,\mathrm{d}x\,\mathrm{d}t, \\
\mathscr{M}_{11}^3 &= \int_I \int_\Omega \nabla\cdot(D_1\chi\delta u\nabla c\lambda_1)\,\mathrm{d}x\,\mathrm{d}t - \int_I \int_\Omega D_1\chi\delta u\nabla c\cdot\nabla\lambda_1\,\mathrm{d}x\,\mathrm{d}t, \\
&= \int_I \int_{\partial\Omega} D_1\chi\nabla c\cdot\vec{n}\lambda_1\delta u\,\mathrm{d}x\,\mathrm{d}t - \int_I \int_\Omega D_1\chi\delta u\nabla c\cdot\nabla\lambda_1\,\mathrm{d}x\,\mathrm{d}t, \\
\mathscr{M}_{11}^4 &= \int_I \int_\Omega \nabla\cdot(D_1\chi u\nabla\delta c\lambda_1)\,\mathrm{d}x\,\mathrm{d}t - \int_I \int_\Omega D_1\chi u\nabla\delta c\cdot\nabla\lambda_1\,\mathrm{d}x\,\mathrm{d}t, \\
&= \int_I \int_{\partial\Omega} D_1\chi u\lambda_1\nabla\delta c\cdot\vec{n}\,\mathrm{d}x\,\mathrm{d}t + \int_I \int_\Omega \nabla\cdot(D_1\chi u\nabla\lambda_1)\delta c\,\mathrm{d}x\,\mathrm{d}t - \int_I \int_{\partial\Omega} D_1\chi u\nabla\lambda_1\cdot\vec{n}\delta c\,\mathrm{d}x\,\mathrm{d}t\,.
\end{aligned}$$

Thanks to the weak formulation of the Keller Segel system from Equation (21), we have

$$\int_I \int_{\partial\Omega} D_1\nabla\delta u\cdot\vec{n}\lambda_1\,\mathrm{d}x\,\mathrm{d}t - \int_I \int_{\partial\Omega} D_1\chi u\lambda_1\nabla\delta c\cdot\vec{n}\,\mathrm{d}x\,\mathrm{d}t = 0, \tag{33}$$

which implies that each term in $\mathscr{M}_{11}$ is well factorized. With similar reasoning, we can show that

$$\begin{aligned}
\mathscr{M}_{12} = &\int_\Omega \delta c(t_f)\lambda_2(t_f)\,\mathrm{d}x - \int_\Omega \delta c(t_0)\lambda_2(t_0)\,\mathrm{d}x - \int_I \int_\Omega \frac{\partial\lambda_2}{\partial t}\delta c\,\mathrm{d}x\,\mathrm{d}t \\
&- \int_I \int_\Omega \nabla\cdot(D_2\nabla\lambda_2)\delta c\,\mathrm{d}x\,\mathrm{d}t + \int_I \int_{\partial\Omega} D_2\nabla\lambda_2\cdot\vec{n}\delta c\,\mathrm{d}x\,\mathrm{d}t \\
&- \int_I \int_\Omega \begin{bmatrix}\alpha - \gamma c & -\beta - \gamma u\end{bmatrix}\begin{bmatrix}\delta u \\ \delta c\end{bmatrix}\lambda_2\,\mathrm{d}x\,\mathrm{d}t - \int_I \int_\Omega \begin{bmatrix}0 & 0 & u & -c & -uc\end{bmatrix}\delta\boldsymbol{\theta}\lambda_2\,\mathrm{d}x\,\mathrm{d}t\,.
\end{aligned}$$

Finally, we can gather every term previously given with their specific factorization. For both equations on λ_1 and λ_2, we have the factorization on the space solution at time t_f (34) and the space solution at time t_0 (35)

$$\int_\Omega \delta\mathbf{\Psi}(t_f)^\top \boldsymbol{\lambda}^*(t_f) = 0, \qquad \forall \delta\mathbf{\Psi}(t_f), \tag{34}$$

$$\int_\Omega \delta\mathbf{\Psi}(t_0)^\top (K_{aa}^{-1} \star (\mathbf{\Psi}_0^\star - \mathbf{\Phi_0}) - \boldsymbol{\lambda}^*(t_0)) = 0, \qquad \forall \delta\mathbf{\Psi}(t_0). \tag{35}$$

To complete, we have the boundary conditions over time,

$$\int_{\partial\Omega}\int_I \delta u\,((D_1\nabla\lambda_1 - D_1\chi\lambda_1\nabla c)\cdot\vec{n}) = 0, \forall \delta u \text{ on } I\times\partial\Omega, \tag{36}$$

$$\int_{\partial\Omega}\int_I \delta c\,((D_2\nabla\lambda_2 - D_1\chi u\nabla\lambda_1)\cdot\vec{n}) = 0, \forall \delta c \text{ on } I\times\partial\Omega, \tag{37}$$

the partial differential system over time and space followed by λ_1 and λ_2

$$\int_I\int_\Omega \delta u\left(\frac{\partial\lambda_1}{\partial t} - \nabla\cdot(D_1\nabla\lambda_1) - D_1\chi\nabla c\cdot\nabla\lambda_1 - \frac{\partial g_1}{\partial u}\lambda_1 - \frac{\partial g_1}{\partial c}\lambda_2 - \mathscr{C}(u,c)_1\right) = 0, \forall \delta u \text{ on } I\times\Omega, \tag{38}$$

$$\int_I\int_\Omega \delta c\left(\frac{\partial\lambda_2}{\partial t} - \nabla\cdot(D_2\nabla\lambda_2) + \nabla\cdot(D_1\chi u\nabla\lambda_1) - \frac{\partial g_2}{\partial u}\lambda_1 - \frac{\partial g_2}{\partial c}\lambda_2 - \mathscr{C}(u,c)_2\right) = 0, \forall \delta c \text{ on } I\times\Omega. \tag{39}$$

Moreover, we have the factorization by $\delta\boldsymbol{\theta}$

$$\int_I\int_\Omega (\lambda_1\frac{\partial g_1}{\partial\boldsymbol{\theta}} + \lambda_2\frac{\partial g_2}{\partial\boldsymbol{\theta}})^\top \,\mathrm{dx}\,\mathrm{dt}\,\delta\boldsymbol{\theta} = 0, \forall\delta\boldsymbol{\theta}. \tag{40}$$

Since the equations (34), (36), (37), (38) and (39) hold for all δu and δc on the appropriate spaces, we conclude that λ_1 and λ_2 are a solution of System (KS 2).

Equation (35) leads to

$$K_{aa}^{-1}\star(\mathbf{\Psi_0} - \mathbf{\Phi_0}) - \boldsymbol{\lambda}(t_0) = 0, \text{ on } \Omega,$$

which, by multiplying by $K_{aa}\star$, is equivalent to

$$\mathbf{\Psi_0} = \mathbf{\Phi_0} + K_{aa}\star\boldsymbol{\lambda}(t_0), \text{ on } \Omega.$$

This implies that $\mathbf{\Psi}$ is a solution of (KS 1).

Lastly, Equation (40) leads to

$$\int_I\int_\Omega (\lambda_1\frac{\partial g_1}{\partial\boldsymbol{\theta}} + \lambda_2\frac{\partial g_2}{\partial\boldsymbol{\theta}})^\top \,\mathrm{dx}\,\mathrm{dt} = 0,$$

that is, $(\mathbf{\Psi}, \boldsymbol{\theta}, \boldsymbol{\lambda})$ follows Equation (KS 3). □

We conclude by stating that the solution of our inverse problem is also the solution of the System (KS 1)-(KS 3), for the Keller-Segel problem, and (PF 1)-(PF 3) for the Pattern Formation system. What remain, is to show the numerical solvability of those systems. In the next section, we address this issue.

4 Numerical resolution of the inverse problem

4.1 Numerical scheme

In this section, we address the numerical approximations of System (KS 1)-(KS 3) and System (PF 1)-(PF 3), that are nonlinear with explicit and implicit forms of the state variables and of the input

parameters in each equation. Then applying classical Newton methods to solve numerically System (KS 1)-(KS 3) or System (PF 1)-(PF 3) is unlikely to achieve convergence due to the complexity and the dimensionality of the discrete associated system.

Here, we propose an efficient strategy for the numerical approximation of the systems. The main idea is to decouple the unknown variables and use an iterative method to come up with a numerical solution. Specifically, we start by fixing a value of $\boldsymbol{\eta}$, then we get the value of $\boldsymbol{\Psi}(\boldsymbol{\eta})$ and $\boldsymbol{\lambda}(\boldsymbol{\eta})$, associated with $\boldsymbol{\eta}$, using the iterative method described in Algorithm 4.2. Once we get $(\boldsymbol{\Psi}(\boldsymbol{\eta}), \boldsymbol{\lambda}(\boldsymbol{\eta}), \boldsymbol{\eta})$, we give a better estimation of $\boldsymbol{\eta}$ by using a projected Newton-Raphson method with the function $\mathscr{G}$ for System (KS 1)-(KS 3) or the function $\mathscr{F}$ for System (PF 1)-(PF 3). With this new estimation of $\boldsymbol{\eta}$, we iterate the previous process until $\boldsymbol{\eta}$ is well calibrated as described in Algorithm 4.1.

Algorithm 4.1: Projected Newton-Raphson method to calibrate the coefficients in $\boldsymbol{\eta}$.

