



HAL
open science

Parameter Estimation using Integral Equations

François Lemaire, Louis Roussel

► **To cite this version:**

François Lemaire, Louis Roussel. Parameter Estimation using Integral Equations. 2024. hal-04560846

HAL Id: hal-04560846

<https://hal.science/hal-04560846v1>

Preprint submitted on 26 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Parameter Estimation using Integral Equations

François Lemaire¹ and Louis Roussel¹

¹Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000, Lille, France, {francois.lemaire,louis.roussel}@univ-lille.fr

Abstract

We investigate the interest of using nonlinear integral equations instead of nonlinear differential equations in a modelisation context. In particular, we perform parameter estimation on an academic example using different input-output differential and integral equations.

Keywords: integro-differential equations, parameter estimation, differential and integral elimination

1 Introduction

Many models (such as the SIR¹ epidemiological model) consist of parametric non-linear differential equations. When parameters values are unknown, they can sometimes be estimated thanks to experimental data through a process called parameter estimation.

For instance, in an epidemiological model, one might be interested in determining the rate of spread of the virus from experimental data.

Typical nonlinear ODE systems are composed of n differential equations of the form $\dot{x}_i(t) = f_i(x_1(t), \dots, x_n(t), k_1, \dots, k_p, u(t))$, where the $x_i(t)$'s define the state of system (which is unknown), the k_j 's are the parameters and $u(t)$ is a vector of inputs (which is known and can be freely chosen). Moreover, some measurements are available though the so-called output variables $y_1(t), \dots, y_k(t)$ which are all functions of the state, the parameters and the input. In order to estimate the parameters, a typical approach consists in computing from the initial model some differential equations only involving the input, the output, and the parameters, which are naturally called input/output (I/O) equations. In a polynomial settings, those I/O equations can be computed through a differential elimination procedure such as [6].

Recent work [1, 5, 3] investigates the treatment of integro-differential equations i.e. equations involving both derivation and integration operations. Considering models with integro-differential equations is motivated by the following

¹Susceptible, Infectious, Recovered

reasons: on some examples, the introduction of integral equations increases the expressiveness of the models, improves the estimation of parameter values from error-prone measurements, and reduces the size of the equations.

Parameter estimation using integral I/O equations is also possible. Like in the differential case, one can obtain an integral I/O equation through an elimination procedure (Figure 1). However, there is currently no algorithm for the integral elimination, hence calculus are done by hand (either a by a direct elimination, or by integrating a differential I/O equation possibly several times, see Figure 1). Figure 1 illustrates two I/O integral and differential equations for a simple system with no input, a single state $x(t)$ and a single output variable $y(t)$.

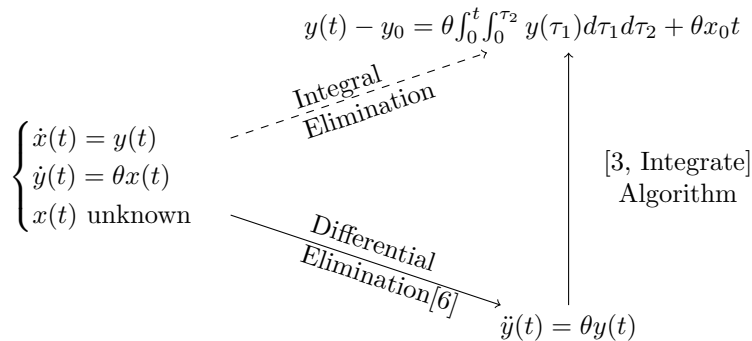


Figure 1: **Modelling, elimination and [3, Integrate] algorithm.** From a differential system, we can obtain an input/output equation [7] which only involves the known function $y(t)$, by using differential elimination techniques [4]. From this equation, we can use the [3, Integrate] algorithm to compute an integral equation. Using numerical methods (e.g. least squares), it is possible to evaluate the value of θ from any of the two input/output equations.

In this paper, we will consider an academic example to illustrate parameter estimation using both integral and differential equations. We will focus on :

- comparing the quality of parameter estimation using differential or integral equations,
- comparing different integral equations, some of which contains exponential terms.

Our long term goal is to develop a prototype for the integral elimination. This article is a first step for choosing the type of equations an integral elimination process should produce, and also in which algebraic structures they should be taken.

