User Manual

StochSens - matlab package for sensitivity analysis of stochastic chemical systems

Michał Komorowski[1], Justina Žurauskienė[1], Michael Stumpf[1]

1. Centre for Bioinformatics, Division of Molecular Biosciences, Imperial College London, UK

This is a user manual for the matlab package *StochSens* for sensitivity analysis of stochastic chemical systems. The theory behind this software has been described in the paper [1]:

Sensitivity, identifiability and robustness in stochastic chemical kinetics models; Komorowski M.,Costa M.J., Stumpf M.P.H., and Rand D.A.; Proceedings of the National Academy of Sciences, 2011. We refer to the above paper as to MP and to its Supplementary information as to SI.

This user Manual consists of two sections: *Theory* and *The package*. The *Theory* section aims to introduce basic principles behind the algorithm for calculation of the Fisher Information Matrix (FIM), and explains how the FIM is related to the concepts of sensitivity analysis, robustness and optimal experimental design. Although this section provides necessary information needed for the user it is not a complete guide as a more comprehensive study of the applicability of the package is presented in the MP. *The package* section contains instructions how to use the provided matlab code in order to analyse models using the developed theoretical concepts.

1 Theory

In this section we derive model equations and the likelihood function, and then present a framework for numerical computation of the FIMs. We consider a general system of chemical reactions that consists of N chemical species and interacts in a fixed volume through R reactions. Let $x = (x_1, \ldots, x_N)^T$ be the vector representing the numbers of molecules for the N species and $S = \{s_{ij}\}_{i=1,2...N;\ j=1,2...R}$ be the stoichiometry matrix that describes changes in the population sizes due to each of the reactions, so that occurrence of reaction j results in a change

$$(x_1, ..., x_N) \to (x_1 + s_{1j}, ..., x_N + s_{Nj}).$$

We assume that a reaction of type j occurs at rate $f_j(x, \Theta)$ and is dependent on the vector of all model parameters $\Theta = (\theta_1, ..., \theta_L)$. This specification leads to a model, that in the Linear Noise Approximation (LNA) can be written as

$$\begin{aligned} x(t) &= \varphi(t) + \xi(t) && (1)\\ \dot{\varphi} &= S\ F(\varphi, \Theta, t) && (2)\\ d\xi &= A(\varphi, \Theta, t)\xi + E(\varphi, \Theta, t)dW, && (3) \end{aligned}$$

where

$$F(\varphi, \Theta, t) = (f_1(\varphi, \Theta, t), ..., f_R(\varphi, \Theta, t)) \quad (4)$$

$$\{A(\varphi, \Theta, t)\}_{ik} = \sum_{j=1}^{R} S_{ij} \frac{\partial f_j}{\partial \phi_k} \quad (5)$$

$$E(\varphi, \Theta, t) = S\sqrt{diag(F(\varphi, \Theta, t))}. \quad (6)$$

Equation (2) is an ordinary differential equation that in general does not have an explicit solution but can be solved numerically, whereas equation (3) is a linear stochastic differential equation that has a solution of the form

$$\xi(t) = \Phi(t_0, t)\xi_{t_0} + \int_{t_0}^{t} \Phi(s, t)E(\varphi, \Theta, s)dW(s), \quad (7)$$

where the integral is in the Itô sense and $\Phi(t_0, s)$ is the fundamental matrix of the non-autonomous system of ODEs

$$\frac{d\Phi(t_0, s)}{ds} = A(\varphi, \Theta, s)\Phi(t_0, s), \quad \Phi(t_0, t_0) = I. \quad (8)$$

Given that the initial condition has a multivariate normal distribution (MVN) $x(0) \sim MVN(\varphi(0), V(0))$. equations (1 - 3, 7) imply that $x(t)$ has a multivariate normal distribution

$$x(t) \sim MVN(\varphi(t), V(t)) \quad t > 0, \quad (9)$$

where $\varphi(t)$ is a solution of the macroscopic rate eqaution (MRE), Eqn. (2), with initial condition $\varphi(0)$, and $V(t)$ is a variance at time t. Direct calculations using equations (1 - 3, 7) show that V satisfies

$$\frac{dV(t)}{dt} = A(\varphi, \Theta, t)V(t) + V(t)A(\varphi, \Theta, t)^T + E(\varphi, \Theta, t)E(\varphi, \Theta, t)^T. \quad (10)$$

In order to evaluate the likelihood function, covariances $\mathrm{cov}(x(s), x(t))$ $(t > s)$ are needed, and therefore we calculate these here. As[1] $\langle x(t) \rangle = \varphi(t)$ we have that $\mathrm{cov}(x(s), x(t)) = \langle \xi(t)\xi(s)^T \rangle$ and therefore equation (7) implies

$$\mathrm{cov}(x(s), x(t)) = V(s)\Phi(s,t)^T. \tag{11}$$

Now we can use equations (1-8) to derive the likelihood of experimental data.

1.1 Derivation of the likelihood function

In order to account for various experimental settings we consider three types of data: time series (TS), time-point (TP) and deterministic (DT). For TS measurements at different times are taken from a single trajectory (e.g. following the same cell) and therefore are statistically dependent; in practise TS data are usually obtained using fluorescent microscopy. TP measurements at each time point are taken from different trajectories (e.g. end time points of trajectories following different cells) and are thus independent. These data reflect experimental setups where the sample is sacrificed and the sequence of measurements is not strictly associated with the same sample path (e.g. flow-cytometry, Q-PCR). DT data are defined as a solution of MRE (2) with normally distributed measurement error with zero mean and variance σ_ϵ^2 and refer to measurements averaged over population of cells.