Data: l_b, u_b, C_{ineq}, b_{ineq}, the function $\mathscr{F}$ (resp.$\mathscr{G}$), $0 < h \ll 1$
input : $\boldsymbol{\eta}^0 \leftarrow$ An initial guess of $\boldsymbol{\eta}$
output: The final guess of $\boldsymbol{\eta}$, minimizing $\mathscr{J}$ and solving (KS 3) or (PF 3)

Check that $\boldsymbol{\eta}^0 \in K = \{\boldsymbol{x} \in \mathbb{R}^{\#\boldsymbol{\eta}} : l_b \leq \boldsymbol{x} \leq u_b \text{ and } C_{ineq}x \leq b_{ineq}\}$;
$m \leftarrow 0$;
while $(\boldsymbol{\eta}^k)_{k\geq 0}$ *has not converged* **do**
  /* Approximating the Jacobian matrix of $\mathscr{F}$ around $\boldsymbol{\eta}^k$ */
  for $i, j \leftarrow 1$ **to** $\#\boldsymbol{\eta}$ **do**
    $\tilde{Jac}[i,j] \leftarrow e_i^\top(\frac{\mathscr{F}(\boldsymbol{\eta}^k + h \times e_j) - \mathscr{F}(\boldsymbol{\eta}^k)}{h})$; /* $e_i = [\delta_{ij}]_{1\leq j\leq \#\boldsymbol{\eta}}$ */
    /* $\mathscr{F}(\boldsymbol{\eta})$ is calculated using Algorithm 4.2 to get $\boldsymbol{\Psi}(\boldsymbol{\eta})$ and $\boldsymbol{\lambda}(\boldsymbol{\eta})$ */
  end
  /* Projected Newton-Raphson method */
  $\tilde{\boldsymbol{\eta}}^{k+1}$ solves $\tilde{Jac} \times (\tilde{\boldsymbol{\eta}}^{k+1} - \boldsymbol{\eta}^k) = -\mathscr{F}(\boldsymbol{\eta}^k)$; /* Newton-Raphson method */
  $\boldsymbol{\eta}^{k+1} \leftarrow p_K(\tilde{\boldsymbol{\eta}}^{k+1})$; /* projection of $\tilde{\boldsymbol{\eta}}^{k+1}$ on K */
  $k \leftarrow k + 1$;
end
$\boldsymbol{\eta} \leftarrow \boldsymbol{\eta}^k$;

Algorithm 4.2: Iterative method to get $\boldsymbol{\Psi}$ and $\boldsymbol{\lambda}$ associated with a value of $\boldsymbol{\eta}$.

input : $\boldsymbol{\eta} \leftarrow$ A set of coefficients
output: $\boldsymbol{\Psi}, \boldsymbol{\lambda} \leftarrow$ Numerical approximation of $\boldsymbol{\Psi}$ and $\boldsymbol{\lambda}$ associated with $\boldsymbol{\eta}$

/* Initialization */
$\boldsymbol{\Psi}^0$ solves *System* (1) *or* (3);
$m \leftarrow 0$; /* m is the number of iteration of the process */
/* We decouple $\boldsymbol{\Psi}$ and $\boldsymbol{\lambda}$ from System (KS 1)-(KS 3) or (PF 1)-(PF 3) */
while $(\boldsymbol{\Psi}^m)_{m\geq 0}$ *and* $(\boldsymbol{\lambda}^m)_{m\geq 0}$ *has not converged* **do**
  $\boldsymbol{\lambda}^m$ solves *System* (KS 2) *or* (PF 2) *with* $\boldsymbol{\Psi} = \boldsymbol{\Psi}^m$; /* see Algorithm A */
  $\boldsymbol{\Psi}^{m+1}$ solves *System* (KS 1) *or* (PF 1) *with* $\boldsymbol{\lambda} = \boldsymbol{\lambda}^m$; /* see Algorithm B */
  $m \leftarrow m + 1$;
end
$(\boldsymbol{\Psi}, \boldsymbol{\lambda}) \leftarrow (\boldsymbol{\Psi}^m, \boldsymbol{\lambda}^m)$;

In Algorithm 4.2, iterations of $\boldsymbol{\Psi}^m$ and $\boldsymbol{\lambda}^m$ are performed using a Finite Difference as detailed in Appendix A and B in Algorithm (A.1), (A.2), (B.1), and (B.2). Convergence of $(\boldsymbol{\Psi}^{(m)}, \boldsymbol{\lambda}^{(m)})_{m\geq 0}$, while being not provided in this work, is observed in practise.

To help the search of $\boldsymbol{\eta}$, it is usual to assume some prior-knowledge on the feasible values of it. Then we suppose that there is a lower and upper bound value for each coefficient, i.e., there exists $l_b > 0 \in \mathbb{R}^{\#\boldsymbol{\eta}}$ and $u_b > 0 \in \mathbb{R}^{\#\boldsymbol{\eta}}$ such that $l_b \leq \boldsymbol{\eta} \leq u_b$ and a positive number of N_c linear inequality

constraints, i.e., there exists $C_{ineq} \in \mathbb{R}^{N_c \times \#\boldsymbol{\eta}}$ and $b_{ineq} \in \mathbb{R}^{N_c}$ such that $C_{ineq} \times \boldsymbol{\eta} \leq b_{ineq}$.

Then, we can define the constraint space as follows

$$K = \{\boldsymbol{\eta} \in \mathbb{R}^5 : l_b \leq \boldsymbol{\eta} \leq u_b \text{ and } C_{ineq}\boldsymbol{\eta} \leq b_{ineq}\},$$

and its associated orthogonal projector is denoted by p_K.

4.2 Example of numerical resolutions

In this section, we show one example of numerical results for the Keller-Segel System and the Pattern Formation System. The purpose is to showcase how the numerical resolution performs in term of accuracy and convergence of the calibrated parameters. Although there are a lot of hyperparameters including in the continuous and discrete settings, they will be tuned in this section. Their study and impact analysis will be deferred until Section 4.3.

4.2.1 Keller-Segel problem

Because there is an increase of the tumour cells and the O_2 concentration over time for the Keller-Segel system, we deal with $\rho > \delta$ and $\alpha > \beta$, that is, $C_{ineq} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$ and $b_{ineq} = \begin{bmatrix} -1e-8 \\ -1e-8 \end{bmatrix}$.

For our test, we choose $l_b = [0.07, 0.06, 0.07, 0.009, 0.05]$ and $u_b = [0.6, 0.5, 0.4, 0.06, 0.2]$ to bound the search of $\boldsymbol{\theta}$, where the vector of the true parameters is $\boldsymbol{\theta_{ex}} = [0.2; 0.1; 0.1; 0.03; 0.08]$.

Our numerical simulations are performed on a uniform mesh with $N = 100$ points on the unit interval $[0, 1]$, and for time variables varying from $t_0 = 0$ to $t_f = 7.1$, with an uniform time-step $\Delta t = 0.01$. We consider measurement at each space point ($m = N$) and every hundred time steps which gives $N_{mea} = 8$. The initial conditions for u and c are

$$\begin{cases} u_0(x) & = \exp\left(1 - \frac{1}{1-y(x)^2}\right) \times \mathbb{1}_{[0.5,0.7]}(x), \qquad y(x) = \frac{x-0.6}{0.1}, \\ c_0(x) & = 0.5 \times u_0(\boldsymbol{x}). \end{cases} \tag{41}$$

We deal with the following values for the diffusion coefficients $D_1 = 5 \cdot 10^{-5}$, $D_2 = 1 \cdot 10^{-3}$ and for the chemotaxis coefficient $\chi = 1 \cdot 10^{1}$. The covariance matrices are given by

$$\begin{cases} \Sigma_{\epsilon\epsilon} & = \sigma_\epsilon^2 \times I_{N_{mea}}, \\ K_{aa}(x, y) & = \sigma_a^2 \times \delta(x - y) \times I_2, \\ K_{qq}(t, x, s, y) & = \sigma_q^2 \times \min(t, s) \times \delta(x - y) \times I_2, \end{cases} \tag{42}$$

where the standard deviation errors σ_ϵ, σ_a and σ_q are given by $\sigma_\epsilon^2 = (10^{-\frac{1}{2}})^2$, $\sigma_a^2 = 10^{-2}$ and $(\sigma_q)^2 = (10^{-2})^2$. In practise, since we are more likely to make an error in the power of 10, we numerically calibrate $\log_{10}(\boldsymbol{\theta})$ rather than $\boldsymbol{\theta}$ itself.

We start by performing simulation of the forward problem with an initial guess $\boldsymbol{\theta}_0 \approx [0.2864, 0.0647, 0.3075, 0.0529, 0.128$ that is given by the uniform law on the constraint space K. In Figure 1, we can see the values of $\Psi_1 = u$ and $\Psi_2 = c$ associated with that first guess. Those can be compared with their exact values obtained with $\boldsymbol{\theta}_{exact}$, and the values of λ_1 and λ_2 that implicitly highlights the error between $\boldsymbol{\Psi}$ and $\boldsymbol{\Psi_{exact}}$. We observe that $\boldsymbol{\theta}_0$ is not the right value for the inverse problem. Indeed, we can visually see difference of shape between u and u_{exact}, and more clearly between c and c_{exact}. This is also reported in the value of λ_1 and λ_2 through the maximum amplitude of $\boldsymbol{\lambda}$, which is here, around 0.1.

The last guess given by Algorithm 4.1 is displayed in Figure 2. Here, there is no visual difference between (u, c) and their exact values, which suggests that the value of $\boldsymbol{\theta}$ is correct. This is strengthened by the estimated magnitude of $\boldsymbol{\lambda}$ being low, around 10^{-8}.

In our example, Algorithm 4.1 has converged after 36 iterations. Those iterations are displayed in Figure 3. We can see that the coefficients do converge to their exact values. The convergence of some coefficient can be faster than other as we can see from Figure 3 that the parameters ρ and δ converge within 10 iterations.

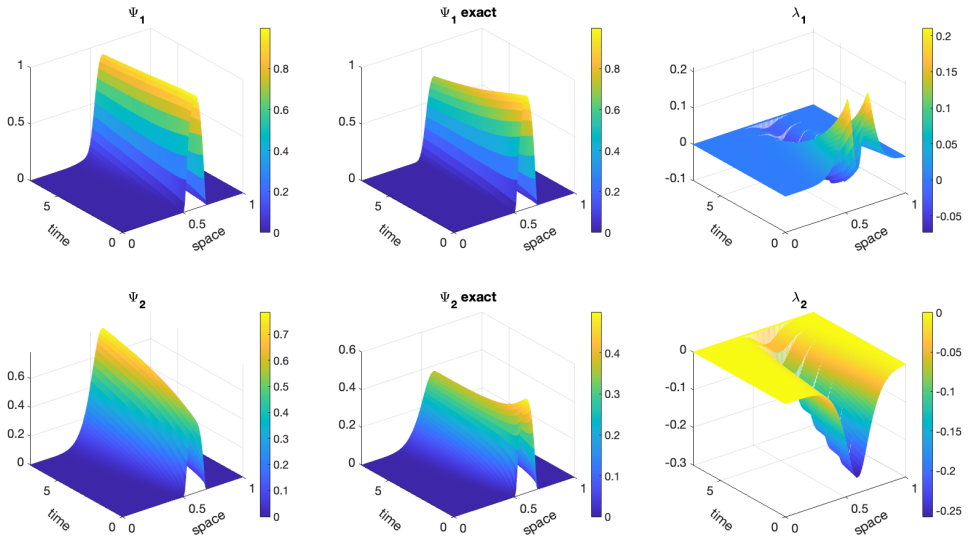

Figure 1: Solutions of the Keller-Segel System without solving the inverse problem. In the left column, we have the numerical solution of System (1) associated with $\boldsymbol{\theta}^0 \approx [0.2864, 0.0647, 0.3075, 0.0529, 0.1281]$, compared to the expected solution in the middle column obtained with $\boldsymbol{\theta} = \boldsymbol{\theta_{ex}}$. In the last column, we have the adjoint variable of the poorly calibrated state vector.