This last point concerning algebraic structures is certainly a difficult problem, and is considered in [10]. Introducing fractions, exponential terms, ... amongst integration is an extra challenge. For this reason, all the computations

in the article are made in an analytic context, where the expression $\int f$ denotes $\int_0^t f(\tau)d\tau$, i.e. the primitive of f cancelling at $t = 0$.

First, we will describe the chosen academic model and related work (Section 2). Then, in Section 3 we explain the idea of parameter estimation by treating this example with differential algebra. Section 4 details the different methods (integration and elimination by hand) used to obtain 4 different input/output integral equations.

Finally, in Section 5, we compare the quality of the differential and integral I/O equations in terms of parameter estimation.

2 Related Work

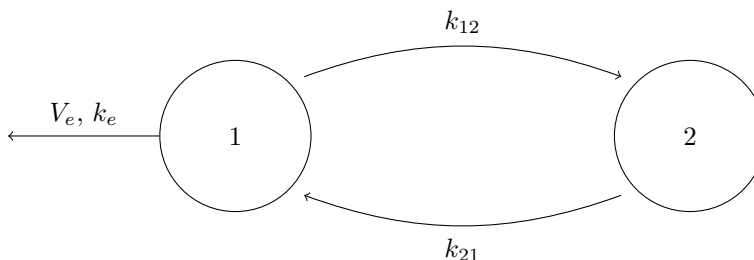


Figure 2: Two compartment model

The Figure 2 illustrates the considered model associated to the system (1) studied in [2]. Compartment 1 corresponds to the blood system and compartment 2 to some organ (e.g. liver). A medical drug is injected into the blood system and diffuses between the two compartments following linear laws with parameters named k_{12} and k_{21} . The drug leaves the blood system with an implicit enzymatic reaction which involves two parameters V_e and k_e .

In the system (1), the measurement (known values) $y(t)$ is the concentration of drugs in compartment 1. The variable $x(t)$ is the unknown drug concentration in the other compartment (the organ).

$$\begin{cases} \dot{x} = k_{12}y - k_{21}x \\ \dot{y} = -k_{12}y + k_{21}x - \frac{V_e y}{k_e + y} \end{cases} \quad (1)$$

The work done in [2] consists in computing both differential and integral I/O equations and comparing the quality of the parameter estimation. In [2], a simplification was made in the system: k_e was assumed equal to 1 to allow the use of least squares with the integral equation. We will see in Section 5 that k_e can be estimated, by replacing the linear least squares algorithm by a more general optimisation method.

3 Differential I/O Equations and Parameter Estimation

This section provides the computation details for obtaining the differential I/O equation (using differential algebra techniques) and how to use it to perform the parameter estimation. The following calculations can be fully automated in [6] with the so-called **RosenfeldGröbner** algorithm [4]. The details of the computations are too complex to be exposed here. However, the computations can be summarised in a loose way using rules based computations. We chose a ranking that sorts the derivatives of x and y , and which eliminates the unknown derivatives of x :

$$\dots > \ddot{x} > \dot{x} > x > \dots > \ddot{y} > \dot{y} > y$$

Based on this ranking, we write the equations of the system (1) as two rules:

$$\begin{aligned} \dot{x} &\xrightarrow{(R_1)} k_{12}y - k_{21}x \\ k_{21}x &\xrightarrow{(R_2)} \dot{y} + k_{12}y + \frac{V_e y}{k_e + y} \end{aligned}$$

Using Rule (R_2) , we can replace the value of $k_{21}x$ in Rule (R_1) which becomes:

$$\dot{x} \xrightarrow{(R'_1)} -\dot{y} - \frac{V_e y}{k_e + y}$$

By differentiating Rule (R_2) and multiplying by k_{21} the new rule (R'_1) , we obtain Rules (R_3) and (R_4) :

$$\begin{aligned} k_{21}\dot{x} &\xrightarrow{(R_3)} -k_{21}\dot{y} - k_{21}\frac{V_e y}{k_e + y} \\ k_{21}\dot{x} &\xrightarrow{(R_4)} \ddot{y} + k_{12}\dot{y} + \frac{V_e \dot{y}}{k_e + y} - \frac{V_e y \dot{y}}{(k_e + y)^2} \end{aligned}$$

and consequently, equating the right hand sides, the equation

$$k_{21}\dot{y} + k_{21}\frac{V_e y}{k_e + y} + \ddot{y} + k_{12}\dot{y} + \frac{V_e k_e \dot{y}}{(k_e + y)^2} = 0$$

