



**HAL**  
open science

# IDEAS: a parameter identification toolbox with symbolic analysis of uncertainty and its application to biological modelling

Rafael Munoz Tamayo, Béatrice Laroche, Marion Leclerc, Eric Walter

## ► To cite this version:

Rafael Munoz Tamayo, Béatrice Laroche, Marion Leclerc, Eric Walter. IDEAS: a parameter identification toolbox with symbolic analysis of uncertainty and its application to biological modelling. 15. IFAC Symposium on System Identification (SYSID 2009), Jul 2009, Saint Malo, France. 10.3182/20090706-3-FR-2004.00211 . hal-01204242

**HAL Id: hal-01204242**

**<https://hal.science/hal-01204242v1>**

Submitted on 8 Oct 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.

## IDEAS: a Parameter Identification Toolbox with Symbolic Analysis of Uncertainty and its Application to Biological Modelling

R. Muñoz-Tamayo<sup>\*,¶,§</sup> B. Laroche<sup>\*</sup> M. Leclerc<sup>¶</sup> E. Walter<sup>\*</sup>

<sup>\*</sup> UMR8506 Univ Paris Sud-CNRS-SUPELEC, Laboratoire des  
Signaux et Systèmes, 3 rue Joliot Curie 91190 Gif sur Yvette, France  
(Tel: 033-169-851-722; e-mail: Beatrice.Laroche@lss.supelec.fr,  
Eric.Walter@lss.supelec.fr)

<sup>¶</sup> UR910 Institut National de la Recherche Agronomique (INRA)  
Jouy-en-Josas, Unité d'Ecologie et Physiologie du Système Digestif,  
Domaine de Vilvert 78350 Jouy en Josas, France (e-mail:  
marion.leclerc@jouy.inra.fr)

<sup>§</sup> UR341 INRA Jouy-en-Josas, Unité de Mathématiques et  
Informatiques Appliquées, Domaine de Vilvert, 78350 Jouy en Josas,  
France (e-mail: rafael.munoztamayo@jouy.inra.fr)

---

**Abstract:** IDEAS (IDentification and Analysis of Sensitivity) is a Matlab toolbox that automatically estimates parameters of ODE models and assesses their uncertainty, via a symbolic computation of the sensitivity functions. The use of the toolbox is illustrated on a real-life biological model, in the field of microbiology. IDEAS helped improving the model structure, by revealing a lack of practical identifiability that may have not been noticed otherwise.

Keywords: parameter estimation, sensitivity functions, software, symbolic differentiation, uncertainty.

---

### 1. INTRODUCTION

Mathematical models cannot be dispensed with for a better understanding of the behavior of systems in all fields of pure and applied sciences, including biology. For knowledge-based models, there are usually physically meaningful parameters that need to be estimated from experimental data, with the confidence intervals associated to the estimates. Once experimental data are available, the estimation problem is classically translated into an optimization problem, namely finding a feasible parameter vector that best fits the model to the data, in the sense of some appropriate criterion. In the context of systems biology, a number of software packages devoted to modelling, simulation, parameter estimation and sensitivity analysis have been developed over the last 15 years. For example, the softwares COPASI (Hoops et al. [2006]) and SBML-PET (Zi and Klipp [2006]) allow for parameter estimation (among other features). The software package SBML-SAT (Zi et al. [2008]) is devoted to sensitivity analysis. Other softwares have been built as toolboxes to be run in the Matlab<sup>®</sup> environment, which allows the use of functions already implemented in Matlab. They include the recently launched commercial software SimBiology<sup>®</sup> and the freely available toolboxes PEAS (Checchi et al. [2007]), System Biology (SBtoolbox) (Schmidt and Jirstrand [2006]) and PottersWheel (Maiwald and Timmer [2008]).

All of the parameter-estimation softwares mentioned above use a (possibly weighted) quadratic cost function as their measure of optimality. PEAS and PottersWheel

quantify the accuracy of the estimated parameters through a *finite-difference* evaluation of the Fisher information matrix (FIM) or an approximation of this matrix at the estimates. They also provide a Monte-Carlo analysis particularly adapted for models with non-differentiable dynamics.

Even though SBtoolbox and SimBiology estimate parameters, they do not assess their uncertainty. Both toolboxes provide routines for sensitivity analysis, which can be used to assess the accuracy of the estimates, but this not so trivial calculation is left to the user. For sensitivity analysis, SimBiology uses the complex-step approximation method (Squire and Trapp [1998]). SBtoolbox only provides sensitivity analysis at steady state, but an extension to non-steady state trajectories was developed (Schmidt [2007]) to accomplish this task through an interface with the free software CVODES, using the forward or adjoint method (Hindmarsh and Serban [2006]). The current version SBToolbox2 allows forward sensitivity analysis.