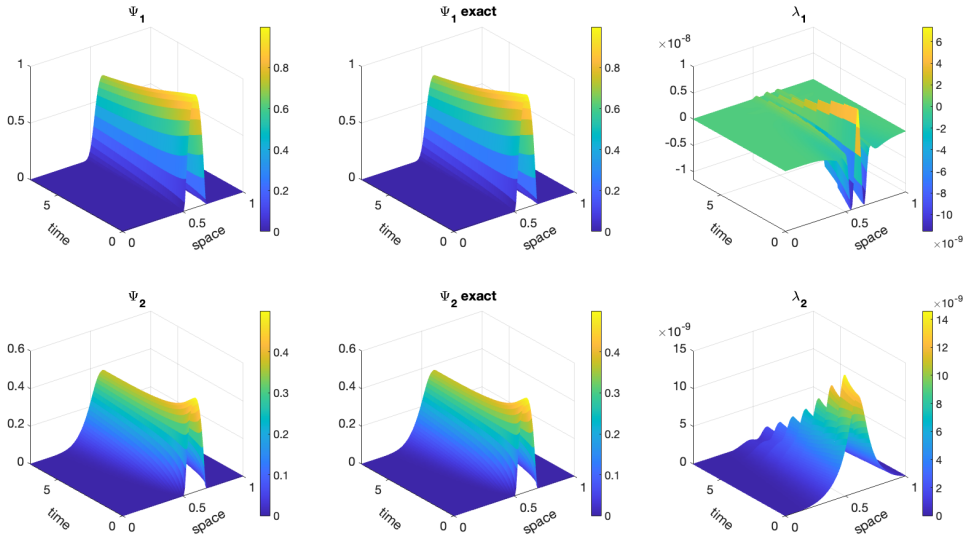

Figure 2: Solution of the Keller-Segel inverse problem solving System (KS 1)-(KS 3) starting with the first guess $\boldsymbol{\theta}^0$. In the left column, we have the final state vector predicted, compared to the exact state vector in the middle column. In the last column, we have the adjoint variable associated to the calibrated state vector.

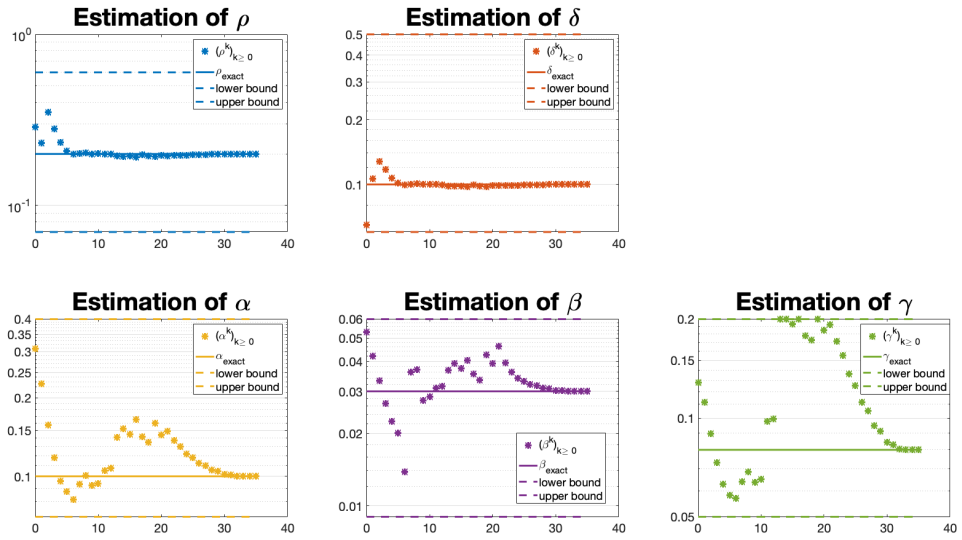

Figure 3: $(\boldsymbol{\theta}^k)_{k\geq 0}$: calibrations of $\boldsymbol{\theta}$ in terms of the number of iterations, during the numerical process from Algorithm 4.1.

4.2.2 Pattern Formation problem

For the Pattern Formation system, there is no prior-knowledge in the relationship between d and μ. For our test, we choose $l_b = [15, 25]$ and $u_b = [25, 35]$ to delineate the search area of $\boldsymbol{\vartheta}$, with the true parameters being $\boldsymbol{\vartheta}_{ex} = [20; 30]$. Here also, the numerical simulations are performed in a uniform mesh with $N = 100$ points on the unit interval $[0, 1]$, and from the initial time $t_0 = 0$ to the terminal time $t_f = 2$ with an uniform time-step $\Delta t = 0.001$. We also use measurement at each point ($m = N$), and every hundred time steps, that gives $N_{mea} = 21$. The initial space conditions for A and B are given by $f_1(A, B) = f_2(A, B) = 0$ with some additional noise, given by

$$\begin{cases} A_0(x) &= 9.9338 + 0.01 \times \cos(\pi x), \\ B_0(x) &= 9.2892 + 0.01 \times \mathscr{N}(0, 1)(x). \end{cases}$$

We fix the coefficients $(k_i)_{i=1,\cdots,5}$ as $k_1 = 92$, $k_2 = 64$, $k_3 = 1.5$, $k_4 = 18.5$ and $k_5 = 0.1$ and we consider the covariance matrices given by

$$\begin{cases} \Sigma_{\epsilon\epsilon} &= \sigma_\epsilon^2 \times I_{N_{mea}}, \\ K_{aa}(x, y) &= \sigma_a^2 \times \delta(x - y) \times I_2, \\ K_{qq}(t, x, s, y) &= \sigma_q^2 \times \min(t, s) \times \delta(x - y) \times I_2, \end{cases} \tag{43}$$

where the standard deviation errors σ_ϵ, σ_a and σ_q are given by $\sigma_\epsilon^2 = (10^{-1})^2$, $\sigma_a^2 = (10^{-4})^2$ and $\sigma_q^2 = (10^{-4})^2$. Here the dimensionality of the parameter of interest $\boldsymbol{\vartheta}$ drops from five to two, that speed up the numerical resolution compared to the Keller-Segel model. This enables further investigations on the applicability of this technique to calibrate coefficients.

An example of calibration starts with the first guess $(d_0, \mu_0) \approx (23.23, 26.75)$, that is given by a uniform law on the constraint space K. The state estimation associated with this first guess is depicted in Figure 4. At first glance, the values of A and B seem to be close to their exact values. However, they are mismatched as highlighted by the magnitude of $\boldsymbol{\lambda}$ around 0.1. The final estimation leads to the values of A and B given in Figure 5 and look correct visually, according to the exact values. Yet, the magnitude of $\boldsymbol{\lambda}$ is around 10^{-5}. In this example, convergence was observed after eight calibrations to the exact coefficients. The values of $\boldsymbol{\vartheta}$ in terms of the number of iterations through the process are represented in Figure 6.

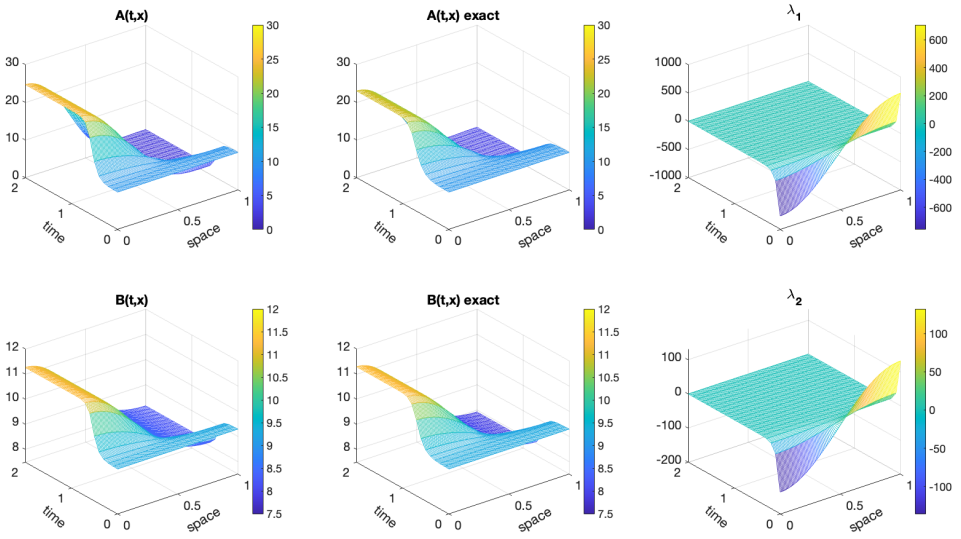

Figure 4: Solutions of the Pattern Formation System without solving the inverse problem. In the left column, we have the numerical solution of System (3) associated with $\boldsymbol{\vartheta}^0 = [23.23; 26.75]$, compared to the expected solution in the middle column obtained with $\boldsymbol{\vartheta} = \boldsymbol{\vartheta}_{ex}$. In the last column, we have the adjoint variable of the poorly calibrated state vector.

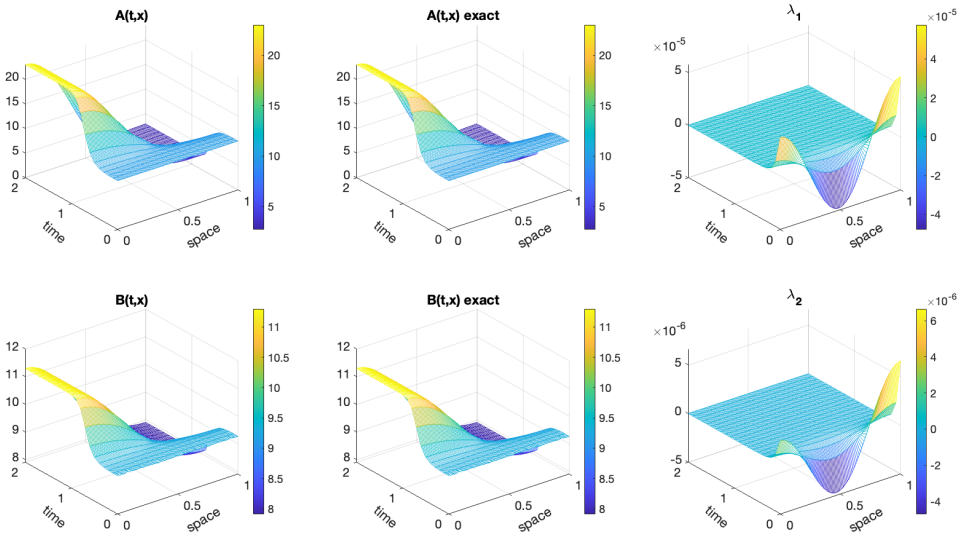

Figure 5: Solution of the Keller-Segel inverse problem solving System (PF 1)-(PF 3) starting with the first guess $\boldsymbol{\vartheta}^0$. In the left column, we have the final state vector predicted, compared to the exact state vector in the middle column. In the last column, we have the adjoint variable associated to the calibrated state vector.