corresponding to the expected differential I/O equation. Indeed, this equation does not involve x and its derivatives anymore (x has been eliminated). Up to the denominator $(k_e + y)^2$, this result is equal to the differential polynomial computed by **RosenfeldGröbner**:

$$\begin{aligned} V_e k_{21} k_e y + V_e k_{21} y^2 + V_e k_e \dot{y} + k_{12} k_e^2 \dot{y} + 2k_{12} k_e y \dot{y} + k_{12} y^2 \dot{y} \\ + k_{21} k_e^2 \dot{y} + 2k_{21} k_e y \dot{y} + k_{21} y^2 \dot{y} + k_e^2 \ddot{y} + 2k_e y \ddot{y} + y^2 \ddot{y} = 0 \end{aligned} \quad (2)$$

Before using least squares to estimate the parameters, we first rewrite this equation by assuming $(k_e = 1)$ as in [2]:

$$\ddot{y} y^2 + 2\dot{y} y + \ddot{y} + \theta_2 \dot{y} y^2 + \theta_2 2\dot{y} y + \theta_3 \dot{y} + \theta_1 y^2 + \theta_1 y = 0 \quad (3)$$

where the θ_i stand for the so-called blocks of parameters:

$$\theta_1 = k_{21}V_e, \quad \theta_2 = k_{12} + k_{21}, \quad \theta_3 = k_{12} + k_{21} + V_e. \quad (4)$$

Since the values of $y(t)$ are known, the values of \dot{y} and \ddot{y} can be estimated numerically. As a consequence, Equation (3) can be specialised at different times values, yielding an (overdetermined) linear system of linear equations in the θ_i . Using linear least squares, values for θ_1 , θ_2 and θ_3 can be estimated. Finally, Equations (4) can be utilised to compute an estimation of the parameters values. If we keep the parameter k_e in (2), the procedure for parameter estimation is similar and yield different blocks of parameters.

4 Computing I/O Integral Equations

4.1 Integration of the Differential Equation

The most straightforward method for computing the I/O integral equation consists in integrating the differential equation (2). To this aim, [3, Integrate] algorithm can be used. Unfortunately, [3, Integrate] applied on Equation (2) yield equations which still involve derivatives of y . This is not suitable since our purpose is to get rid of all derivatives. However, it is possible to first divide Equation (2) by $(y + k_e)^2$, thus obtaining

$$k_{21}V_e \frac{y}{y + k_e} + (k_{12} + k_{21}) \frac{y\dot{y}(y + 2)}{(y + k_e)^2} + (k_{12} + k_{21} + V_e) \frac{\dot{y}}{(y + k_e)^2} = -\ddot{y} \quad (5)$$

and then apply [3, IteratedIntegrate] (which is an iterated version of Integrate) on (5) which computes the following integral equation:

$$\begin{aligned} & k_{21}V_e \int \int \frac{y}{y + k_e} \\ & + (k_{12} + k_{21}) \int \left(\frac{y^2}{y + k_e} - \frac{y_0^2}{y_0 + k_e} \right) \\ & - (k_{12} + k_{21} + V_e) \int \left(\frac{1}{y + k_e} - \frac{1}{y_0 + k_e} \right) - \dot{y}_0 t = -y + y_0. \end{aligned} \quad (6)$$

Notice that the parameter k_e now appears in the denominator inside integrals. For this reason, it is not possible to write Equation (6) as a linear combination of blocks of parameters, as it was done in Equation (3): this prevents the use of linear least squares.

Equation (6) was obtained thanks to the multiplication by $\frac{1}{(y+k_e)^2}$ that we could loosely call an integrating factor. Computing such a factor is not straightforward, especially in our special context where expressions involve integrals. For this reason, we have tried a deep learning approach that we briefly detail in the next section.

4.2 Deep Learning for Computing Integration Factors

Deep Learning techniques consist in collecting/generating a large amount of data relevant to a particular task to train a model. In our case, given an integro-differential equation p , the model should compute an integro-differential expression f such that fp can be integrated into an equation q , such that the order of derivation of q is lower than that of p .

To narrow the difficulty, we started from the following property: if A , B and C are integro-differential expressions, then the derivative of $A/B + \int(C/B)$ is $\frac{A'B - AB' + CB}{B^2}$. As a consequence, integrating the expression $A'B - AB' + CB$ can be done by first dividing by B^2 and then integrating to retrieve $A/B + \int(C/B)$. Thus, we built a dataset for training the model (Transformer [11]) by generating equations of the form $A'B - A'B + CB$ with the objective of producing B as output. This process described above is a slight adaptation of [9].