This paper describes the Matlab toolbox IDEAS (*IDentification and Analysis of Sensitivity*). It is devoted to knowledge-based ODE models and focuses on parameter estimation and the assessment of the accuracy of the estimates. The estimation is carried out via the maximum-likelihood (ML) approach. As some of the above mentioned toolboxes, IDEAS takes advantage of the functions implemented in Matlab for the optimization and simulation of ODE models. The main originality of IDEAS compared to the previous toolboxes resides in the analytic computation

of the sensitivity functions that are used for the evaluation of the FIM. IDEAS uses the Matlab Symbolic Toolbox to generate automatically a code implementing an exact expression of the differential equations satisfied by the sensitivity functions. These functions are evaluated at the optimal parameter values and used to compute the FIM and assess parameter uncertainty. Our toolbox also provides the parameter correlation matrix and the condition number of the FIM, which are useful indicators of the practical identifiability of the model given the available data. A graphical representation of the local sensitivity analysis is also provided, similar in spirit to that in (Zi et al. [2008]). Another original feature of IDEAS is the opportunity offered to use four different optimization criteria suggested by the ML approach (see Section 2).

The development of IDEAS was initially motivated by our participation to a project of mathematical modelling of carbohydrate degradation by human colonic microbiota (Muñoz-Tamayo et al. [2007]). Our model is a complex set of interacting submodels describing biochemical reactions and transport phenomena. The scarcity of available data is a serious obstacle for the estimation of its parameters. However, *in vitro* bacterial growth experiments and artificial chemostat models have been developed to analyze parts of the complete biological process, corresponding to specific submodels. The goal of IDEAS is to provide a convenient and easy-to-use tool to estimate the parameters of these submodels from available data. The resulting estimates, as well as some characterization of the estimation errors, will be used as prior knowledge for estimating the parameters of the complete model in a Bayesian context.

This paper is organized as follows: Sections 2 and 3 recall the basic concepts of parameter estimation and uncertainty analysis that are used in the software. The software is described in Section 4. In Section 5, it is tested on an actual biological model. Concluding remarks and some key points to improve the software performance and capabilities are presented in Section 6.

IDEAS is freely available for academic usage at <http://www.inra.fr/miaj/public/logiciels/ideas/index.html>.

## 2. PARAMETER ESTIMATION

Consider the state-space model

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}, t), \quad \mathbf{x}(0) = \mathbf{x}_0(\boldsymbol{\theta}), \quad (1)$$

where  $\mathbf{x}(t, \boldsymbol{\theta})$  is the state vector ( $\mathbf{x} : \mathbb{R}^+ \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_x}$ ),  $\boldsymbol{\theta}$  is the parameter vector ( $\boldsymbol{\theta} \in \mathbb{R}^{n_p}$ ), and  $\mathbf{f}$  is a  $C^1$  (continuous with continuous first-order partial derivatives) vector-valued function of the state and parameters ( $\mathbf{f} : (\mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \times \mathbb{R}^+) \rightarrow \mathbb{R}^{n_x}$ ).

The model output is the  $\mathbb{R}^{n_y}$  vector

$$\mathbf{y}_m(t, \boldsymbol{\theta}) = \mathbf{h}(\mathbf{x}(t, \boldsymbol{\theta}), \boldsymbol{\theta}, t), \quad (2)$$

where  $\mathbf{h}$  is a  $C^1$  vector-valued function ( $\mathbf{h} : (\mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \times \mathbb{R}^+) \rightarrow \mathbb{R}^{n_y}$ ).

We consider here the special case of synchronous observations. The vector of data collected at time  $t_i$  is modelled as

$$\mathbf{y}(t_i) = \mathbf{y}_m(t_i, \boldsymbol{\theta}^*) + \boldsymbol{\varepsilon}_i, \quad i = 1, \dots, n_t, \quad (3)$$

with  $n_t$  the number of observation times,  $\boldsymbol{\theta}^*$  the true value of the parameter vector. The measurement errors  $\boldsymbol{\varepsilon}_i$  ( $i = 1, \dots, n_t$ ) are assumed to be independent, homoscedastic,

zero mean and Gaussian, which means that  $\boldsymbol{\varepsilon}_i \sim \mathbf{N}(\mathbf{0}, \boldsymbol{\Sigma})$ . Under these hypotheses, the ML estimator is

$$(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\Sigma}}) = \arg \min_{\boldsymbol{\theta}, \boldsymbol{\Sigma}} L(\boldsymbol{\theta}, \boldsymbol{\Sigma}), \quad (4)$$

where

$$L(\boldsymbol{\theta}, \boldsymbol{\Sigma}) = \frac{n_t}{2} \ln \det \boldsymbol{\Sigma} + \frac{1}{2} \sum_{i=1}^{n_t} [\mathbf{y}(t_i) - \mathbf{y}_m(t_i, \boldsymbol{\theta})]^T \boldsymbol{\Sigma}^{-1} [\mathbf{y}(t_i) - \mathbf{y}_m(t_i, \boldsymbol{\theta})]. \quad (5)$$