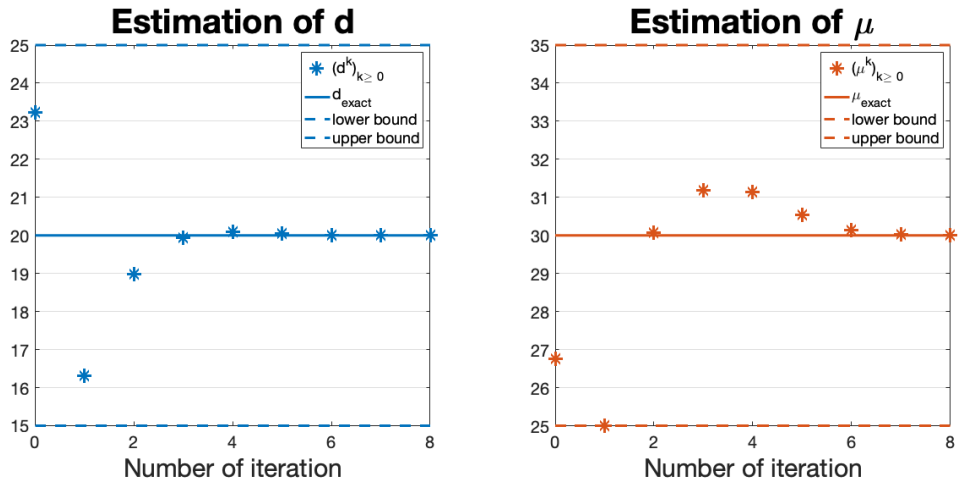

Figure 6: $(\boldsymbol{\vartheta}^k)_{k\geq 0}$: calibrations of $\boldsymbol{\vartheta}$ in terms of the number of iterations, during the numerical process from Algorithm 4.1.

4.3 Impact of the hyperparameters on the numerical resolution

The performance of the numerical simulation in terms of accuracy and convergence of the numerical solution depends on the choice of the hyperparameters that are fixed prior to the numerical approximation. We give here the list of numerical hyperparameters can affect the numerical resolution in practise. We have

- the choice of the covariance matrices $\Sigma_{\epsilon\epsilon}$, K_{aa} and K_{qq}: while other expression can be chosen, we can first study the impact of the values of σ_ϵ, σ_a and σ_q as defined in Equations (42) and (43).

- the amount of data available during the measurement process: we distinguish the number of measurements N_{mea} extracted and the number of data m that is extracted at each observation. N_{mea} is directly linked to the time between two consecutive measurements Δm that is supposed to be constant during the process.

- the presence or absence of the modelling uncertainty: there is error inherent to the modelling process that would be characterized by the Gaussian process $\boldsymbol{q_{t,x}}$, or the lack of complete knowledge about some coefficients in the model. If the error is in the initial condition, the characterization is thorough the Gaussian noise $\boldsymbol{a_x}$ and the Gaussian error $\boldsymbol{\epsilon}$ if the error is from the measurement process.

- the range of the constraint space K use to restrain the search of the coefficients. This works by choosing l_b and u_b.

We investigate the impact of each of the identified hyperparameters. For the sake of simplicity, we present only results for the Pattern Formation System due to the fast forward simulation time. We perform 21 different simulations where each one is dedicated to the analysis of the impact of a single hyperparameter. Each simulation starts with the same first guess $\boldsymbol{\vartheta}^0$ and the same initial condition (A_0, B_0) as chosen in the previous section. The set up of those simulations are detailed in Table 2 in Appendix C.

To study the impact of hyperparameters, we report in Table 1 the errors on the substrate oxygen concentration $||A - A_{ex}||_{L^\infty(t_0,t_f,L^2(\Omega))}$, on the enzyme uricase concentration $||B - B_{ex}||_{L^\infty(t_0,t_f,L^2(\Omega))}$, on the adjoint variable of A $||\lambda_1||_{L^\infty(t_0,t_f,L^2(\Omega))}$, on the adjoint variable of B $||\lambda_2||_{L^\infty(t_0,t_f,L^2(\Omega))}$, the number of iterations used in Algorithm (4.1) to calibrate $\boldsymbol{\eta}$, and the error on the vector of coefficients $|\boldsymbol{\eta} - \boldsymbol{\eta_{ex}}|_\infty$ for each simulation as described in Table 2.

Table 1: Quantities of interest for every simulation described in Table 2.

	$\|\|A - A_{ex}\|\|$ $L^\infty(I, L^2(\Omega))$	$\|\|B - B_{ex}\|\|$ $L^\infty(I, L^2(\Omega))$	$\|\|\lambda_1\|\|$ $L^\infty(I, L^2(\Omega))$	$\|\|\lambda_2\|\|$ $L^\infty(I, L^2(\Omega))$	calibrations	$\|\boldsymbol{\eta} - \boldsymbol{\eta_{ex}}\|_\infty$
1	$4.9931\ 10^{-4}$	$9.3711\ 10^{-5}$	0.1427	0.0263	5	0.0019
2	$1.6887\ 10^{-5}$	$1.9228\ 10^{-6}$	0.5232	0.0930	5	$6.9554\ 10^{-5}$
3	$4.9977\ 10^{-4}$	$9.3187\ 10^{-5}$	0.0014	$2.6345\ 10^{-4}$	5	0.0018
4	$1.6513\ 10^{-5}$	$1.9279\ 10^{-6}$	0.0051	$9.1024\ 10^{-4}$	5	$6.3318\ 10^{-5}$
5	$5.0180\ 10^{-4}$	$9.2631\ 10^{-5}$	0.1444	0.0265	5	0.0017
6	$4.9102\ 10^{-4}$	$9.3452\ 10^{-5}$	0.1395	0.0257	5	0.0020
7	$4.9248\ 10^{-4}$	$9.3843\ 10^{-5}$	0.1399	0.0258	5	0.0020
8	6.2518	0.8789	$6.2589\ 10^{+3}$	572.6649	2	5.0000
9	$1.3992\ 10^{-4}$	$3.3861\ 10^{-5}$	0.0337	0.0050	10	0.0019
10	$2.7143\ 10^{-4}$	$4.7258\ 10^{-5}$	0.0251	0.0040	5	0.0025
11	$3.7811\ 10^{-5}$	$8.8072\ 10^{-6}$	$3.7620\ 10^{-4}$	$6.8078\ 10^{-5}$	5	$4.9928\ 10^{-4}$
12	4.6437	0.5457	0.5151	0.0908	3	5.0000
13	$3.2559\ 10^{-5}$	$9.1306\ 10^{-6}$	0.0031	$5.7046\ 10^{-4}$	5	$4.1640\ 10^{-4}$
14	0.3846	0.1743	80.4624	8.1984	5	0.6745
15	0.0959	0.0442	12.5195	1.3404	5	0.1884
16	$8.0603\ 10^{-4}$	$1.1286\ 10^{-4}$	0.1606	0.0170	5	0.0029
17	0.0065	$9.5257\ 10^{-4}$	1.3231	0.1363	5	0.0170
18	0.0355	0.0049	3.6368	0.3718	5	0.0670
19	0.1611	0.0402	3.8503	0.3581	4	2.0268
20	$4.1736\ 10^{-4}$	$8.6558\ 10^{-5}$	0.0715	0.0123	6	0.0026
21	$6.3127\ 10^{-6}$	$1.5385\ 10^{-6}$	$5.2645\ 10^{-4}$	$7.6921\ 10^{-5}$	8	$8.4519\ 10^{-5}$

4.3.1 Influence of σ_ϵ

Simulations 1, 2 and 3 only differ according to the value σ_ϵ. We can observe the impact of that hyperparameter in Figure 7. This hyperparameter does not impact the number of calibrations required to get $\boldsymbol{\eta}$, because it only affects the magnitude of $\boldsymbol{\lambda}$. Indeed, when the value of σ_ϵ decreases, the magnitude of $\boldsymbol{\lambda}$ increases. This comes from the PDE governing the dynamic of $\boldsymbol{\lambda}$ in System (PF 2) as the term $\mathscr{C}(A, B)$ is linear in σ_ϵ^{-1}.

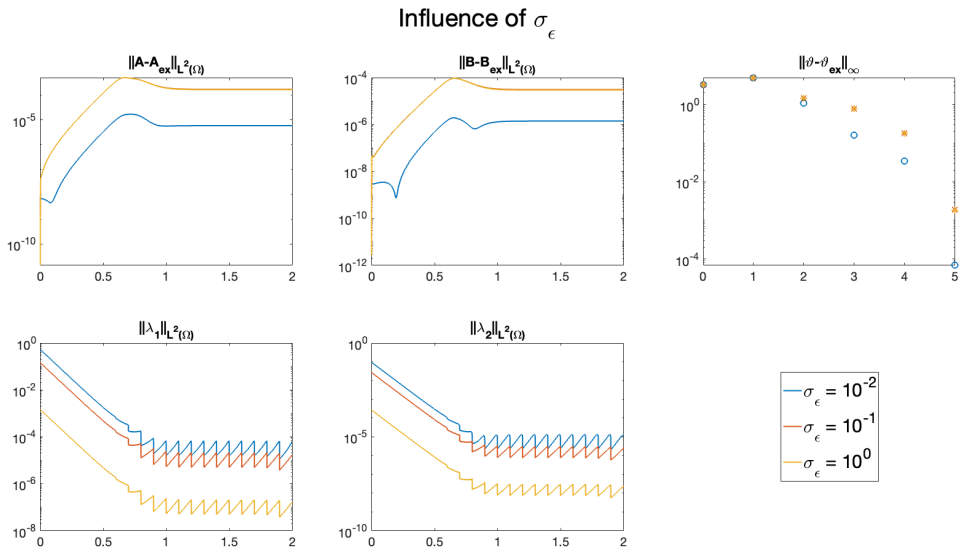

Figure 7: Comparison of errors in time in Simulations 1, 2 and 3 associated with the values $\sigma_\epsilon = 10^{-2}, 10^{-1}, 10^{0}$ of the standard deviation, respectively. The yellow and red curves, while being distinct, are really close for the state variable errors (first and second plots in the first line).

4.3.2 Influence of σ_a

Simulations 1, 4 and 5 only differ according to the value of the standard deviation σ_a. The impact of that hyperparameter is observed in Figure 14, from which, we see that the hyperparameter does not impact the number of calibrations required to get $\boldsymbol{\eta}$, but it impacts the importance of fitting correctly the initial condition. Indeed, in Table 1, the values of $||A - A_{ex}(t_0, \cdot)||_{L^2(\Omega)}$ and $||B - B_{ex}(t_0, \cdot)||_{L^2(\Omega)}$ decrease as σ_a decreases.

4.3.3 Influence of σ_q

The value of the standard deviation σ_q is the only difference in the simulations 1, 6 and 7. From Table 1, we observe that the hyperparameter does not impact at all the numerical resolution. This comes from the fact that we use synthetic data, i.e., measurements are produced directly from the model with the correct coefficients in $\boldsymbol{\eta}$. Thus, there is no modelling error in the process, which implies that the calibrations in $\boldsymbol{\eta}$ fit easily the model.