Applied to our example, the input of the model is the differential equation (2) with all the parameters substituted by 1 (needed to reduce the number of variables, hence to reduce the difficulty of the problem):

$$y + y^2 + \dot{y} + \dot{y} + 2y\dot{y} + y^2\dot{y} \\ + \dot{y} + 2y\dot{y} + y^2\dot{y} + \ddot{y} + 2y\ddot{y} + y^2\ddot{y} = 0.$$

Our model is able to compute $B = y + 1$ which yields to the expected integrating factor ($\frac{1}{B^2} = \frac{1}{(y+1)^2}$) with $k_e = 1$. To retrieve the correct value of $B = y + k_e$, we could increase the size of model (in particular the number of variables/parameters it can treat), or use the value $B = y + 1$ as a first guess.

4.3 Integral Elimination

As seen before, integrating the differential I/O equation can be difficult, especially because an integrating factor is sometimes needed.

We now present on our example another approach (which is not algorithmic yet) to perform the elimination, which directly produces the expected integral I/O equation. This approach is similar to the one described in Section 3 (it relies on manipulating rewriting rules), but only integrations are allowed. Computations below follow those of [5, page 17] using rule based computations.

First, let's consider the integral version of the differential system (1)

$$\begin{cases} x = x_0 + k_{12} \int y - k_{21} \int x, \\ y = y_0 + k_{21} \int x - k_{12} \int y - \int \frac{V_e y}{k_e + y}. \end{cases} \quad (7)$$

We fix a kind of ranking where terms involving x are greater than terms involving y only. Note that fixing a ranking of the form

$$\dots > \int \int x > \int x > x > \dots > \int \int y > \int y > y$$

does not work since it does not permit to compare the expressions $k_{12} \int x$ and $\int \frac{V_e y}{k_e + y}$ as needed below.

Let us write the equations of System (7) as rules:

$$\begin{aligned} k_{21} \int x \xrightarrow{(R_1)} x_0 - x + k_{12} \int y, \\ k_{21} \int x \xrightarrow{(R_2)} y - y_0 + k_{12} \int y + \int \frac{V_e y}{k_e + y}. \end{aligned}$$

Equating right hand sides of Rules (R_1) and (R_2) yields $x_0 - x = y - y_0 + \int \frac{V_e y}{k_e + y}$ which can be written as a third rule:

$$x \xrightarrow{(R_3)} x_0 - y + y_0 - \int \frac{V_e y}{k_e + y}.$$

Rules (R_2) and (R_3) can be unified by integrating and multiplying Rule (R_3) by k_{21} . This results in a new rule:

$$k_{21} \int x \xrightarrow{(R_4)} k_{21} x_0 t - k_{21} \int y + k_{21} y_0 t - k_{21} \int \int \frac{V_e y}{k_e + y}.$$

Then, equating right hand sides of Rules (R_4) and (R_2) yield the following equation

$$-k_{21} V_e \int \int \frac{y}{k_e + y} - (k_{12} + k_{21}) \int y - V_e \int \frac{y}{k_e + y} + (x_0 + y_0) k_{21} t - y + y_0 = 0$$

which is almost free of x since it still involves the initial condition x_0 .

Evaluating System (3) at $t = 0$ yields $x_0 = \frac{1}{k_{21}}(\dot{y}_0 + k_{12}y_0 + \frac{V_e y_0}{k_e + y_0})$, which is used to rewrite x_0 in the previous equation, thus obtaining

$$\begin{aligned} k_{21} V_e \int \int \frac{y}{k_e + y} \\ + (k_{12} + k_{21}) \int (y - y_0) \\ - V_e \int \left(\frac{y}{k_e + y} - \frac{y_0}{k_e + y_0} \right) - \dot{y}_0 t = -y + y_0 \end{aligned} \tag{8}$$

which is now free of x . Note that this last equation (8) is equivalent to Equation (6) by using the properties $y^2/(1+y) = y - 1 + 1/(1+y)$ and $1/(y+1) = 1 - y/(y+1)$. We will only use Equation (8) for the experiments.

4.4 Integral Equations using Exponential

This section will present three additional integral equations involving exponentials. To this extent, we need the following two lemmas.