IDEAS offers several options corresponding to different hypotheses on the covariance matrix (see Goodwin and Payne [1977] and Walter and Pronzato [1997]). If  $\boldsymbol{\Sigma}$  is known (provided by the user), the ML estimator corresponds to the Gauss-Markov estimator, which minimizes the cost function

$$J_1(\boldsymbol{\theta}) = \sum_{i=1}^{n_t} [\mathbf{y}(t_i) - \mathbf{y}_m(t_i, \boldsymbol{\theta})]^T \boldsymbol{\Sigma}^{-1} [\mathbf{y}(t_i) - \mathbf{y}_m(t_i, \boldsymbol{\theta})]. \quad (6)$$

If  $\boldsymbol{\Sigma}$  is assumed to be proportional to the identity matrix, the ML estimator for  $\boldsymbol{\theta}$  is the unweighted least-squares estimator, which minimizes the cost function

$$J_2(\boldsymbol{\theta}) = \sum_{i=1}^{n_t} [\mathbf{y}(t_i) - \mathbf{y}_m(t_i, \boldsymbol{\theta})]^T [\mathbf{y}(t_i) - \mathbf{y}_m(t_i, \boldsymbol{\theta})], \quad (7)$$

and the ML estimate of the covariance matrix is

$$\hat{\boldsymbol{\Sigma}} = \frac{J_2(\hat{\boldsymbol{\theta}})}{n_t} \mathbf{I}. \quad (8)$$

If  $\boldsymbol{\Sigma}$  is unknown and diagonal, the ML estimator for  $\boldsymbol{\theta}$  minimizes the cost function

$$J_3(\boldsymbol{\theta}) = \sum_{k=1}^{n_y} \frac{n_t}{2} \ln \left[ \sum_{i=1}^{n_t} [y_k(t_i) - y_{m_k}(t_i, \boldsymbol{\theta})]^2 \right], \quad (9)$$

and the ML estimate of the covariance matrix is

$$\hat{\boldsymbol{\Sigma}} = \text{diag}(\hat{\sigma}_1^2, \dots, \hat{\sigma}_{n_y}^2), \quad (10)$$

with

$$\hat{\sigma}_k^2 = \frac{1}{n_t} \sum_{i=1}^{n_t} [y_k(t_i) - y_{m_k}(t_i, \hat{\boldsymbol{\theta}})]^2. \quad (11)$$

Finally, if the covariance matrix is unknown, the ML estimator for  $\boldsymbol{\theta}$  minimizes the cost function

$$J_4(\boldsymbol{\theta}) = \ln \left[ \det \sum_{i=1}^{n_t} [\mathbf{y}(t_i) - \mathbf{y}_m(t_i, \boldsymbol{\theta})][\mathbf{y}(t_i) - \mathbf{y}_m(t_i, \boldsymbol{\theta})]^T \right], \quad (12)$$

and the ML estimate of the covariance matrix is

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n_t} \sum_{i=1}^{n_t} [\mathbf{y}(t_i) - \mathbf{y}_m(t_i, \hat{\boldsymbol{\theta}})][\mathbf{y}(t_i) - \mathbf{y}_m(t_i, \hat{\boldsymbol{\theta}})]^T. \quad (13)$$

In the current version of IDEAS, the minimization of any of these cost functions is performed using the Quasi-Newton method for unconstrained optimization, as implemented in the Matlab Optimization Toolbox.