However, we expect this hyperparameter to have an importance when using real data where the underlying model can be inaccurate.

4.3.4 Influence of the time between measurements

The only difference in simulations 1, 8 and 9 is the time step between two consecutive measurements Δm. The effects of the hyperparameter are presented in Figure 8. That hyperparameter impacts directly the efficiency of the calibration in Algorithm 4.1. Indeed, if we have insufficient measurements (Simulation 8 with $\Delta m = 200\Delta t$) the calibration fails because the vector of coefficients $\boldsymbol{\eta}$ is evaluated at the boundary of the constraint space K. In that case, there is a need of large enough number of measurements to get a correct calibration of $\boldsymbol{\eta}$. However, having too many measurements is counter-productive because more iterations in the calibration are requested when $\Delta m = 50\Delta t$ than the case with $\Delta m = 100\Delta t$, despite both calibrations are correct.

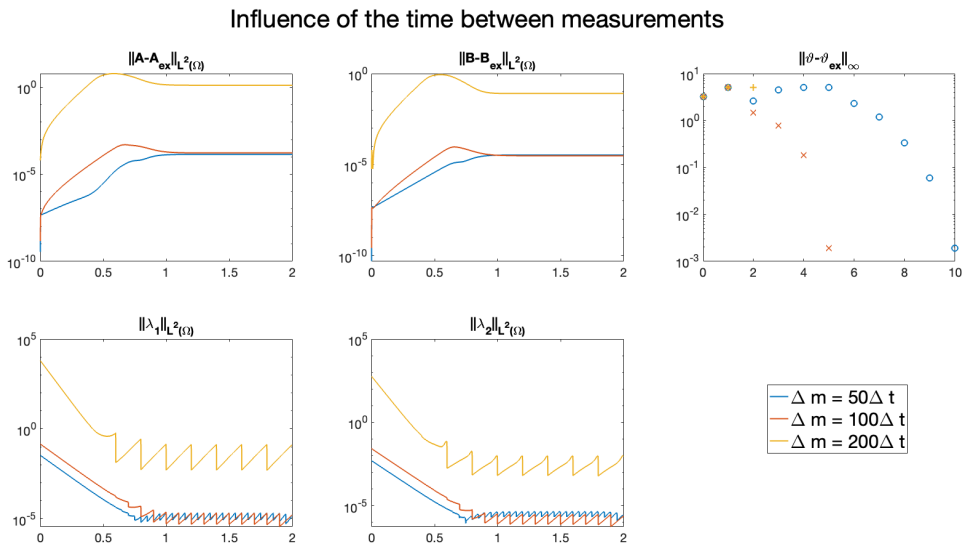

Figure 8: Evolution of the errors in time between simulations 1, 8 and 9 according to the hyperparameter Δm.

4.3.5 Influence of m, the dimensionality of an observation

In simulations 1, 10, 11 and 12 the only parameter that is changing is the number m of data points of each measurement. We observe in Figure 9 the effects of that hyperparameter on the errors of the state variables A and B, the error of the parameter of interest, and the approximation of the adjoint variables. As Δm, this hyperparameter directly impacts the efficiency of the calibration. Having not enough data points (Simulation 12 with $m = 1$) leads to a poor calibration of $\boldsymbol{\eta}$, while too many data points can be counterproductive even though the calibration is correct because the magnitude of $\boldsymbol{\lambda}$ increases as m increases.

It is worth mentioning that Δm and m do not have the same role in the calibration. Indeed, when analysing the calibration according to the total volume data (linear with $\frac{m}{\Delta m}$), we see that some calibrations can not work despite having a high number of data. For example, Simulations 8 and 10 have the same amount of data, but calibration in Simulation 8 fails, while it performs well in Simulation 10.

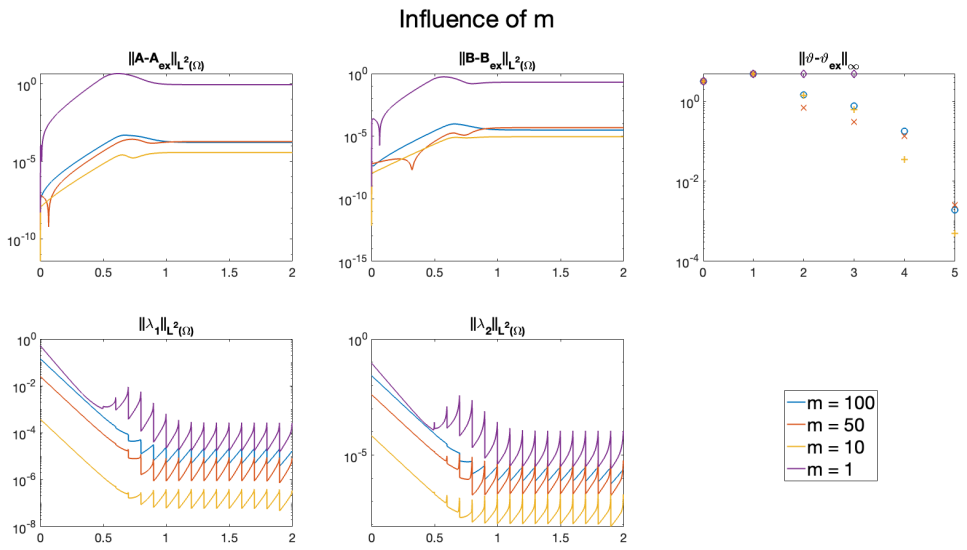

Figure 9: Comparison of results between Simulation 1, 10, 11 and 12.

4.3.6 Influence of the modelling errors

Simulations 1, 13, 14 and 15 differ according to the error in the modelling choices. The impact of that hyperparameter is observed in Figure 10. We consider two distinct sources of error in the modelling: one is the additive Gaussian noise (following a Wiener process) to pollute the exact solution (Simulation 13), the second one consists in changing the value of k_4 for the exact solution. Those two type of errors do not impact the numerical resolution in the same way. Indeed, adding a Gaussian noise does not impact the calibration while modifying the value of k_4 leads to a calibration that gives coefficients different to the expected ones: the further away k_4 is from its right value, the further away the calibration is to the expected coefficients.

In practise, there is a risk that the modelling errors can not be spotted looking at the adjoint variable $\boldsymbol{\lambda}$ (see the visual impact in Figure 13 as comparison). This means that in practise the modelling error can not be quantified because only the adjoint variable is available.

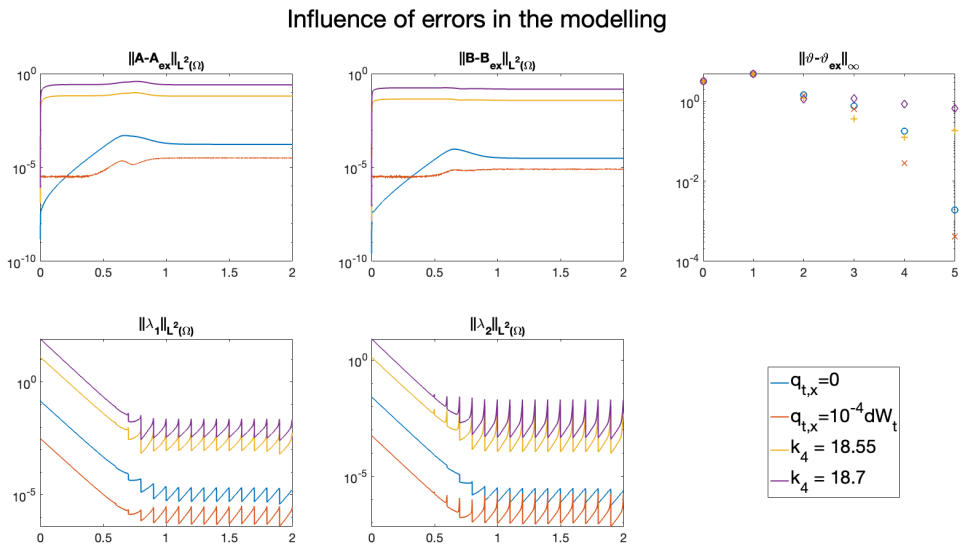

Figure 10: Comparison of results between Simulation 1, 13, 14 and 15.

4.3.7 Influence of errors in the initial condition

The error in estimating the initial condition is the only hyperparameter that is changing between Simulations 1, 16 and 17. We observe the impact of that hyperparameter in Figure 11. This hyperparameter does not impact the number of calibration required to get $\boldsymbol{\eta}$, nor the efficiency of the calibration. Indeed, the characterization of the error using a Gaussian form in the calibration, induces a spike at the initial time t_0 of the errors $||A - A_{ex}||_{L^2(\Omega)}$ and $||B - B_{ex}||_{L^2(\Omega)}$. Like the modelling error in Section 4.3.6, a poor estimation of the initial state cannot be seen in the adjoint variable $\boldsymbol{\lambda}$.

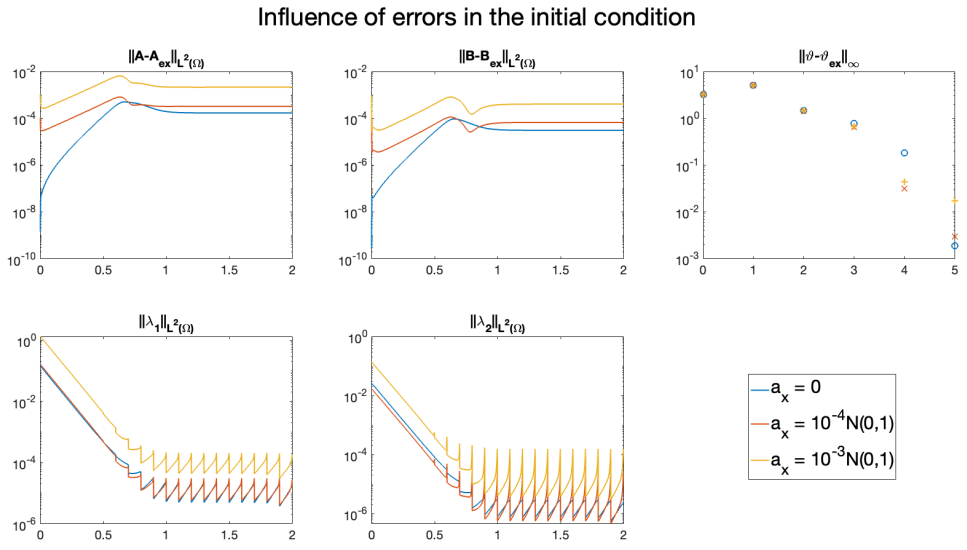

Figure 11: Comparison of results between Simulation 1, 16 and 17.