Lemma 1 (Product of integral equations). *If $A_1(t) = A_1(0) + \int C_1(t)$ and $A_2(t) = A_2(0) + \int C_2(t)$, then*

$$A_1(t)A_2(t) = A_1(0)A_2(0) + \int C_1(t)A_2(t) + \int A_1(t)C_2(t).$$

Proof. Just differentiate $A_1(t)A_2(t)$ and use the fact that $A'_1(t) = C_1(t)$, $A'_2(t) = C_2(t)$, and that evaluating an integral at $t = 0$ yields 0. \square

Lemma 2. *If $A(t) = A(0) + \int(A(t) \cdot G(t) + F(t))$, then by introducing the equation $v(t) = v(0) - \int v(t)G(t)$ (which encodes: $v(t) = v(0)e^{-\int G(t)}$) we have*

$$A(t)v(t) = A(0)v(0) + \int v(t)F(t).$$

Proof. Direct application of Lemma 1. \square

First equation: Applying Lemma 2 on the first equation in the system (7)

$$\underbrace{x}_A = x_0 + \int \underbrace{k_{12}y}_F + \underbrace{(-k_{21})}_G \underbrace{x}_A,$$

and by introducing

$$v = 1 + \int vk_{21},$$

one obtains

$$xv = x_0 + \int vk_{12}y. \quad (9)$$

Here v simply encodes $e^{k_{21}t}$. Introducing the inverse u of v (i.e. $u = e^{-k_{21}t}$), and multiplying Equation (9) by u , one obtains the following rule

$$x \xrightarrow{(R_1)} x_0u + u \int vk_{12}y. \quad (10)$$

The second equation of System (7) can be written as the rule

$$k_{21} \int x \xrightarrow{(R_2)} y - y_0 + k_{12} \int y + \int \frac{V_e y}{k_e + y}.$$

Writing u and v as exponentials, and after replacing x_0 as done in Equation (8), the critical pair with $\int k_{21}(R_1)$ and (R_2) yields the following equation

$$\begin{aligned} & -k_{12} \int y - \int \frac{V_e y}{k_e + y} + \left(\frac{V_e y_0}{k_e + y_0} + \dot{y}_0 + k_{12}y_0 \right) \int e^{-k_{21}t} \\ & \quad \quad \quad -y + y_0 \\ & \quad \quad \quad + k_{21}k_{12} \int (e^{-k_{21}t} \int e^{k_{21}t} y) = 0 \end{aligned} \quad (11)$$

where the parameter k_{21} is “stuck” inside an exponential term.

Second equation: The rule from Equation (10) yields $x = x_0 e^{-k_{21}t} + e^{-k_{21}t} \int e^{k_{21}t} k_{12}y$, that is used to replace x in the first equation of System (1) to obtain

$$k_{21} \left(x_0 e^{-k_{21}t} + e^{-k_{21}t} \int e^{k_{21}t} k_{12}y \right) = \dot{y} + k_{12}y + \frac{V_e y}{k_e + y}.$$

Multiplying this last equation by $y + k_e$, replacing x_0 as done in Equation (8), and integrating yields

$$\begin{aligned} & -y^2 + y_0^2 - 2k_e y + 2k_e y_0 \\ & - 2k_e k_{12} \int y - 2k_{12} \int y^2 - 2V_e \int y \\ & + 2k_e \left(\frac{V_e y_0}{k_e + y_0} + \dot{y}_0 + k_{12}y_0 \right) \int e^{-k_{21}t} \\ & + 2 \left(\frac{V_e y_0}{k_e + y_0} + \dot{y}_0 + k_{12}y_0 \right) \int y e^{-k_{21}t} \\ & + 2k_e k_{21} k_{12} \int (e^{-k_{21}t} \int e^{k_{21}t} y) \\ & + 2k_{21} k_{12} \int (y e^{-k_{21}t} \int e^{k_{21}t} y) = 0. \end{aligned} \tag{12}$$

Third equation : Previously, we applied Lemma 2 to the first equation of System (7) to obtain Equation (10). We can also apply Lemma 2 on the second equation of System (7) with $A = y$, $G = -(k_{12} + \frac{V_e}{k_e + y})$ and $F = k_{21}x$ to obtain

$$y e^{\int (k_{12} + \frac{V_e}{k_e + y})} = y_0 + \int k_{21} x e^{\int (k_{12} + \frac{V_e}{k_e + y})}.$$

Finally, substituting x (using Equation (10)) and x_0 into the last equation yields:

$$\begin{aligned} & -y e^{\int (k_{12} + \frac{V_e}{k_e + y})} \\ & + y_0 + \left(\frac{V_e y_0}{k_e + y_0} + \dot{y}_0 + k_{12}y_0 \right) \int e^{\int (k_{12} + \frac{V_e}{k_e + y})} e^{-k_{21}t} \\ & + k_{12} k_{21} \int (e^{\int (k_{12} + \frac{V_e}{k_e + y})} e^{-k_{21}t} \int e^{k_{21}t} y) = 0 \end{aligned} \tag{13}$$

This last equation seems quite complicated since there are 3 parameters stuck inside exponential or fractions.