## 3. PARAMETER UNCERTAINTY

Under a number of technical assumptions that include theoretical identifiability, the covariance matrix  $\mathbf{P}$  of the ML estimator for  $\boldsymbol{\theta}$  is classically approximated by the inverse of the FIM,  $\hat{\mathbf{P}}$ , computed at  $\hat{\boldsymbol{\theta}}$ . However it must be kept in mind that, except in very special cases, approximating the covariance of the parameter estimates by the inverse

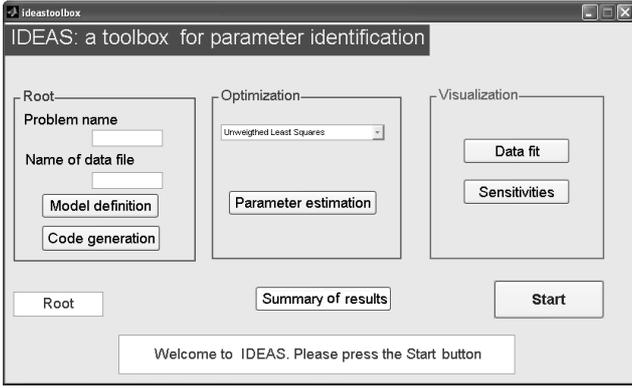


Fig. 1. IDEAS interface

of the FIM is only valid asymptotically, when the number of data points tends to infinity, the statistical hypotheses on the noise are satisfied, and the estimates  $\hat{\theta}$  are close to the true values  $\theta^*$ . When these idealized conditions are far from being satisfied, this evaluation of the uncertainty on the estimates has to be considered with care.

If the covariance matrix is known,  $\Sigma_0 = \Sigma$  and the FIM can be written as

$$\mathbf{F}(\hat{\theta}) = \sum_{i=1}^{n_t} \left[ \frac{\partial \mathbf{y}_m}{\partial \theta} \right]_{(t_i, \hat{\theta})}^T \Sigma_0^{-1} \left[ \frac{\partial \mathbf{y}_m}{\partial \theta} \right]_{(t_i, \hat{\theta})}. \quad (14)$$

When  $\Sigma$  is unknown (see Section 2), a widely used approach is to approximate it by its ML estimate  $\hat{\Sigma}$ . The FIM is then computed taking  $\Sigma_0 = \hat{\Sigma}$  in (14). The square root  $\eta_j$  of the  $j$ th diagonal element of  $\hat{\mathbf{P}}$  is an estimate of the standard deviation of  $\hat{\theta}_j$ , which is used to obtain an approximate 95% confidence interval for  $\theta_j$  as:  $[\hat{\theta}_j \pm 2\eta_j]$ . This is the approach followed in IDEAS.

To evaluate (14), IDEAS computes the first-order sensitivity of the output and state with respect to the parameters.

Let  $\mathbf{s}_j$  denote the first-order sensitivity  $\frac{\partial \mathbf{x}}{\partial \theta_j}$  of the state w.r.t. the parameter  $\theta_j$ . The vector  $\mathbf{s}_j$  is the solution of

$$\dot{\mathbf{s}}_j = \left[ \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right]_{(\mathbf{x}, \theta, t)} \mathbf{s}_j + \left[ \frac{\partial \mathbf{f}}{\partial \theta_j} \right]_{(\mathbf{x}, \theta, t)}, \quad (15)$$

where  $\mathbf{x}$  is computed via (1) and the initial condition is

$$\mathbf{s}_j(0) = \frac{\partial \mathbf{x}(0)}{\partial \theta_j}, \quad j = 1, \dots, n_p. \quad (16)$$

In the current version of IDEAS, the initial conditions on the state are supposed to be known, so  $\mathbf{s}_j(0) = \mathbf{0}$ . According to (2), the sensitivity of the output w.r.t. the parameter  $\theta_j$  is evaluated as

$$\frac{\partial \mathbf{y}_m}{\partial \theta_j} = \left[ \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right]_{(\mathbf{x}, \theta, t)} \mathbf{s}_j + \left[ \frac{\partial \mathbf{h}}{\partial \theta_j} \right]_{(\mathbf{x}, \theta, t)}, \quad \frac{\partial \mathbf{y}}{\partial \theta_j}(0) = \mathbf{0}. \quad (17)$$

Equation (15) is also called *forward sensitivity equation*. The evaluation of the sensitivities can be handled in various ways. They include finite-difference approximation, analytic differentiation, automatic differentiation (Schittkowski [2002]) and the complex-step approximation method (Squire and Trapp [1998]). The most popular method is certainly finite-difference approximation. It is easy to implement, but its accuracy is subordinated to an adequate selection of the step sizes on the parameters.