Suppose measurements are collected at times $t_1, ..., t_n$. For simplicity we consider the case where at each time point t_i all components of x_i are measured (see SI for the case with unobserved variables). First let $\mathbf{x}_Q \equiv (x_{t_1}, \ \ldots, x_{t_n})$ be an nN column vector that contains all measurements of type Q, where $Q \in \{TP, TS, DT\}$. It can be shown (see SI for derivation) that

$$\mathbf{x}_Q \sim \mathrm{MVN}(\mu(\Theta), \Sigma_Q(\Theta)) \tag{12}$$

where MVN denotes the multivariate normal distribution,

$$\mu(\Theta) = (\tilde{\varphi}(t_1), ..., \tilde{\varphi}(t_n)), \tag{13}$$

and $\tilde{\varphi}(t)$ is a solution of the MRE (2) such that $\tilde{\varphi}(0) = \varphi_0$ and Σ_Q is a $(nN) \times (nN)$ symmetric block matrix $\Sigma_Q(\Theta) = \left\{\Sigma_Q(\Theta)^{(i,j)}\right\}_{i=1,...,N;j=1,...,N}$ such that

$$\Sigma_Q(\Theta)^{(i,j)} = \begin{cases} \tilde{V}(t_i) & \text{for } \ i = j \quad Q \in \{TS, TP\} \\ \sigma_\epsilon^2 I & \text{for } \ i = j \quad Q \in \{DT\} \\ 0 & \text{for } \ i < j \quad Q \in \{TP, DT\} \\ \tilde{V}(t_i)\Phi(t_i, t_j)^T & \text{for } \ i < j \, Q \in \{TS\} \end{cases} \tag{14}$$

and $\tilde{V}(t)$ is a solution of eq. (10) for a given initial condition $\tilde{V}(0) = V_0$.

[1] We will write $\langle X \rangle$ to denote the the expected value of a random variable X.

1.2 Calculation of the Fisher Information Matrix (FIM)

The multivariate normal distribution (12) of the data $\mathbf{x}_Q$ allows us to express the formula for the FIM in terms of solutions of ordinary differential equations. Suppose first, that a random variable X has an N-variate normal distribution with mean $\mu(\Theta) = (\mu_1(\Theta), ..., \mu_N(\Theta))^T$ and covariance matrix $\Sigma(\Theta)$. We define the FIM for this variable to be $I(\Theta)= \{I(\Theta)_{k,l}\}$

$$I(\Theta)_{k,l} = E_\Theta \left[\left(\frac{\partial}{\partial \theta_k} \log(\psi(X, \Theta)) \right) \left(\frac{\partial}{\partial \theta_l} \log(\psi(X, \Theta)) \right) \right], \tag{15}$$

where $\psi(.)$ is the density function of a multivariate normal distribution with mean $\mu(\Theta)$ and covariance $\Sigma(\Theta)$. As the random variable X is normally distributed the elements $I(\Theta)_{k,l}$ can be also expressed explicitly as

$$I(\Theta)_{k,l} = \frac{\partial \mu}{\partial \theta_k}^T \Sigma(\theta) \frac{\partial \mu}{\partial \theta_l} + \frac{1}{2} trace(\Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_k} \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_l}). \tag{16}$$

Formula (16) indicates that in order to be able to compute FIM, mean μ, covariance Σ and derivatives $\frac{\partial \mu}{\partial \theta_k}$ and $\frac{\partial \Sigma(\Theta)}{\partial \theta_k}$ need to be known. In the LNA equations (13) and (14) give formulae for mean and variance, respectively, of experimental measurements, x_Q. The mean is given as the solution of an ODE, and the variance is either given as a product of solutions of ODEs (TS), directly as a solution of an ODE (10) (TP), or is simply constant (DT). Hence, in order to calculate the FIM we can use the well known theory of calculating derivatives of a solution of an ODE with respect to parameters. We focus on TS data here (see SI for other data types). Let $Y(t)$ be the concatenated vector of $\varphi(t)$ and upper diagonal of the symmetric matrix V

$$Y(t) = (\phi_1(t), ..., \phi_N(t), V_{1,1}(t), ..., V_{N,N}(t), ..., V_{1,2}(t), ..., V_{N-1,N}(t)) \tag{17}$$

and $\tilde{Y}(Y_0, \Theta, t)$ be the concatenation of $\tilde{\varphi}(\varphi_0, \Theta, t)$ and upper diagonal of $\tilde{V}(V_0, \Theta, t)$. Similarly denoting the concatenation of the right hand sides of equations (2) and (10) by W we can write

$$\frac{d}{dt} Y(t) = W(Y(t), \Theta, t). \tag{18}$$

To determine the derivative $Z_k(t) = \frac{\tilde{Y}(t)}{\partial \theta_k}$ we use the fact that it satisfies the following equation

$$\frac{d}{dt} Z_k(t) = J(\tilde{Y}(t), \Theta, t) Z_k(t) + K_k(t), \tag{19}$$

where $J(\tilde{Y}(t), \Theta, t)$ is the Jacobian $\frac{\partial}{\partial Y(t)} W(Y(t), \Theta, t)$ evaluated at the solution $\tilde{Y}(t)$ and $K_k(t)$ is the vector $\frac{\partial}{\partial \theta_k} W(Y(t), \Theta, t)$ also evaluated at $\tilde{Y}(t)$.