Open question. The four integral I/O equations (8), (11), (12) and (13) all involve parameters either stuck in fractions inside integrals, or stuck in exponentials. We tried without success to obtain an integral I/O equation without these limitations i.e. we could not find any integral I/O equation that could be written as a linear combination of blocks of parameters. At this stage, we do not know whether such an I/O equation exists or not.

5 Experiments and Interpretation

In this section, we study the quality of the different I/O equations obtained previously (see Section A of the appendix for a summary). We first conduct experiments by performing many parameter estimations with different levels of noise, and discuss the quality of the different I/O equations.

5.1 Experiments

An exact signal $\bar{y}(t)$ has been generated by numerically solving System (1) with $x(0) = 10, y(0) = 1, \bar{k}_{12} = 1, \bar{k}_{21} = 5, \bar{V}_e = 3, \bar{k}_e = 1$ for $0 \leq t \leq 4$, and then sampling with $N = 1000$ (equally spaced) time values denoted $(t_i)_{0 \leq t_i \leq N-1}$. When adding some noise to the exact signal $\bar{y}(t)$, estimating the parameters produces approximate values of the parameters k_{12}, k_{21}, V_e and k_e . Those approximate parameters defines a relative error equal to the 2-norm of the difference between the vector of predicted values and known values, divided by the vector of known values.

For our experiments, we have used additive white noises (defined by their standard deviation). To study the dependency of the I/O equations to the level of noise, we increase the standard deviation of the noise from zero (no noise at all) to 0.8, by steps of 0.01. To smooth the experimental curves, we compute for each level of noise the mean of the relative errors of 50 different parameter estimations (each one for a different noise).

Finally, to be closer to the experiments from [2] where the parameter k_e was fixed to 1, we have performed two experiments. In the first one (see Figure 3), we fix $k_e = 1$ and estimate parameters k_{12}, k_{21}, V_e (plus the parameter \dot{y}_0 for the integral equations). In the second one (see Figure 4), we estimate in addition the parameter k_e .

For all plots in Figures 3 and 4, the x-axis is the standard deviation of the noise level, and the y-axis is the relative error.

Details on parameter estimation For a given I/O equation p , the parameter estimation is performed by cancelling the I/O equation p . More precisely, we introduce the following error

$$E(k_{12}, k_{21}, k_e, V_e) = \frac{1}{N} \sum_{i=0}^{N-1} p(t_i)^2 \quad (14)$$

that we minimize. Note that each $p(t_i)$ is evaluated numerically using a 4 points scheme to estimate derivatives, and the Simpson rule for integrals.

We used the Covariance Matrix Adaptation Evolution Strategy [8, CMA-ES] algorithm with each parameter bounded in the interval $[10^{-5}, 100]$. This method behaves quite well on our examples. We also tried different variants of gradients descents which were not conclusive.

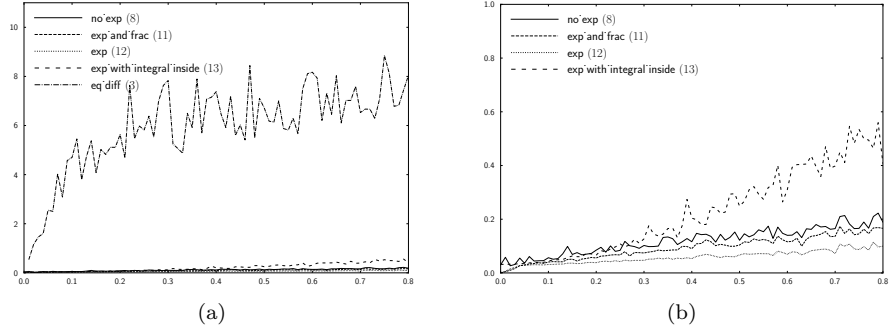


Figure 3: Estimation using CMA-ES of the parameters k_{12} , k_{21} , V_e and \hat{y}_0 for I/O equations. Equations `no_exp`, `exp_and_frac`, `exp`, `exp_with_integral_inside` and `eq_diff` correspond respectively to Equations (8), (11), (12), (13) and (3). Figures (a) and (b) correspond to the same experiment with and without the differential equation.