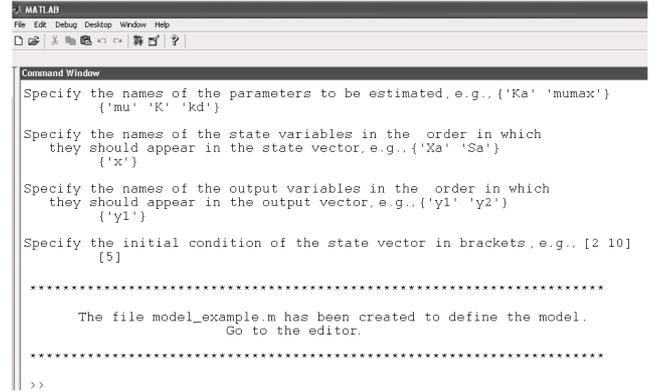


Fig. 2. Naming the parameters, state and output variables of the model

For complex models, hand-made analytic differentiation requires complicated, error-prone and time-consuming manipulations. An interesting option to overcome these difficulties is symbolic manipulation. IDEAS uses an integration of the augmented model (1,15) at  $\hat{\theta}$ . One of its specificities resides in an exact and automatically generated formal evaluation of the right-hand sides of (15) and (17), using the Symbolic Toolbox of Matlab to differentiate  $\mathbf{f}$  and  $\mathbf{h}$  with respect to the state and the parameters. There is thus no need for the user to provide supplementary routines to evaluate the right-hand sides of (15) and (17). The finite-difference approximation method as well as direct integration of the augmented model can become numerically intensive for models with many parameters. The adjoint method (Cao et al. [2003]) becomes especially interesting when the dimension of the parameter vector is large, however its implementation requires a considerable programming effort. This method is not implemented so far in IDEAS.

The FIM is a useful indicator of the practical identifiability of the model given the available data. The condition number of the FIM for the spectral norm, *i.e.*, the ratio of the largest singular value of the FIM to the smallest, provides information on the numerical conditioning of the optimization problem; the higher the condition number, the more difficult the optimization. Moreover, the matrix  $\hat{\mathbf{P}}$  is used to compute an estimate of the parameter correlation matrix Cov. The element  $\text{Cov}(i, j)$  is the approximate correlation coefficient between the  $i$ th and  $j$ th estimated parameters

$$\text{Cov}(i, j) = \frac{\hat{\mathbf{P}}(i, j)}{[\hat{\mathbf{P}}(i, i) * \hat{\mathbf{P}}(j, j)]^{1/2}}. \quad (18)$$

#### 4. SOFTWARE DESCRIPTION

IDEAS is operated through a graphical interface (Fig. 1) and dialog boxes in the Command Window. To illustrate its main features, consider the simplistic model

$$\dot{x} = \mu \frac{x}{K + x} - k_d x, \quad x(0) = 5, \quad (19)$$

$$y_m = x, \quad (20)$$

```
(a)
...
% Write the elements of the vector F, e.g., F(1)= Kh*x
F(1) = mu *x /(K + x) -kd*x;
% Write the elements of the vector H, e.g., H(1) = x*s
H(1) = x;
% save the file and return to the interface

(b)
function [Fv] = examplese(t,X)
global param
...
% State variables
x = X(1);
% State equations
Fv(1,:) = mu*x/(K+x) - kd*x;
% Derivative of F with respect to the parameters
dFdpv(1,1) = x/(K+x);
dFdpv(1,2) = -mu*x/(K+x)^2;
dFdpv(1,3) = -x;
% Derivative of F with respect to the state variables
dFdxv(1,1) = mu/(K+x) - mu*x/(K+x)^2 - kd;
% Sensitivity equations
j = Nx+1;
for k=1:Np;
Fv(j:j+Nx-1,:) = dFdxv*X(j:j+Nx-1) + dFdpv(:,k);
j = j+Nx;
end
```

Fig. 3. Operation of IDEAS: (a) Defining the state equation; (b) Automatically generated code to evaluate the augmented model (1), ( 15)

with  $\mu$ ,  $K$  and  $k_d$  to be estimated. The user provides the data in a .txt file. Afterwards, the names of the parameters and the state variables are requested in the Command Window (Fig. 2). The user defines the mathematical model following the instructions of a template automatically generated (Fig. 3a). From the input files, the software automatically generates the routines required for estimation and visualization, using symbolic differentiation to generate the sensitivity computation routine from the equations provided by the user. Fig. 3b shows the code generated to evaluate the augmented model. The terms  $dFdpv$  and  $dFdxv$  are the automatically generated derivatives of the vector  $f$  in (1) with respect to the parameters and the state, respectively. The Optimization Toolbox and ODE solver are used to find the estimates and compute the FIM. Finally, at the visualization step, plots of the data fit and the sensitivity trajectories are displayed. Figure 4 shows the integration of these steps in a typical execution. All the functions generated are accessible and can be used in other user-defined routines, and modified if needed.