The solution of equation (19), $\tilde{Z}(t)$, provides us with $\frac{\partial\tilde{\phi}(t)}{\partial\theta_k}$ and therefore with $\frac{\partial\mu}{\partial\theta_k}$. Similarly $\tilde{Z}(t)$ contains diagonal elements of $\frac{\partial\Sigma}{\partial\theta_k}$.

Non-diagonal elements of $\frac{\partial\Sigma}{\partial\theta_k}$ can be computed from diagonal elements using the recursive relation

$$\frac{\partial}{\partial\theta_k}\Sigma^{(i,j+1)} = \frac{\partial}{\partial\theta_k}\left(\Sigma^{(i,j)}\Phi(t_j,t_{j+1})^T\right) = \frac{\partial}{\partial\theta_k}\left(\Sigma^{(i,j)}\right)\Phi(t_i,t_{i+1})^T + \Sigma^{(i,j)}\frac{\partial}{\partial\theta_k}\left(\Phi(t_i,t_{i+1})^T\right). \tag{20}$$

from elements $\Phi(t_j,t_{j+1})$, $\Sigma^{(i,i)}$, $\frac{\partial}{\partial\theta_k}\Sigma^{(i,i)}$ that are given by equations (8), (14) and (19) respectively. To simplify notation denote $\Xi_k(s,t) = \frac{\partial\Phi(s,t)}{\partial\theta_k}$. As $\Phi(s,t)$ is a solution of an ODE we use similar techniques as in equations (19) and write $\Xi_k(s,t)$ as a solution of the differential equation

$$\frac{d\Xi_k}{dt}(s,t) = A(\tilde{\varphi},\Theta,t)\Xi(s,t) + M_k(t), \tag{21}$$

where

$$M_k(t) = \frac{\partial}{\partial\theta_k}(A(\tilde{\varphi},\Theta,t)\Phi(s,t)) = \left(\frac{\partial}{\partial\theta_k}A(\tilde{\varphi},\Theta,t) + (\frac{\partial}{\partial\varphi}A(\varphi,\Theta,t)_{\varphi=\tilde{\varphi}})\frac{\partial\tilde{\varphi}}{\partial\theta_k}\right)\Phi(s,t), \tag{22}$$

and $\Xi(s,s) = 0$ for all s. To summarise, for the experimental data distribution (12) the FIM (16) can be computed using equations (19 - 22):

- the parameter derivative of the mean, $\frac{\partial\mu(\Theta)}{\partial\theta_k}$, can be extracted from a solution of (19)
- diagonal elements of the parameter derivatives of the variance, $\frac{\partial\Sigma_{TS}(\Theta)}{\partial\theta_k}$, can be extracted from a solution of (19)
- non-diagonal elements of parameter derivatives of the variance, $\frac{\partial\Sigma_{TS}(\Theta)}{\partial\theta_k}$, are given by formula (20), which involves (21) and (22).

1.2.1 FIM and sensitivity

Having demonstrated how the FIM can be computed we now show how it relates to sensitivity of model parameters. The classical sensitivity coefficient for an observable Q and parameter θ is

$$S = \frac{\partial Q}{\partial\theta}.$$

The behaviour of a stochastic system is defined by observables that cannot be measured exactly in a reproducible way but are instead drawn from a probability distribution. The FIM is a measure of how this distribution changes in response to infinitesimal changes in parameters. It follows, for instance, from its role as the expected Hessian of the log-likelihood at a maximum. Suppose that $\ell(\Theta; X) = log(\psi(X, \Theta))$ and $\ell(\Theta) = -E\left[\ell(\Theta; X)\right]$. Than, as it is well-known

$$I(\Theta)_{k,l} = -E\left[\frac{\partial^2 \ell(\Theta; X)}{\partial\theta_k \partial\theta_l}\right]. \tag{23}$$

Thus the FIM is the expected Hessian of $\ell(\Theta, X)$. Therefore, if Θ^* is the maximume likelihood estimate of a parameter there is a $L \times L$ orthogonal matrix C such that, in the new parameters $\theta' = C\ (\Theta - \Theta^*)$,

$$\ell(\Theta) \approx \ell(\Theta^*) - \frac{1}{2}\sum_{i=1}^{L} \lambda_i\ {\theta'_i}^2 \tag{24}$$

for Θ near Θ^*. From this it follows that the λ_i are the eigenvalues of the FIM and that the matrix C diagonalises it. If we assume that the λ_i are ordered so that $\lambda_1 \geq \cdots \geq \lambda_L$ then it follows that around the maximum the likelihood is most sensitive when θ'_1 is varied and least sensitive when θ'_L is varied, and λ_i is a measure of this. Since $\theta'_i = \sum_{j=1}^{L} C_{ij}(\theta_j - \theta^*_j)$ we can regard $\mathcal{S}_{ij} = \lambda_i^{1/2}\ C_{ij}$ as the contribution of the parameter θ_j to varying θ'_i and thus

$$\mathcal{S}_j^2 = \sum_{i=1}^{L} \mathcal{S}_{ij}^2 \tag{25}$$

can be regarded as a measure of the sensitivity of the system to θ_j. It is sometimes appropriate to normalise this and instead consider

$$\mathcal{T}_j = \frac{\mathcal{S}_j^2}{\sum_{i=1}^{L} \mathcal{S}_i^2}. \tag{26}$$