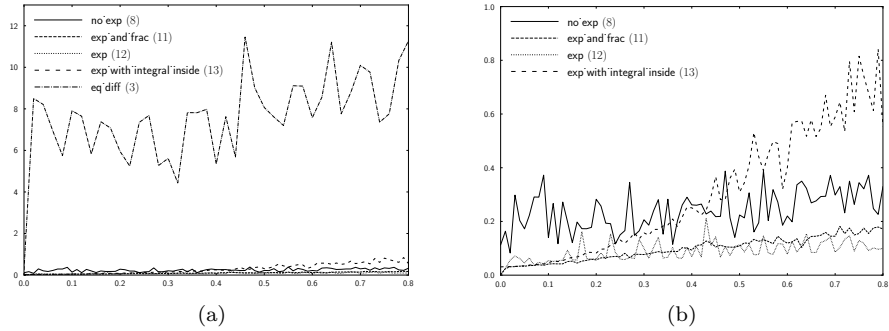


Figure 4: Estimation using CMA-ES of the parameters k_{12} , k_{21} , V_e , k_e and \hat{y}_0 for integral equations. Equations `no_exp`, `exp_and_frac`, `exp`, `exp_with_integral_inside` and `eq_diff` correspond respectively to Equations (8), (11), (12), (13) and (3). Figures (a) and (b) correspond to the same experiment with and without the differential equation.

Numerical consistency of parameter estimation using linear least squared or CMA-ES In [2], only the values of the blocks (4) were estimated. However, in this work, we directly compute the parameters. To check whether the two approaches are numerically consistent, we computed the values of the blocks for the parameters values computed by CMA-ES, and we could verify that the quality of the blocks are indeed consistent (i.e. similar relative errors) between the two approaches.

5.2 Interpretation

Both experiments present the same behaviour when comparing integral and differential I/O equations: integral equations perform much better than the differential I/O equation. Even with a significant amount of noise (0.8) the relative error remains below one for all integral equations.

In the first experiment, the three integral Equations (8), (11) and (12) perform quite well, and the last integral equation (13) is slightly worse. The differential I/O equation is not conclusive at all.

In the second experiment, the integral equations (11) and (12) perform quite well, the integral equation (13) is still slightly worse. The differential I/O equation is even less conclusive. Surprisingly, Equation (8) is now much more sensitive, even with a low level of noise. In fact, in Figure 4, a problem arises with Equations (8) and (13): for around 5% of parameter estimations, the parameter estimation fails because a parameter is equal to either 10^{-5} or 100 (which are the interval bounds). This increases the mean relative error for those equations, because estimated values are too far from the expected values.

As a conclusion, the best equations are integral equations that involves exponentials and fractions. This is a bit surprising, since Equation (8) seems simpler and does not involve exponentials.

6 Conclusion

We have compared the quality of different I/O equations in a context of parameter estimation, on a specific example. The integral equations behave quite well, even if they involve exponentials or fractions. Contrary to the approach of [2], we directly estimated the parameters instead of the block parameters, because our integral equations had no block structures. This difference of method did not harm the quality of the parameter estimation.

As mentioned in the introduction, our long term goal is to build an integral elimination prototype. The integral equations presented here give an idea of the kind of equations an integral elimination algorithm should produce.

Acknowledgements

This work has been partially supported by the THIA ANR Program “AI-PhD@Lille”, from the French National Research Agency, grant ANR-20-THIA-0014.