For the solution of (1) and (15), we use a solver adapted to stiff models, based on an implicit method proposed by (Klopfenstein [1971]).

### 5. BIOLOGICAL EXAMPLE

The biological system to be analyzed corresponds to an *in vitro* model to study the kinetics of homoacetogenesis by human colonic bacteria. We refer the reader to (Muñoz-

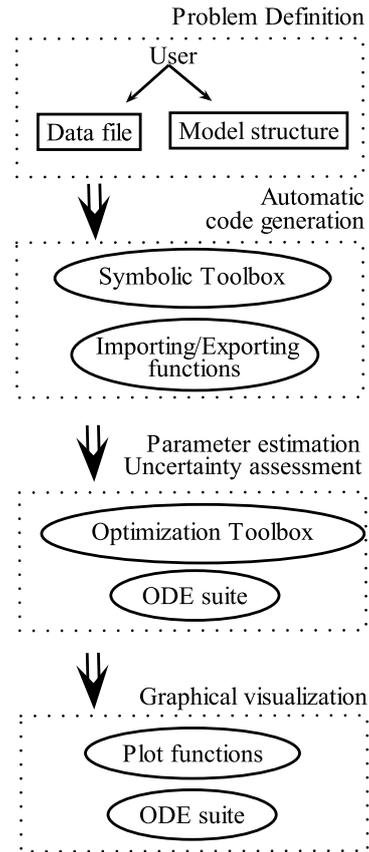


Fig. 4. Steps followed by IDEAS

Tamayo et al. [2008]) for a detailed description of this model, including an identifiability study. The model equations are

$$\dot{x} = \rho_a - k_d x, \quad (21)$$

$$\dot{z} = k_d x - k_l z, \quad (22)$$

$$\dot{s}_{H_2}^l = -\frac{1}{Y_H} \rho_a - k_L a (s_{H_2}^l - K_H R T s_{H_2}^g), \quad (23)$$

$$\dot{s}_{H_2}^g = k_L a (s_{H_2}^l - K_H R T s_{H_2}^g) \frac{V_l}{V_g}, \quad (24)$$

$$\dot{s}_{ac} = \frac{1 - Y_H}{Y_H} \rho_a, \quad (25)$$

where

$$\rho_a = \mu_{\max} \frac{s_{H_2}^l}{K + s_{H_2}^l} x, \quad (26)$$

and the output vector is

$$\mathbf{y}_m = (\alpha(x + z), s_{H_2}^g, s_{ac})^T. \quad (27)$$

The values of the parameters  $\alpha$ ,  $k_L a$ ,  $K_H$ ,  $R$ ,  $T$ ,  $V_l$  and  $V_g$  are known. The parameters to be estimated are the maximum growth rate  $\mu_{\max}$ , the Monod constant  $K$ , the yield factor  $Y_H$ , the bacterial decay factor  $k_d$  and the lysis factor  $k_l$ . Using IDEAS, the minimization of (9) led to a vector of estimates that fitted the data nicely. However, the confidence intervals turned out to be very disappointing (see Table 1, column 2). As could be feared, since this practical identifiability problem is well known (Dochain and Vanrolleghem [2001]), the software indicated an extremely strong correlation between the parameters  $\mu_{\max}$  and  $K$  (Table 2). Other parameters were also strongly correlated. In Fig. 5, we observe the sensitivities

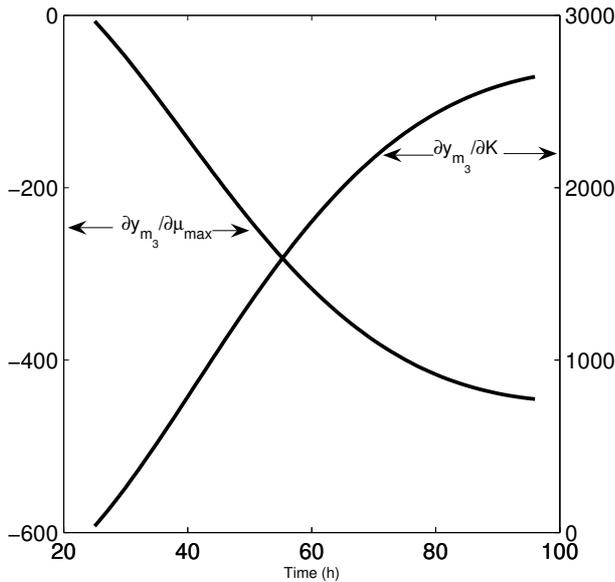


Fig. 5. Sensivities of  $y_{m_3}$  with respect to  $K$  and  $\mu_{\max}$  for the Monod model