1.2.2 FIM and Kullback-Leibler divergence

In the context of applicability of the Fisher Information to study sensitivity of a probability distribution it is also insightful to be aware of its direct link with the Kullback-Leibler (KL) divergence. Suppose, we want to calculate KL distance between two distributions $\psi(\Theta_0, X)$

and $\psi(\Theta, X)$. Using the Taylor expansion we have

$$\begin{aligned} & \int \psi(\Theta_0, x) \log\left(\frac{\psi(\Theta_0, x)}{\psi(\Theta, x)}\right) dx = \qquad (27) \\ = & \int \psi(\Theta_0, x) \left(-\frac{\partial \log(\psi(\Theta_0, x))}{\partial \Theta_0}(\Theta - \Theta) - \frac{1}{2}\frac{\partial^2 \log(\psi(\Theta_0, x))}{\partial \Theta_0^2}(\Theta_0 - \Theta)^2 + O(|\Theta_0 - \Theta|^3)\right) = \\ = & \frac{1}{2} I(\Theta_0)(\Theta_0 - \Theta)^2 + O(|\Theta_0 - \Theta|^3) \end{aligned}$$

for Θ_0 close to Θ. This representation shows that locally KL divergence behaves as the FI and, therefore, FI is a good measure of local sensitivity to parameter changes.

1.2.3 FIM and robustness

The above relation, also, clearly demonstrates how the FIM can be used to study robustness of a system. Robustness is usually understood as persistence of a system to perturbations to external conditions. Sensitivity considers perturbation in a single parameter whereas robustness takes into account simultaneous changes in all model parameters. Near to a considered parameter value Θ_0 the regions in the parameter space with KL divergence smaller that 2ε are approximately the ellipsoids $NS(\Theta_0, \varepsilon)$ given by the equation

$$NS(\Theta_0, \varepsilon) = \left\{\Theta : (\Theta - \Theta_0)^T I(\Theta_0)(\Theta - \Theta_0) < \varepsilon\right\}. \qquad (28)$$

Sets NS are called neutral spaces as they describe regions of parameter space in which a system's behaviour does not undergo significant changes and arise naturally in the analysis of robustness.

1.2.4 Relation between FIM, parameter identifiability and experimental design

In the statistical literature FIM is a primary tool to study parameter identifiability and serves as a tool to asses optimality of experiments. There exist various definitions of parameter identifiability and here we consider local identifiability. The parameter vector Θ is said to be (locally) identifiable if there exists a neighbourhood of Θ such that no other vector Θ^* in this neighbourhood gives raise to the same density as Θ. Formula (24) implies that Θ is (structurally) identifiable if and only if FIM has a full rank [2]. Therefore the number of non-zero eigenvalues of FIM is equal to the number of identifiable parameters, or more precisely, to the number of identifiable linear combinations of parameters.
The FIM is also a key tool to construct experiments in such a way that the parameters can be estimated from the resulting experimental data with the highest possible statistical

quality. The theory of optimal experimental design uses various criteria to asses information content of experimental sampling methods [3]; among the most popular are the concepts of D-optimality that maximises the determinant of FIM, and A-optimality that minimise the trace of the inverse of FIM.

1.3 Limitations on applicability

The model analysis based on the FIM studies model behaviour around fixed parameters values that are assumed to be known, and therefore is local in parameter space. In consequence the drawn conclusions do not have to be true if the parameter values are changed. Despite this limitation local sensitivity measures has gained popularity due to their simplicity and computational efficiency [4] . Being able to analyse a model locally allows for a global analysis if the local method is computationally feasible to be performed for a large number of parameter values. Then results for a sample of parameters can be combined to provide a global picture of model behaviour [4].

The validity of the Linear Noise Approximation used to compute the FIMs results from its underling assumption about the large number of molecules present in a studied system [5]. Although in a large class of models it delivers very satisfactory results one can easily provide examples where the approximation can not be used. Such limitations can occur either if the number of molecules is low enough so that the LNA implied probability density assigns significant probability to negative values of system's state [6]; or if a system is multistable and therefore linearisation allows for the analysis around a single steady state; or when fixed point or a trajectory is unstable [7, 8]. These limitations should be taken into account and verified when using the package.

2 The package

In this section we describe how to use provided matlab packed to utilise theoretical concepts of the previous section. In particular, how show to install the package, define models, calculate FIM. We pay special attention to how the computed FIM can be used to study a system in the context of sensitivity, robusness and optimal experimental design.

2.1 Installing the package

The package is provided as a zip archive 'stochsens.zip' and requires Matlab R2009a (or higher) with the Symbolic Maths Toolbox. The zip archive should be copied to the directory where one wishes to install the package, e.g. *InstDir*. After unzipping, the folder will contain files with matlab functions and the folder *models* containing examples.

2.1.1 Defining a model

All models are stored in the folder *InstDir/models*, each in a separate folder *modelname*. Each model is defined by the following files

- *modelname_stoich.txt*
- *modelname.par*
- *modelname_rates.m*

that are stored in *InstDir/models/modelname*. File *modelname_stoich.txt* is a text file with a stoichiometry matrix (species in rows and reactions in columns).
File *modelname.par* is also a text file with parameter definitions and values. It should contain three columns and as many rows as parameters. Each line should have the following format

parametername parametervalue "optional description"

The file *modelname_rates.m* is a file containing rates of reactions in the same sequence as defined in stoichiometry matrix. It should have the form of a matlab function named *modelname_rates()* with arguments *(x, par, t)*, where x is a vector describing systems state, *par* is a vector of model parameters, and t is time. The output of this function is a vector of reaction rates.