4.3.8 Influence of errors in the measurement process

Simulations 1, 18 and 19 only differ according to the error in the measurement process. Figure 12 depicts the impact of that hyperparameter in the simulation outputs. This hyperparameter affects the calibration's results. Indeed, The higher the error $\boldsymbol{\epsilon}$ is, the further away the calibration is from the expected values, as one can see the errors in the measurements' process in Figure 12. Those errors are apparent in the evolution of the adjoint variable $\boldsymbol{\lambda}$: the curves of $t \mapsto ||\lambda_1(t,\cdot)||_{L^2(\Omega)}$, $t \mapsto ||\lambda_2(t,\cdot)||_{L^2(\Omega)}$ are less smooth. This implies that this type of errors can be spotted in practise.

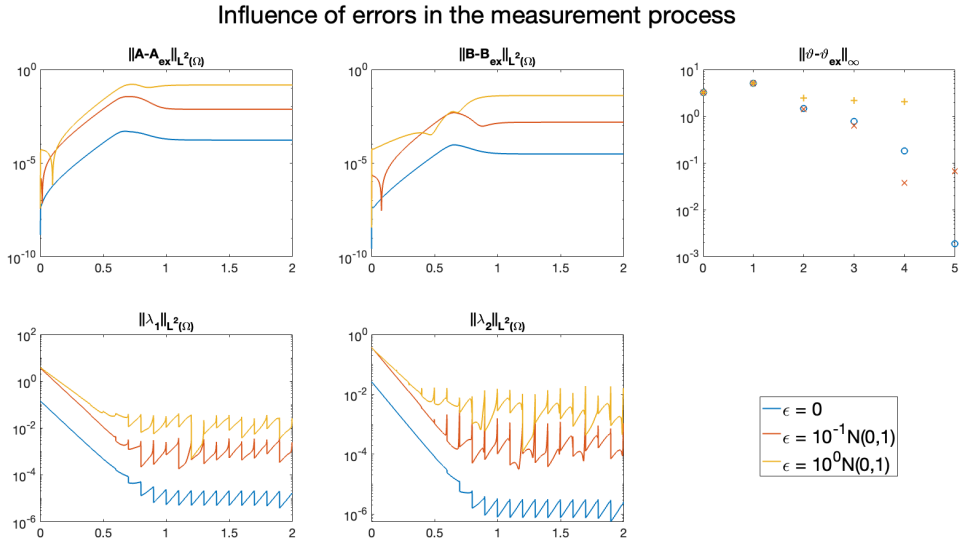

Figure 12: Comparison of results between Simulation 1, 18 and 19.

4.3.9 Influence of the range between l_b and u_b

The error in the measurement process changes between the simulations 1, 18 and 19. We can observe the impact of that hyperparameter in Figure 13. It affects the efficiency of the calibration: the wider the constraint space is (distance between l_b and u_b is higher), the more iterations are requested for the calibration. In practise this means that knowledge over the order of magnitude of the parameter of interest $\boldsymbol{\eta}$ leads to a faster calibration. However, a good calibration can be performed without some prior-knowledge on the parameter of interest $\boldsymbol{\eta}$.

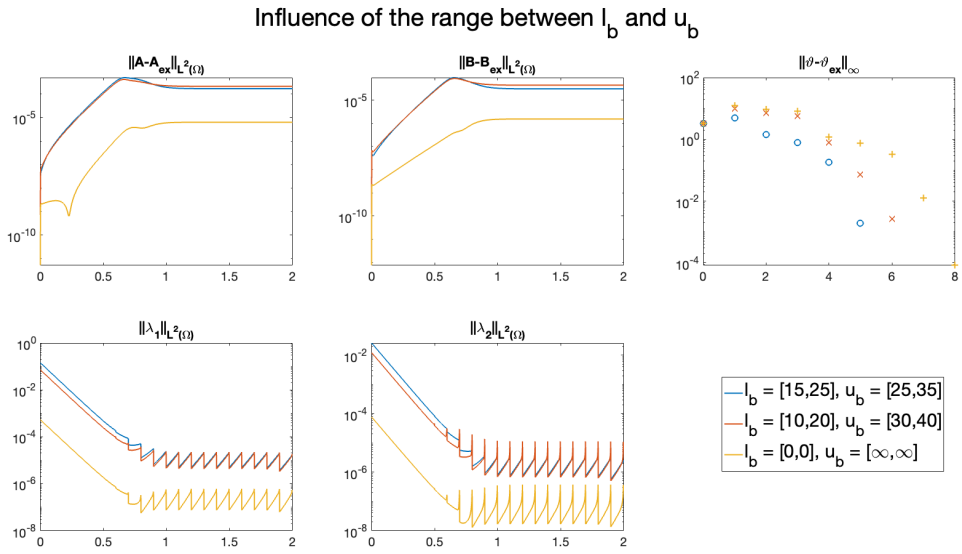

Figure 13: Comparison of results between Simulation 1, 20 and 21.

5 Conclusion

In this work, we have developed a method to solve inverse problems brought on by a reaction-diffusion model. This method relies on solving two PDE systems (corresponding to the direct model and its adjoint system) and the cancel out of a vectorial function with the same dimension as the number of coefficients that is needed to be calibrated. While the complete inverse problem looks complicated to solve, it has the benefits of being a deterministic system, i.e., it does not require statistical observations. Moreover, we have shown that those inverse problems can be numerically solved and that the combined state and parameter estimation was successful with synthetic data. The inverse problem formulation comes from the rewriting of an optimization problem by manipulating the variational form of the forward model. The optimization problem takes into account the errors in the modelling choices, the estimation of the initial state, and the observation measurement that stands as the difference between the data and the output of the predictive forward model. Each system of PDE has its associated inverse problem. Hyperparameters of the inverse problems are fitted according to the practical frame of each of the systems.

One future consideration would be the analysis of this methodology with real data to assert its robustness on real-world applications.

References

F. Alonzo, M. Saad, and A. A. Serandour. Simulating the behaviour of glioblastoma multiforme based on patient mri during treatments. *JOMB*, 2021. doi: https://doi.org/10.1007/s00285-022-01747-x.

G. Arumugam and J. Tyagi. Keller-segel chemotaxis models: A review. *Acta Applicandae Mathematicae*, 171(6), 2020. doi: https://doi.org/10.1007/s10440-020-00374-2.

M. Bendahmane, K. H. Karlsen, and J. M. Urbano. On a two-sidedly degenerate chemotaxis model with volume-filling effect. *Mathematical Models and Methods in Applied Sciences*, 17, 2007. doi: https://doi.org/10.1142/S0218202507002108.

A. Blanchet, J. Dolbeault, and B. Perthame. Two-dimensional keller-segel model: Optimal critical mass and qualitative properties of the solutions. *Electronic Journal of Differential Equations*, 44, 2006. URL `https://hal.archives-ouvertes.fr/hal-00021782`.

K. A. Brileya, J. M. Connolly, C. Downey, R. Gerlach, and M. W. Fields. Taxis toward hydrogen gas by methanococcus maripaludis. *Scientific Reports*, 3, 2013. doi: https://doi.org/10.1038/srep03140.

C. K. Chui and G. Chen. *Kalman Filtering*. Springer Cham, 5 edition, 2017. doi: https://doi.org/10.1007/978-3-319-47612-4.

A. Collin, T. Kritter, C. Poignard, and O. Saut. Joint state-parameter estimation for tumor growth model. *SIAM Journal on Applied Mathematics*, 81, 2021. doi: https://doi.org/10.1137/20M131775X.

D. Dochain. State and parameter estimation in chemical and biochemical processes: a tutorial. *Journal of Process Control*, 13, 2003. doi: https://doi.org/10.1016/S0959-1524(03)00026-X.

M. Eknes and G. Evensen. Parameter estimation solving a weak constraint variational formulation for an ekman model. *Journal of Geophysical Research*, 102, 1997. doi: https://doi.org/10.1029/96JC03454.

G. Evensen. The ensemble kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53, 2003. doi: https://doi.org/10.1007/s10236-003-0036-9.

G. Evensen. *Data Assimilation*, chapter 8. Springer-Verlag Berlin Heidelberg, second edition, 2009. doi: https://doi.org/10.1007/978-3-642-03711-5_8.

D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc.75 Arlington Street, Suite 300 Boston, MA, United States, 1989.

E. Grenier, V. Louvet, and P. Vigneaux. Parameter estimation in non-linear mixed effects models with saem algorithm: extension from ode to pde. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(5):1303–1329, 2014. doi: 10.1051/m2an/2013140.

E. Grenier, C. Helbert, V. Louvet, A. Samson, and P. Vigneaux. Population parametrization of costly black box models using iterations between saem algorithm and kriging. *Computational and Applied Mathematics*, 37(1):161–173, 2018. doi: 10.1007/s40314-016-0337-5.

C. Hogea, C. Davatzikos, and G. Biros. An image-driven parameter estimation problem for a reaction-diffusion glioma growth model with mass effects. *Journal of Mathematical Biology*, 56, 2008. doi: https://doi.org/10.1007/s00285-007-0139-x.

M. A. Iglesias, K. J. H. Law, and A. M. Stuart. Ensemble kalman methods for inverse problems. *Inverse Problems*, 29(4):045001, mar 2013. doi: 10.1088/0266-5611/29/4/045001.

D. S. Jones, M. J. Plank, and B. D. Sleeman. *Differential Equations and Mathematical Biology*. Chapman & Hall/CRC, 2 edition, 2009.

E. Konukoglu, O. Clatz, B. H. Menze, B. Stieltjes, M.-A. Weber, E. Mandonnet, H. Delingette, and N. Ayache. Image guided personalization of reaction-diffusion type tumor growth models using modified anisotropic eikonal equations. *IEEE TRANSACTIONS ON MEDICAL IMAGING*, 29, 2010. doi: https://doi.org/10.1109/TMI.2009.2026413.

F. Menolascina, R. Rusconi, V. I. Fernandez, S. Smriga, Z. Aminzare, E. D. Sontag, and R. Stocker. Logarithmic sensing in bacillus subtilis aerotaxis. *npj Systems Biology and Applications*, 3, 2017. doi: https://doi.org/10.1038/npjsba.2016.36.

J. Murray. *Mathematical Biology. II: Spatial Models and biomedical Applications*, chapter 3. Springer, third edition, 2003.

J. D. Murray. *Mathematical Biology I. An Introduction*. Springer New York, NY, 3 edition, 2002. doi: https://doi.org/10.1007/b98868.

NIH. National human genome research institute, 2022. URL `https://www.genome.gov/genetics-glossary/Personalized-Medicine`.

K. Osaki, T. Tsujikawa, A. Yagi, and M. Mimura. Exponential attractor for a chemotaxis-growth system of equations. *Nonlinear Analysis*, 51, 2002.