of the model output  $y_{m_3}$  with respect to the parameters  $K$  and  $\mu_{\max}$ . The curves show their quasi proportionality, which explains the high correlation between these two parameters. Moreover, simulations of the model with the estimated parameter values indicated that the values taken by the state variable  $s_{H_2}^l$  were always lower than the value of the parameter  $K$ . This suggested a modification of the kinetics expression as  $\rho_a = k_r \cdot s_{H_2}^l x$ , with  $k_r$  the ratio  $\mu_{\max}/K$ . The estimation of the modified model with IDEAS resulted in greatly improved confidence intervals, as shown in Table 1, column 3. The estimates of the parameters that are shared by the two models have the same order of magnitude, but the confidence intervals are smaller for the modified model. The correlation of the estimates for the parameters of the modified kinetics model is shown in Table 3.

Table 1. Estimates with their approximate 95% confidence intervals for the two models

Parameters	Monod	Modified
$K$	$0.025 \pm 0.140$	
$\mu_{\max}$	$0.184 \pm 0.763$	
$k_r$		$5.45 \pm 0.465$
$Y_H$	$0.040 \pm 0.011$	$0.040 \pm 0.011$
$k_d$	$0.031 \pm 0.026$	$0.027 \pm 0.003$
$k_l$	$0.060 \pm 0.113$	$0.071 \pm 0.088$

The data fit with the parameters estimated from the modified-kinetics model were qualitatively as satisfactory as those of the original model (Fig. 6). The curves

Table 2. Correlation matrix for the estimates in the Monod model

	$K$	$\mu_{\max}$	$Y_H$	$k_d$	$k_l$
$K$	1	0.9999	0.4423	-0.9830	0.8357
$\mu_{\max}$		1	0.4542	-0.9848	0.8429
$Y_H$			1	-0.5705	0.8501
$k_d$				1	-0.9147
$k_l$					1

Table 3. Correlation matrix for the estimates in the modified-kinetics model

	$k_r$	$Y_H$	$k_d$	$k_l$
$k_r$	1	0.9418	-0.7552	0.9604
$Y_H$		1	-0.7235	0.9764
$k_d$			1	-0.8492
$k_l$				1

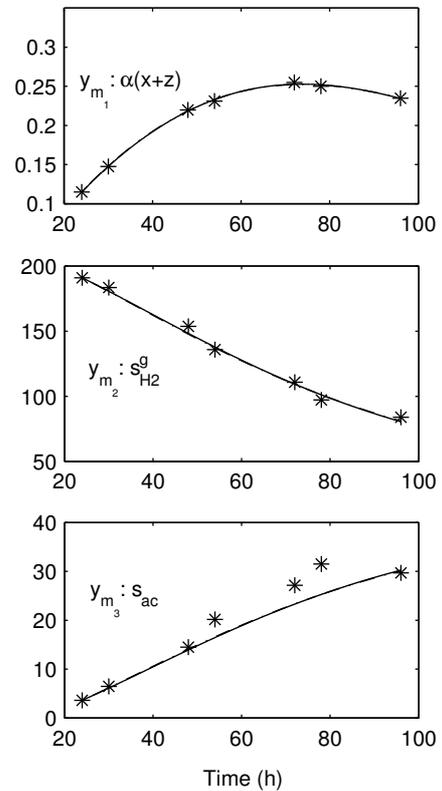


Fig. 6. Data fit: \*: experimental data, solid line: model outputs; the behaviors of the two models are indistinguishable

for the two models are superimposed and cannot be distinguished. Without the assessment of the uncertainty of the estimates, we would have had no reason to prefer the modified model to the original version.

In order to improve the estimation of  $\mu_{\max}$  and  $K$  in the original Monod model, it would be necessary to obtain more informative data.

## 6. CONCLUSIONS AND PERSPECTIVES

We have presented the software IDEAS (IDENTification and Analysis of Sensitivity). IDEAS is implemented in Matlab and requires the Optimization and Symbolic toolboxes. It is dedicated to the estimation of the parameters of ODE models. Parameter estimation is performed in the maximum-likelihood sense, providing the user with four optimization criteria according to the statistical hypothesis made on the measurement errors. The uncertainty of the estimates is assessed through the inverse of the FIM, which is obtained via symbolic differentiation. IDEAS also

plots the sensitivity curves, providing interesting information on the properties of the model. The accurate evaluation of the FIM provided by IDEAS could be exploited for optimal experiment design.

IDEAS generates automatically code that is accessible to the user and can be modified or used in other user-defined routines.