In order to define a new model, a folder *newmodel* containing these three files should be placed in *InstDir/models/*

2.2 Creating the model

Before a model can be analysed using the implemented functions, a number of matrices must be generated from the definition files. This is done using the function *create()*. To generate a model, enter the following

```
cd InstDir;
create('modelname');
```

This will create a folder *models/modelname/symbolic* with the following files

- *modelname_all_equations.m* right hand side of equation (18);
- *modelname_MRE_jacobian.m* jacobian A of macroscopic rate equation (see e.q. (5));
- *modelname_all_equations_jacobian_dvar.m* jacobian J (see eq. (19));
- *modelname_all_equations_jacobian_dpar.m* vectors K_j (see eq. (19));
- *modelname_jacobianMRE_jacobian_dvar.m* matrix A (see eq. (21));
- *modelname_jacobianMRE_jacobian_dpar.m* matrices $\frac{\partial}{\partial\theta_j}A(\tilde{\varphi},\Theta,t)$ (see eq. (22)).

If this folder already exists, it will be overwritten.

2.3 Plotting model trajectories

Model trajectories together with their standard deviations can be plotted using the function *plottraj()*. This function has the following arguments

- *name* is a *modelname* (e.g. name = 'LV');
- N is a number of observations (e.g. N = 10);
- $freq$ is a time distance between observations (e.g. freq = 0.1);
- $init\_T$ is a time of an initial observation (e.g. T = 1);
- $y0$ is a vector of initial conditions of variable Y defined in equation (27) of SI);
- *obsv* is a vector of indices of observed variables (e.g. obs = [1, 2]).

In order to plot, type

```
plottraj(name, N, freq, init_T, y0, obsv);
```

2.4 Computation of the Fisher Information Matrix (FIM)

The function *Fisher()* can be used to compute the FIM for a defined model. To compute the FIM type

```
F = Fisher(name, N, freq, init_T, y0, obsv, sigma, Type, LL);
```

where *name, N, freq, init_T, y0* and *obsv* were introduced in the previous section, and *sigma, Type* and *LL* are:

- *sigma* is a variance of the measurement error and is only used to compute the FIM for a deterministic system (e.g. sigma = 10);
- *Type* describes a kind of model for which the FIM should be calculated; use 'TS' for time series data, 'TP' for time point data, 'DT' for deterministic data and 'All' for all three types;
- *LL* is a type of parametrisation (LL = 'TRUE' for logarithmic, LL = 'FALSE' for standard).

The output of this function is either a matrix (for Type = 'TS', 'TP', 'DT') or a list of three matrices (for Type = 'All') that correspond to TS, TP and DT data, respectively.

2.5 Visualising the FIM

The functions to visualise the FIM are described in Table 1.

2.6 Calculating sensitivity coefficients

The functions to visualise sensitivity coefficients based on the FIM are described in Table 2.

3 Examples

In this section we use the example of the Lotka-Volterra system in order to demonstrate the use of the package. This example is implemented in the file *LV_script.m*

Function name	Arguments	Output	Example
plotdFMcc()	name, rx, ry, A, B, i, j, LabelA, LabelB	contour plot for matrices A and B and parameters i and j	Figure 2
plotdFM()	name, rx, ry, A, B, i, j	heat map for matrix A and contour plot for matrix B for parameters i and j	Figure 3
plotdFMAll()	name, rx, ry, A, B	heat map for matrix A and contour plot for matrix B for all parameters	Figure 4
plotdFMAllcc()	name, rx, ry, A, B, LabelA, LabelB	contour plots for matrices A and B for all parameters	Figure 5

Table 1: Functions to visualise the FIM. Arguments are explained as follows: name is a string for *modelname*; rx,ry maximal values for x and y axis, respectively; A and B are FIMs to be plotted; i and j are indices describing parameters to be plotted.

3.1 Lotka-Volterra equation

The system incorporates two variables (y_1, y_2) and three reactions:

$$R1 \quad : \quad y_1 \xrightarrow{\alpha y_1} 2y_1 \tag{29}$$

$$R2 \quad : \quad y_1 + y_2 \xrightarrow{\beta y_1 y_2} 2y_2 \tag{30}$$

$$R3 \quad : \quad y_2 \xrightarrow{\gamma y_2} \emptyset \tag{31}$$

Function name	Arguments	Output	Example
sensitivities()	A	vector with sensitivity coefficient based on matrix A	-
sensitivitiesAll()	name, F	plot with sensitivity coefficients for TS, TP and DT data types based on list of three FIMs F	Figure 6
diaginvl()	name, F	plot of diagonal elements of the inverses of three FIMs in the list F	Figure 7
decomp()	name, F	plot of contributions of individual parameters into eigenvalues of three FIMs in the list F	Figure 9

Table 2: Functions for sensitivity analysis. Arguments are explained as follows: name is a string for *modelname*; A is a single FIM; F is an output of the function *Fisher()* and provides a list of FIMs for TS, TP and DT data, respectively.

and in terms of stoichiometry matrix and vector of reaction rates is defined as follows

$$S = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}, \tag{32}$$

$$F(y) = (\alpha y_1, \beta y_1 y_2, \gamma y_2). \tag{33}$$

3.1.1 Defining the model

Suppose we want to define the model of the Lotka-Volterra (LV) system for the *StochSens package.* This requires the following steps

- in folder *models* create folder *LV*
- in folder *LV* create file *LV_stoich.txt* containing

```
1  -1   0
0   1  -1
```

- in folder *LV* create file *LV.par* containing

```
\alpha 10 "pray birth rate"
\beta 0.01 "pray death rate"
\gamma 8 "predator death rate"
```

- in folder *LV* create file *LV_rates.m* containing

```
function R = LV_rates(y, par)
R = [par(1)*y(1);
 par(2)*y(1)*y(2);
 par(3)*y(2);];
end
```

- Subsequent elements of input vector `par` must corresponds to the parameters defined in the following lines of the file *LV.par*.