References

- [1] François Boulier, Hélène Castel, Nathalie Corson, Valentina Lanza, François Lemaire, Adrien Poteaux, Alban Quadrat, and Nathalie Verdière. Symbolic-Numeric Methods for Nonlinear Integro-Differential Modeling. working paper or preprint, April 2018.
- [2] François Boulier, Anja Korporal, François Lemaire, Wilfrid Perruquetti, Adrien Poteaux, and Rosane Ushirobira. An Algorithm for Converting Nonlinear Differential Equations to Integral Equations with an Application to Parameter Estimation from Noisy Data. In *Computer Algebra in Scientific Computing*, pages 28 – 43, Warsaw, Poland, September 2014.
- [3] François Boulier, Joseph Lallemand, François Lemaire, Georg Regensburger, and Markus Rosenkranz. Additive Normal Forms and Integration of Differential Fractions. *Journal of Symbolic Computation*, 2016.
- [4] François Boulier, Daniel Lazard, François Ollivier, and Michel Petitot. Computing representations for radicals of finitely generated differential ideals. *Applicable Algebra in Engineering, Communication and Computing*, 20(1):73, Mar 2009.
- [5] François Boulier, François Lemaire, Markus Rosenkranz, Rosane Ushirobira, and Nathalie Verdière. On Symbolic Approaches to Integro-Differential Equations. In Alban Quadrat and Eva Zerz, editors, *Algebraic and Symbolic Computation Methods in Dynamical Systems*, volume 9 of *Advances in Delays and Dynamics*, pages 161–182. Springer, 2020.
- [6] François Boulier. The DifferentialAlgebra project, 2023.
- [7] Michel Fliess. Automatique et corps différentiels. *Forum Mathematicum - FORUM MATH*, 1:227–238, 01 1989.
- [8] Nikolaus Hansen, Youhei Akimoto, and Petr Baudis. CMA-ES/pycma on Github. Zenodo, DOI:10.5281/zenodo.2559634, February 2019.
- [9] François Lemaire and Louis Roussel. Integration of Integro-differential Equations and Deep Learning. Differential Algebra and Related Topics XI, June 2023. Poster.
- [10] Markus Rosenkranz and Georg Regensburger. Integro-differential polynomials and operators. In *Proceedings of the Twenty-First International Symposium on Symbolic and Algebraic Computation, ISSAC '08*, page 261–268, New York, NY, USA, 2008. Association for Computing Machinery.
- [11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

A Summary of the I/O Equations

$$\begin{aligned}
& V_e k_{21} k_e y + V_e k_{21} y^2 + V_e k_e \dot{y} + k_{12} k_e^2 \dot{y} + 2k_{12} k_e y \dot{y} + k_{12} y^2 \dot{y} \\
& + k_{21} k_e^2 \dot{y} + 2k_{21} k_e y \dot{y} + k_{21} y^2 \dot{y} + k_e^2 \ddot{y} + 2k_e y \ddot{y} + y^2 \ddot{y} = 0 \quad (3)
\end{aligned}$$

$$\begin{aligned}
& k_{21} V_e \int \int \frac{y}{k_e + y} + (k_{12} + k_{21}) \int (y - y_0) \\
& - V_e \int \left(\frac{y}{k_e + y} - \frac{y_0}{k_e + y_0} \right) - \dot{y}_0 t + y - y_0 = 0 \quad (8)
\end{aligned}$$

$$\begin{aligned}
& -y + y_0 - k_{12} \int y - \int \frac{V_e y}{k_e + y} \\
& + \left(\frac{V_e y_0}{k_e + y_0} + \dot{y}_0 + k_{12} y_0 \right) \int e^{-k_{21} t} + k_{21} k_{12} \int (e^{-k_{21} t} \int e^{k_{21} t} y) = 0 \quad (11)
\end{aligned}$$

$$\begin{aligned}
& -y^2 + y_0^2 - 2k_e y + 2k_e y_0 - 2k_e k_{12} \int y - 2k_{12} \int y^2 - 2V_e \int y \\
& + 2k_e \left(\frac{V_e y_0}{k_e + y_0} + \dot{y}_0 + k_{12} y_0 \right) \int e^{-k_{21} t} + 2 \left(\frac{V_e y_0}{k_e + y_0} + \dot{y}_0 + k_{12} y_0 \right) \int y e^{-k_{21} t} \\
& + 2k_e k_{21} k_{12} \int (e^{-k_{21} t} \int e^{k_{21} t} y) + 2k_{21} k_{12} \int (y e^{-k_{21} t} \int e^{k_{21} t} y) = 0 \quad (12)
\end{aligned}$$

$$\begin{aligned}
& -y e^{\int (k_{12} + \frac{V_e}{k_e + y})} + y_0 + \left(\frac{V_e y_0}{k_e + y_0} + \dot{y}_0 + k_{12} y_0 \right) \int e^{\int (k_{12} + \frac{V_e}{k_e + y})} e^{-k_{21} t} \\
& + k_{12} k_{21} \int (e^{\int (k_{12} + \frac{V_e}{k_e + y})} e^{-k_{21} t} \int e^{k_{21} t} y) = 0 \quad (13)
\end{aligned}$$