IDEAS was tested on an actual biological modelling problem to study the kinetics of the reductive acetogenesis by a human colonic bacteria. It was instrumental in the detection of a practical identifiability problem and the suggestion of a modification of the parametrization of the model. It has not been tested yet for large-scale models.

IDEAS is still at an early stage of its development. Some extensions are planned in the near future in order to enhance its capabilities. The additional features in the next version will include the possibility to consider the initial conditions as parameters to be estimated. This issue is a serious matter, especially for biomass concentrations, which are technically difficult to measure for mixed culture systems. The next version will also take into account missing data in the output vector components at some measurement times, or asynchronous measurements. This situation is frequently encountered in biological experiments.

Concerning the statistical framework used in IDEAS, the main extensions will be to allow for heteroscedastic errors and for Bayesian estimation. Special interest will be devoted to the maximum a posteriori estimator, which enables one to deal with situations where few experimental data are available.

At the optimization stage, a multi-start routine will be included in order to decrease the sensitivity to the user-provided initial guess on the parameters, and stand-alone optimization routines will be implemented to make IDEAS usable without the Optimization Toolbox. Another additional feature will be the symbolic computation of the gradient of the cost function, which should speed-up optimization. This computation will be based on the sensitivity functions that are already being evaluated by IDEAS.

#### ACKNOWLEDGEMENTS

Rafael Muñoz Tamayo is supported by a PhD fellowship of INRA Departments MICA and MIA.

#### REFERENCES

- Y. Cao, S. Li, P. Linda, and S. Radu. Adjoint sensitivity analysis for differential-algebraic equations: the adjoint DAE system and its numerical solution. *SIAM Journal on Scientific Computing*, 24(3):1076–1089, 2003.
- N. Checchi, E. Giusti, and S. Marsili-Libelli. PEAS: A toolbox to assess the accuracy of estimated parameters in environmental models. *Environmental Modelling & Software*, 22(6):899–913, 2007.
- D. Dochain and P. Vanrolleghem. *Dynamical Modelling and Estimation in Wastewater Treatment Processes*. IWA Publishing, London, 2001.
- G. C. Goodwin and R. L. Payne. *Dynamic System Identification: Experiment Design and Data Analysis*. Academic Press, New York, 1977.
- A. C. Hindmarsh and R. Serban. User documentation for CVODES v2.5.0, 2006. URL <http://www.llnl.gov/CASC/sundials/>.
- S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer. COPASI—a COmplex Pathway simulator. *Bioinformatics*, 22(24):3067–3074, 2006.
- R. W. Klopfenstein. Numerical differentiation formulas for stiff systems of ordinary differential equations. *RCA Review*, 32:447–462, 1971.
- T. Maiwald and J. Timmer. Dynamical modeling and multi-experiment fitting with PottersWheel. *Bioinformatics*, 24(18):2037–2043, 2008.
- R. Muñoz-Tamayo, J. P. Steyer, B. Laroche, and M. Leclerc. Human colon: a complex bioreactor. conceptual modelling for the anaerobic digestion of the functional trophic chain. In *Proc. 11th World Congress Anaerobic Digestion Bio-energy for our Future. Brisbane, Australia*. 6 pages on CD-Rom, 2007.
- R. Muñoz-Tamayo, B. Laroche, M. Leclerc, and E. Walter. Modelling and identification of *in vitro* homoacetogenesis by human-colon bacteria. In *Proc. 16th IEEE Mediterranean Conference on Control and Automation. Ajaccio, France*, pages 1717–1722, 2008.
- K. Schittkowski. *Numerical data fitting in dynamical systems*. Kluwer Academic Publishers, 2002.
- H. Schmidt. Sbaddn: high performance simulation for the systems biology toolbox for MATLAB. *Bioinformatics*, 23(5):646–647, 2007.
- H. Schmidt and M. Jirstrand. Systems Biology Toolbox for MATLAB: a computational platform for research in systems biology. *Bioinformatics*, 22(4):514–515, 2006.
- W. Squire and G. Trapp. Using complex variables to estimate derivatives of real functions. *SIAM Review*, 40(1):110–112, 1998.
- E. Walter and L. Pronzato. *Identification of Parametric Models from Experimental Data*. Springer, London, 1997.
- Z. Zi and E. Klipp. SBML-PET: a systems biology markup language-based parameter estimation tool. *Bioinformatics*, 22(21):2704–2705, 2006.
- Z. Zi, Y. Zheng, A. E. Rundell, and E. Klipp. SBML-SAT: a systems biology markup language (SBML) based sensitivity analysis tool. *BMC Bioinformatics*, 9:342(14 pages), 2008.