3.1.2 Creating the model

In order to create the objects needed to numerically compute the FIM for this model, function *create()* with argument *LV* should be used

```
create('LV')
```

Function *create()* will produce a subfolder *symbolic* within the folder *LV* that contains the following files

- *LV_all_equations.m* right hand side of equation (27) of SI;
- *LV_MRE_jacobian.m* jacobian A (see eq. (6) of SI);
- *LV_all_equations_jacobian_dvar.m* jacobian J (see eq. (28) of SI);
- *LV_all_equations_jacobian_dpar.m* vectors K_j (see eq. (28) of SI);
- *LV_jacobianMRE_jacobian_dvar.m* matrix A (see eq. (30) of SI);
- *LV_jacobianMRE_jacobian_dvar.m* matrices $\frac{\partial}{\partial\theta_j}A(\tilde{\varphi},\Theta,t)$ (see eq. (31) of SI).

3.1.3 Calculating FIM

After the above files have been created the FIM functions for model analysis can be used. Model trajectories can be plotted using function *plottraj()*. Function *Fisher()* should be used for the purpose of computing the FIM. The applicability of these functions is demonstrated below.
First, we define the arguments of the function

```
name = 'LV'; % name of the model
y0 = [1000, 1000 , 1, 1, 0]; % initial condition
N = 15; % number of observations
freq = 0.1; % time distance between observations
init_T = 2; % time of an initial observation
obsv = [1, 2];  % indices of observed variables
sigma = 10;  % variance of measurement error
```

Then the subsequent functions can be applied

```
plottraj(name, N, freq, init_T, y0, obsv) % see figure (see Figure 1)
F = Fisher(name, N, freq, init_T, y0, obsv, sigma, 'All', 'TRUE'); % a list
% of 3 FIMs for TS, TP and DT is returned
```

If it is required to compute the FIM only for one kind of model (e.g. TS) then one should execute

```
FTS = Fisher(name, N, freq, init_T, y0, obsv, sigma, 'TS', 'TRUE'); % a
% single matrix is returned
```

3.2 Visualising FIM

Visualisation of a FIM is usually done by plotting quadratic forms that corresponds to the FIM for pairs of the parameters. In our package it can be done using functions *plotdFMcc(), plotdFM(), plotdFMAll()* and *plotdFMAllcc()*.

To plot the FIM as a contour plot (Figure 2) for the first and second parameters, and TS and DT kind of data, execute function *plotdFMcc()*. First of all, let define the arguments

```
rx = 1;  % maximal x argument for a quadratic form
ry = 2;  % maximal y argument for a quadratic form
i = 1; % index of a first parameter
j = 2; % index of a second parameter
FIM_TS = F{1};
FIM_DT = F{2};
plotdFMcc(name, rx, ry, FIM_TS, FIM_DT, i, j, 'TS', 'DT');
```

The FIM can be plotted as a contour plot and/or heatmap (Figure 3); for the first and second parameters, and TS and DT data, execute the following function

```
plotdFM(name, rx, ry, FIM_TS, FIM_DT, i, j);
```

For all pairs of parameters, given TS and DT kind of data, FIMs can be plotted as contour plots (Figure 5). To produce these plots, type

```
plotdFMAllcc(name, rx, ry, FIM_TS, FIM_DT, 'TS', 'DT');
```

For all pairs of parameters for given TS and DT data, FIMs can be visualised as a combination of a contour plots and heatmaps (Figure 4) by executing function *plotdFMALL()*

```
plotdFMAll(name, rx, ry, FIM_TS, FIM_TP);
```

3.3 Calculating sensitivity coefficients

Sensitivity coefficients can be computed from the FIM using function *sensitivities()*.

```
SC1 = sensitivities(F{1});
```

To plot sensitivity coefficients for all three types of data (Figure 6) use the function

```
sensitivitiesAll(name, F);
```

and

```
diaginv(name, F);
```

to compare diagonal elements of the inverse matrices of FIMs (Figure 7).

Contributions of individual parameters into each eigenvalue (Figure 9) for all three data types are obtained using the function *decomp()*

```
decomp(name, F);
```

3.4 Robustness analysis using FIM

As described in the section 1.2.2, the FIM allows us to understand the influence of parameter perturbations on the behaviour of the system. Using the concept of Neutral Spaces we can identify regions of parameter space that result with small and large deviations from an initial state. Figure 4 presents NS's for LV system for each pair of the three parameters. Heat maps represent NS's for the stochastic TS model and contour plots for the deterministic DT model. For the pair α, β NS's of both versions of the model overlap and the two eigen-directions are almost perpendicular and with substantially different lengths of radii. Therefore a change in α will change significantly the behaviour of the system in contrast to β. Perpendicularity of the two parameters implies that a change in one of the parameter cannot be compensated by a change in the other parameter. The plot of the NS of α, γ exhibits a difference how the deterministic model (contour plot) and a stochastic model (heat map) respond to changing their values. In the deterministic model increase in γ can be compensated by decrease of α whereas in the stochastic case compensation requires increase of α.