M. Rafiee, A. Tinka, J. Thai, and A. M. Bayen. Combined state-parameter estimation for shallow water equations. *Proceedings of the 2011 American Control Conference*, 2011. doi: https://doi.org/10.1109/ACC.2011.5991496.

M. Rochoux, A. Collin, C. Zhang, A. Trouvé, D. Lucor, and P. Moireau. Front shape similarity measure for shape-oriented sensitivity analysis and data assimilation for eikonal equation. *ESAIM: PROCEEDINGS AND SURVEYS*, 63, 2018. doi: https://doi.org/10.1051/proc/201863258.

M. M. Salek, F. Carrara, V. Fernandez, J. S. Guasto, and R. Stocker. Bacterial chemotaxis in a microfluidic t-maze reveals strong phenotypic heterogeneity in chemotactic sensitivity. *Nature Communications*, 10, 2019. doi: https://doi.org/10.1038/s41467-019-09521-2.

E. Simon, A. Samuelsen, L. Bertino, and S. Mouysset. Experiences in multiyear combined state–parameter estimation with an ecosystem model of the north atlantic and arctic oceans using the ensemble kalman filter. *Journal of Marine Systems*, 152, 2015. doi: https://doi.org/10.1016/j.jmarsys.2015.07.004.

Z. Song, H. Wang, J. Hou, H. F. Hofmann, and J. Sun. Combined state and parameter estimation of lithium-ion battery with active current injection. *IEEE TRANSACTIONS ON POWER ELECTRONICS*, 35, 2020. doi: https://doi.org/10.1109/TPEL.2019.2945513.

K. R. Swanson, E. C. A. Jr, and J. D. Murray. A quantitative model for differential motility of gliomas ingrey and white matter. *Cell Prolif.*, 33, 2000. doi: https://doi.org/10.1046/j.1365-2184.2000.00177.x.

A. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London B.*, 237, 1952. doi: https://doi.org/10.1098/rstb.1952.0012.

E. A. Wan and R. van der Menwe. The unscented kalman filter for nonlinear estimation. *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000. doi: https://doi.org/10.1109/ASSPCC.2000.882463.

T. A. Wenzel, K. J. Burnham, M. V. Blundell, and R. A. Williams. Dual extended kalman filter for vehicle state and parameter estimation. *Vehicle System Dynamics*, 44, 2006. doi: https://doi.org/doi/10.1080/00423110500385949.

A Algorithm 3

A.1 Keller-Segel System

Algorithm A.1: Numerical resolution of System (KS 2)

Data: N, Δt, $P = \frac{t_f - t_0}{\Delta t} + 1$
input : A guess $(\rho, \delta, \alpha, \beta, \gamma)$ of $\boldsymbol{\theta}$ and an approximation of $(u, c) \in \mathscr{M}_{N,P}(\mathbb{R})$ solving (KS 1)
output: An approximation of $(\lambda_1, \lambda_2) \in \mathscr{M}_{N,P}$ solving (KS 2)

$$G = \frac{D_1 \Delta t}{h^2} \begin{bmatrix} 1 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 1 \end{bmatrix} + I_N; \qquad H = \frac{D_2 \Delta t}{h^2} \begin{bmatrix} 1 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 1 \end{bmatrix} + I_N;$$

$\boldsymbol{\lambda}^P \leftarrow 0$;
$m \leftarrow P - 1$;
while $m \geq 1$ **do**
  /* Runge-Kutta 4 numerical scheme */
  Define $h^m : (\lambda_1, \lambda_2) \in (\mathbb{R}^N)^2 \mapsto \begin{bmatrix} \mathrm{diag}(\partial_u g_1(u^m, c^m)) & \mathrm{diag}(\partial_c g_1(u^m, c^m)) \\ \mathrm{diag}(\partial_u g_2(u^m, c^m)) & \mathrm{diag}(\partial_c g_2(u^m, c^m)) \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$;
  $k_1 = h^m(u^m, c^m)$; $\quad k_2 = h^m((u^m, c^m) + \frac{\Delta t}{2} k_1)$;
  $k_3 = h^m((u^m, c^m) + \frac{\Delta t}{2} k_2)$; $\quad k_4 = h^m((u^m, c^m) + \Delta t k_3)$;
  $\begin{bmatrix} \lambda_1^{m+\frac{1}{2}} \\ \lambda_2^{m+\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} \lambda_1^{m+1} \\ \lambda_2^{m+1} \end{bmatrix} - \frac{\Delta t}{6}(k_1 + k_2 + k_3 + k_4)$;
  /* λ_1 resolution */
  $$M_v = \begin{bmatrix} -1 & 1 & & & 0 \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ 0 & & & -1 & 1 \end{bmatrix}; \qquad V^m = \frac{1}{2h} M_v c^m; \qquad M_{conv} = \frac{D_1 \chi \Delta t}{2h} \mathrm{diag}(V^m) M_v;$$
  λ_1^m solves $(-G + M_{conv})\lambda_1^m = -\lambda_1^{m+\frac{1}{2}} - \Delta t \mathscr{C}(u^m, c^m)_1$;
  /* λ_2 resolution using λ_1^m */
  /* Upwind scheme for the convective term in λ_1 */
  $$N_{conv}^m = \frac{D_1 \chi \Delta t}{h^2} \begin{bmatrix} -(\lambda_1(1) - \lambda_1(2))^- & (\lambda_1(1) - \lambda_1(2))^+ & 0 \\ \ddots & \ddots & \ddots \\ (\lambda_1(i) - \lambda_1(i-1))^+ & -(\lambda_1(i) - \lambda_1(i-1))^- - (\lambda_1(i) - \lambda_1(i+1))^- & (\lambda_1(i) - \lambda_1(i+1))^+ \\ \ddots & \ddots & \ddots \\ 0 & (\lambda_1(N) - \lambda_1(N-1))^+ & -(\lambda_1(N) - \lambda_1(N-1))^- \end{bmatrix};$$
  if *there is a measurement available* **then**
    λ_2^m solves $-H\lambda_2^m = -\lambda_2^{m+\frac{1}{2}} - N_{conv}^m \cdot u^m - \Delta t \mathscr{C}(u^m, c^m)_2$;
  else
    λ_2^m solves $-H\lambda_2^m = -\lambda_2^{m+\frac{1}{2}} - N_{conv}^m \cdot u^m$;
  end
  $m \leftarrow m - 1$;
end
$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \leftarrow \begin{bmatrix} \lambda_1^1 & \lambda_1^2 & \dots & \lambda_1^P \\ \lambda_2^1 & \lambda_2^2 & \dots & \lambda_2^P \end{bmatrix}$;

A.2 Pattern Formation System

Algorithm A.2: Numerical resolution of System (PF 2)

Data: N, Δt, $P = \frac{t_f - t_0}{\Delta t} + 1$
input : A guess (d, μ) of $\boldsymbol{\vartheta}$ and an approximation of $(A, B) \in \mathscr{M}_{N,P}(\mathbb{R})$ solving (PF 1)
output: An approximation of $(\lambda_1, \lambda_2) \in \mathscr{M}_{N,P}$ solving (PF 2)

$$G = \frac{\Delta t}{h^2}\begin{bmatrix} 1 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 1 \end{bmatrix} + I_N; \qquad H = d\frac{\Delta t}{h^2}\begin{bmatrix} 1 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 1 \end{bmatrix} + I_N;$$

$\boldsymbol{\lambda}^P \leftarrow 0$;
$m \leftarrow P - 1$;
while $m \geq 1$ **do**
  $Mat_{11} = \mu\Delta t\,\mathrm{diag}(-\partial_A f_1(A^m, B^m))$;
  $Mat_{12} = \mu\Delta t\,\mathrm{diag}(-\partial_B f_1(A^m, B^m))$;
  $Mat_{21} = \mu\Delta t\,\mathrm{diag}(-\partial_A f_2(A^m, B^m))$;
  $Mat_{22} = \mu\Delta t\,\mathrm{diag}(-\partial_B f_2(A^m, B^m))$;
  if *there is a measurement available* **then**
    $\boldsymbol{\lambda}^m$ solves $\begin{bmatrix} -G - Mat_{11} & -Mat_{12} \\ -Mat_{21} & -H - Mat_{22} \end{bmatrix} \boldsymbol{\lambda}^m = \boldsymbol{\lambda}^{m+1} - \Delta t \begin{bmatrix} \mathscr{C}(A^m, B^m)_1 \\ \mathscr{C}(A^m, B^m)_2 \end{bmatrix}$
  else
    $\boldsymbol{\lambda}^m$ solves $\begin{bmatrix} -G - Mat_{11} & -Mat_{12} \\ -Mat_{21} & -H - Mat_{22} \end{bmatrix} \boldsymbol{\lambda}^m = \boldsymbol{\lambda}^{m+1}$
  end
  $m \leftarrow m - 1$;
end
$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \leftarrow \begin{bmatrix} \lambda_1^1 & \lambda_1^2 & \dots & \lambda_1^P \\ \lambda_2^1 & \lambda_2^2 & \dots & \lambda_2^P \end{bmatrix}$;

B Algorithm 4

B.1 Keller-Segel System

Algorithm B.1: Numerical resolution of System (KS 1)

Data: N, Δt, $P = \frac{t_f - t_0}{\Delta t} + 1$

input : A guess $(\rho, \delta, \alpha, \beta, \gamma)$ of $\boldsymbol{\theta}$ and an approximation of $(\lambda_1, \lambda_2) \in \mathscr{M}_{N,P}(\mathbb{R})$ solving (KS 2)

output: An approximation of $(u, c) \in \mathscr{M}_{N,P}$ solving (KS 1)

$$G = \frac{D_1 \Delta t}{h^2} \begin{bmatrix} 1 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 1 \end{bmatrix} + I_N; \qquad H = \frac{D_2 \Delta t}{h^2} \begin{bmatrix} 1 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 1 \end{bmatrix} + I_N;$$

$u^1 \leftarrow u_0 + (K_{aa}\boldsymbol{\lambda}^1)_1$; $\quad c^1 \leftarrow c_0 + (K_{aa}\boldsymbol{\lambda}^1)_2$; `/* Initial conditions */`

$m \leftarrow 2$;

while $m + 1 \leq P$ **do**

`/* Runge-Kutta 4 numerical scheme */`

$$k_1 = \begin{bmatrix} g_1^m(u^m, c^m) \\ g_2^m(u^m, c^m) \end{bmatrix}; \qquad k_2 = \begin{bmatrix} g_1^m((u^m, c^m) + \frac{\Delta t}{2} k_1) \\ g_2^m((u^m, c^m) + \frac{\Delta t}{2} k_1) \end{bmatrix};$$

$$k_3 = \begin{bmatrix} g_1^m((u^m, c^m) + \frac{\Delta t}{2} k_2) \\ g_2^m((u^m, c^m) + \frac{\Delta t}{2} k_2) \end{bmatrix}; \qquad k_4 = \begin{bmatrix} g_1^m((u^m, c^m) + \Delta t k_3) \\ g_2^m((u^m, c^m) + \Delta t k_3) \end{bmatrix};$$

$$\begin{bmatrix} u^{m+\frac{1}{2}} \\ c^{m+\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} u^m \\ c^m \end{bmatrix} + \frac{\Delta t}{6}(k_1 + k_2 + k_3 + k_4);$$