3.5 Interpretation of sensitivity analysis

The sensitivity coefficients $\mathcal{T}_j$ given by formula (26) allow us to quantify how much the behaviour of a system changes when each of its parameters is changed. They represent a change in the behaviour resulting from a change in a single parameter relative to the sum of changes caused by each individual parameter. Figure 6 demonstrates that, for used parameter values, in the LV system almost 100% of variability results from perturbation in parameter β, regardless of which model version (TS, TP, DT) is considered. This is confirmed by looking at the inverses of the diagonal elements of the FIM (Figure 7) which represent Cramér-Rao bounds of model parameters and asymptotic variances of parameter estimates. The sensitive parameter β has variance close to zero and therefore is easy to infer, whereas the insensitive parameters α and γ have much higher variances and are more difficult to estimate.

3.6 Identifiability and experimental design using FIM

As discussed in section 1.2.4 the FIM is a classical criterion for local parameter identifiability. Having calculated FIM the number of identifiable parameters can be found by counting the number of its non-zero eigenvalues

```
sum(eig(F)>0)
```

that is equal to 3 in the studied LV system.

The theoretical possibility to infer parameters does not take into account the quality of obtained estimates, which depends on how an experiment was performed. The FIM allows us to construct experiments in such a way that the parameters can be estimated from the resulting experimental data with the highest possible statistical quality. One criterium of optimal experimental design (D-optimality) postulates to maximise determinant of the FIM what is equivalent to minimising the variance of obtained estimates.

Here we demonstrate how this can be done by finding an optimal sampling frequency, an experimental parameter that can be easily controlled, for LV system using stochastic TS model. Figure 8 shows determinant of the FIM for a range of different time intervals between observations $\Delta = 0.03...0.6$, for fixed number of time points $n = 15$. The maximal amount of information, that corresponds to lowest variances of estimates, is collected when measurements are taken approximately every 0.35 time units.

4 Additional Examples

4.1 Michaelis-Menten enzyme kinetics

A script with the analysis of Michaelis-Menten enzyme kinetics model *EnzStat.m* is distributed together with the package. Parameters used to produce Figure 1 in the main paper are presented in Table 1.

4.2 Single gene expression and p53 system models

Examples of a single gene expression model and the p53 system are been also included in folder *models*. Similar analyses as presented above for Lotka-Volterra model are being provided in scripts *singlegene_script.m* and *p53_script.m*.

Param.	Value
k_0	0.4
k_1	0.4
k_2	10
k_b	60
k_d	2

Table 3: Parameters used for Michaelis-Menten enzyme kinetics.

References

1. Komorowski, M., M. Costa, D. Rand, and M. Stumpf, 2011. Sensitivity, robustness, and identifiability in stochastic chemical kinetics models. *Proceedings of the National Academy of Sciences* 108:8645.

2. Rothenberg, T., 1971. Identification in parametric models. *Econometrica: Journal of the Econometric Society* 39:577–591.

3. Emery, A., and A. Nenarokomov, 1998. Optimal experiment design. *Measurement Science and Technology* 9:864–876.

4. Saltelli, A., and I. ebrary, 2008. Global sensitivity analysis: the primer. Wiley Online Library.

5. Kurtz, T. G., 1972. The Relationship between Stochastic and Deterministic Models for Chemical Reactions. *The Journal of Chemical Physics* 57:2976–2978.

6. Elf, J., and M. Ehrenberg, 2003. Fast Evaluation of Fluctuations in Biochemical Networks With the Linear Noise Approximation. *Genome Res.* 13:2475–2484.

7. R., T., K. H., K. T. J., and A. K., 2004. Multivariate analysis of noise in genetic regulatory networks. *Journal of Theoretical Biology* 229:501–521.

8. Ito, Y., and K. Uchida, 2010. Formulas for intrinsic noise evaluation in oscillatory genetic networks. *Journal of theoretical biology* 267:223–234.

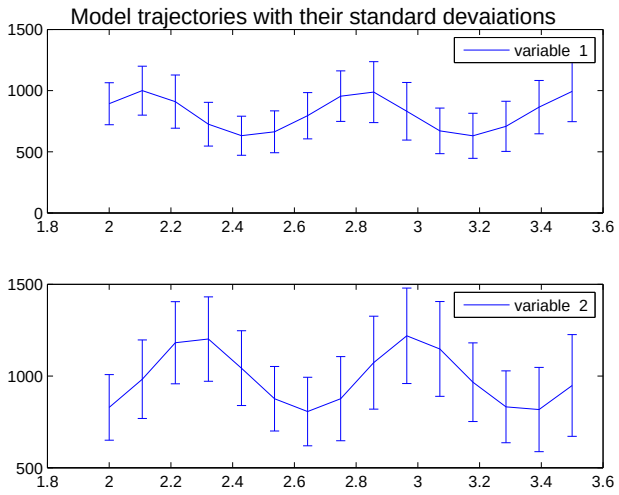

Figure 1: Output of the function *plottraj()* for Lotka-Volterra model.

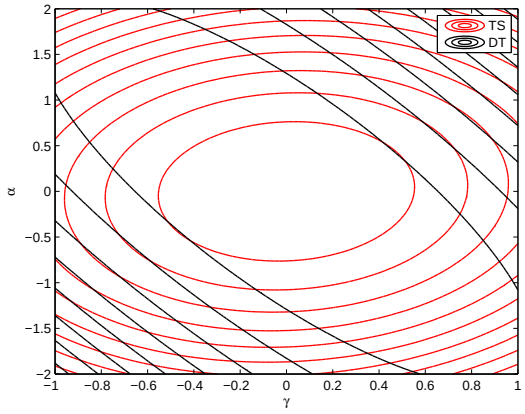

Figure 2: Output of the function *plotdFMcc()* for Lotka-Volterra model.