`/* Finite Difference to get` c^{m+1} `*/`

c^{m+1} solves $Hc^{m+1} = (c^{m+\frac{1}{2}} + \Delta t(K_{qq}\boldsymbol{\lambda}^{m+1})_2)$;

`/* Getting` u^{m+1} `using` c^{m+1} `*/`

`/* Upwind scheme for the convective term in` c^{m+1} `*/`

$$N_{conv}^{m+1} = \frac{D_1 \chi \Delta t}{h^2} \begin{bmatrix} -(c(1) - c(2))^- & (c(1) - c(2))^+ & 0 \\ \ddots & \ddots & \ddots \\ (c(i) - c(i-1))^+ & -(c(i) - c(i-1))^- - (c(i) - c(i+1))^- & (c(i) - c(i+1))^+ \\ \ddots & \ddots & \ddots \\ 0 & (c(N) - c(N-1))^+ & -(c(N) - c(N-1))^- \end{bmatrix};$$

`/* Finite Difference scheme to get` u^{m+1} `*/`

u^{m+1} solves $(G - N_{conv}^{m+1})u^{m+1} = (u^{m+\frac{1}{2}} + \Delta t(K_{qq}\boldsymbol{\lambda}^{m+1})_1)$;

$m \leftarrow m + 1$;

end

$$\begin{bmatrix} u \\ c \end{bmatrix} \leftarrow \begin{bmatrix} u^1 & u^2 & \dots & u^P \\ c^1 & c^2 & \dots & c^P \end{bmatrix};$$

B.2 Pattern Formation System

Algorithm B.2: Numerical resolution of System (PF 1)

Data: N, Δt, $P = \frac{t_f - t_0}{\Delta t} + 1$
input : A guess (d, μ) of $\boldsymbol{\vartheta}$ and an approximation of $(\lambda_1, \lambda_2) \in \mathscr{M}_{N,P}(\mathbb{R})$ solving (PF 2)
output: An approximation of $(A, B) \in \mathscr{M}_{N,P}$ solving (PF 1)

$$G = \frac{\Delta t}{h^2}\begin{bmatrix} 1 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 1 \end{bmatrix} + I_N; \qquad H = d\frac{\Delta t}{h^2}\begin{bmatrix} 1 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 1 \end{bmatrix} + I_N;$$

$A^1 \leftarrow A_0 + (K_{aa}\boldsymbol{\lambda}^0)_1; \qquad B^1 \leftarrow B_0 + (K_{aa}\boldsymbol{\lambda}^0)_2;$
$m \leftarrow 1;$
while $m + 1 \leq P$ **do**
  `/* Runge-Kutta 4 numerical scheme */`
  $k_1 = \begin{bmatrix} f_1(A^m, B^m) \\ f_2(A^m, B^m) \end{bmatrix}; \qquad k_2 = \begin{bmatrix} f_1((A^m, B^m) + \frac{\Delta t}{2} k_1) \\ f_2((A^m, B^m) + \frac{\Delta t}{2} k_1) \end{bmatrix};$
  $k_3 = \begin{bmatrix} f_1((A^m, B^m) + \frac{\Delta t}{2} k_2) \\ f_2((A^m, B^m) + \frac{\Delta t}{2} k_2) \end{bmatrix}; \qquad k_4 = \begin{bmatrix} f_1((A^m, B^m) + \Delta t k_3) \\ f_2((A^m, B^m) + \Delta t k_3) \end{bmatrix};$
  $\begin{bmatrix} A^{m+\frac{1}{2}} \\ B^{m+\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} A^m \\ B^m \end{bmatrix} + \frac{\Delta t}{6}(k_1 + k_2 + k_3 + k_4);$
  `/* Finite Difference scheme */`
  A^{m+1} solves $GA^{m+1} = A^{m+\frac{1}{2}} + \Delta t (K_{qq}\boldsymbol{\lambda}^{m+1})_1;$
  B^{m+1} solves $HB^{m+1} = B^{m+\frac{1}{2}} + \Delta t (K_{qq}\boldsymbol{\lambda}^{m+1})_2;$
  $m \leftarrow m + 1;$
end
$\begin{bmatrix} A \\ B \end{bmatrix} \leftarrow \begin{bmatrix} A^1 & A^2 & \dots & A^P \\ B^1 & B^2 & \dots & B^P \end{bmatrix};$

C Summary of simulations performed in Section 3.3

Table 2: Description of the simulations performed to study the impact of hyperparameters.

Simulation's number	Description of the simulation
1	reference simulation with $\sigma_\epsilon = 10^{-1}$, $\sigma_a = 10^{-4}$, $\sigma_q = 10^{-4}$, $\Delta m = 100\Delta t$, $m = N$, $\boldsymbol{q_{t,x}} = 0$, $\boldsymbol{a_x} = 0$, $\boldsymbol{\epsilon} = 0$, $l_b = [15, 25]$, and $u_b = [25, 35]$.
2	As in simulation 1 but with $\sigma_\epsilon = 10^{-2}$.
3	As in simulation 1 but with $\sigma_\epsilon = 10^{-0}$.
4	As in simulation 1 but with $\sigma_a = 10^{-3}$.
5	As in simulation 1 but with $\sigma_a = 10^{-5}$.
6	As in simulation 1 but with $\sigma_q = 10^{-3}$.
7	As in simulation 1 but with $\sigma_q = 10^{-5}$.
8	As in simulation 1 but with $\Delta m = 200\Delta t$.
9	As in simulation 1 but with $\Delta m = 50\Delta t$.
10	As in simulation 1 but with half measure points, i.e., $m = \frac{N}{2}$, randomly chosen.
11	As in simulation 1 but $m = 10$ points evenly spaced.
12	As in simulation 1 but with $m = 1$ point at the center of the interval of study.
13	As in simulation 1 but with a Wiener process in the exact simulation with a standard deviation of 10^{-4}.
14	As in simulation 1 but the value of k_4 is set to 18.7 for the exact solution.
15	As in simulation 1 but the value of k_4 is set to 18.55 for the exact solution.
16	As in simulation 1 but there is a Gaussian noise added in the initial condition with a standard deviation of 10^{-4}.
17	As in simulation 1 but there is a Gaussian noise added in the initial condition with a standard deviation of 10^{-3}.
18	As in simulation 1 but there is a Gaussian noise added in the measurement process with a standard deviation of 10^{-1}.
19	As in simulation 1 but there is a Gaussian noise added in the measurement process with a standard deviation of 10^{0}.
20	As in simulation 1 but we set l_b to $[10, 20]$ and u_b to $[30, 40]$.
21	As in simulation 1 but we set l_b to $[0, 0]$ and u_b to $[+\infty, +\infty]$.

D Plots related to the influence of σ_a

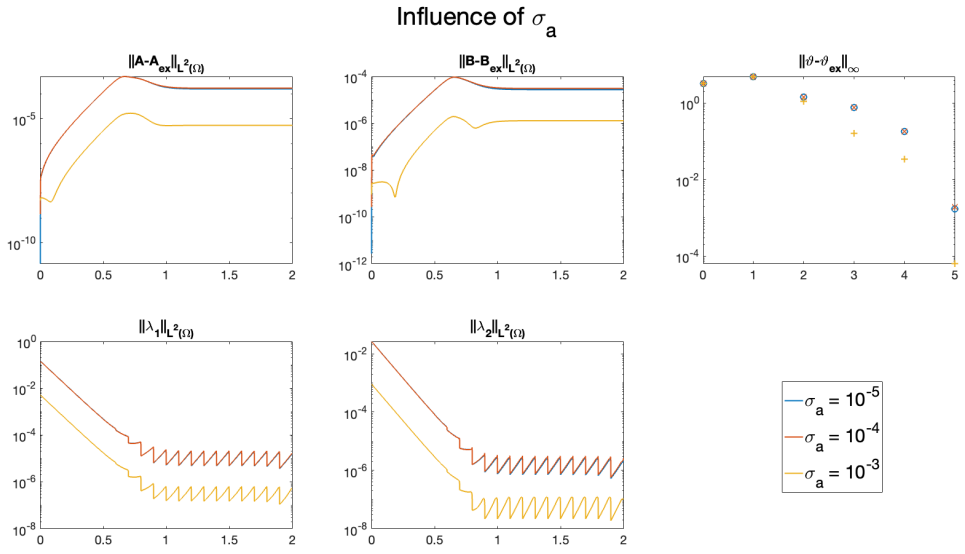

Figure 14: Comparison of results between Simulation 1, 4 and 5. The blue and red curves, while being distinct, are close in every plot.

E Plots related to the influence of σ_q

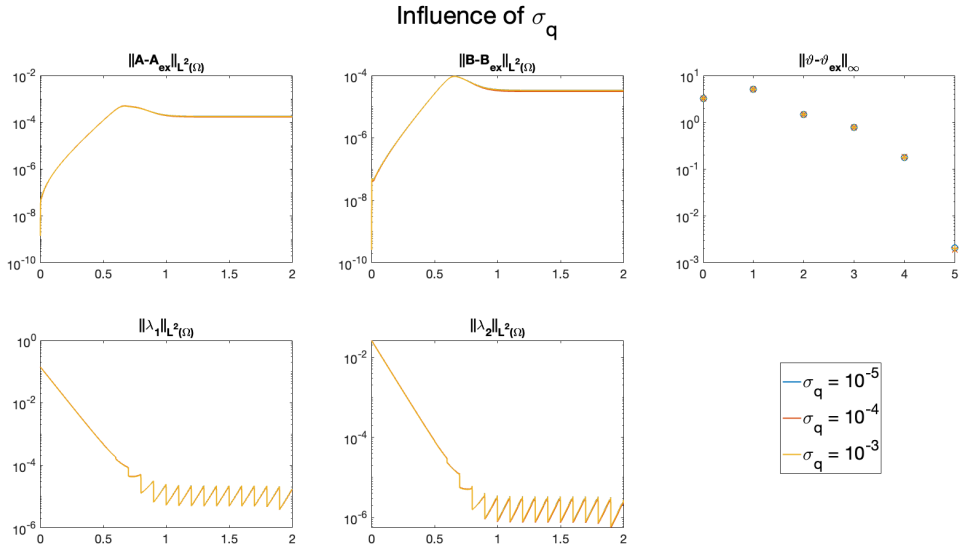

Figure 15: Comparison of results between Simulation 1, 6 and 7. All curves, while being distinct, are close in every plot.