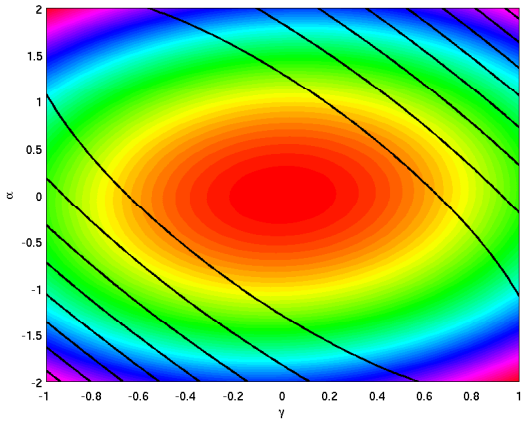

Figure 3: Output of the function *plotdFM()*

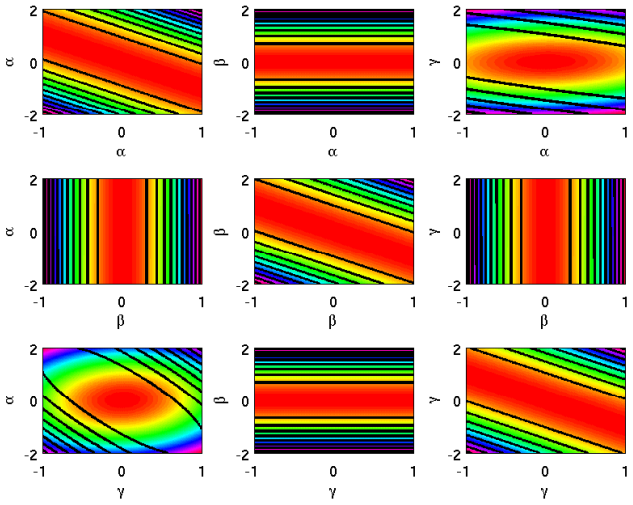

Figure 4: Output of the function *plotdFMAll()* for Lotka-Volterra model.

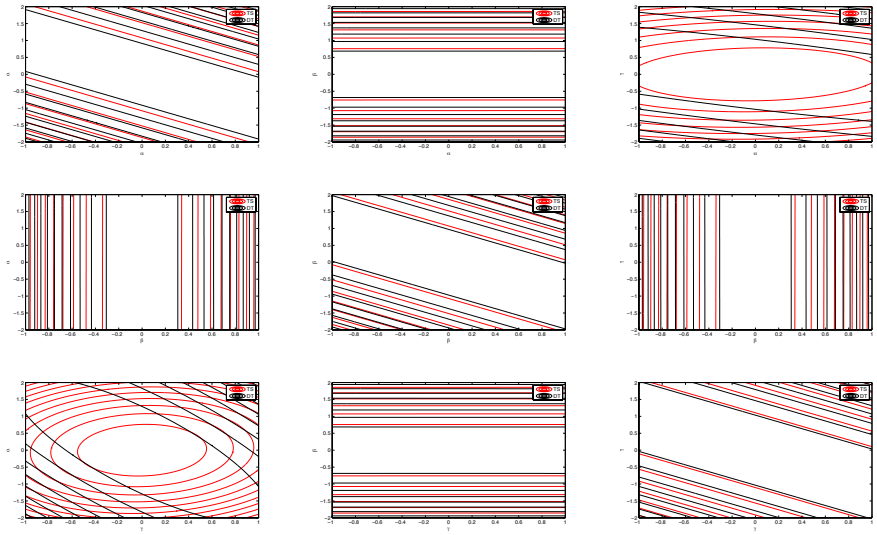

Figure 5: Output of the function *plotdFMAllcc()*

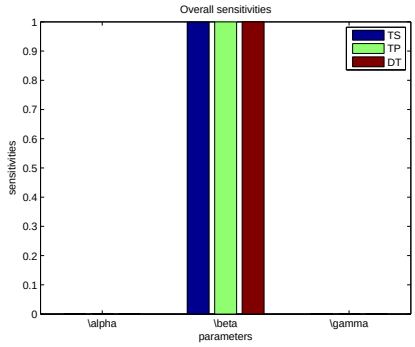

Figure 6: Output of the function *sensitivitiesAll()* for Lotka-Volterra model.

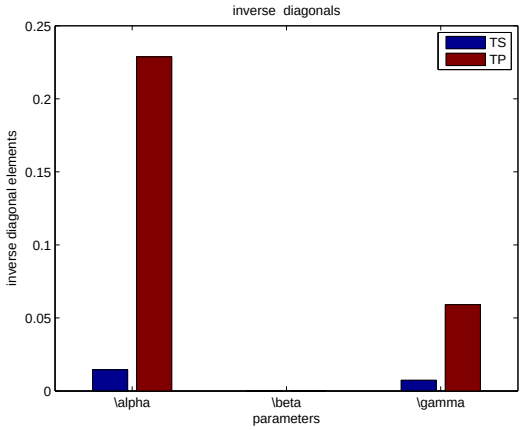

Figure 7: Output of the function *diaginv()* for Lotka-Volterra model.

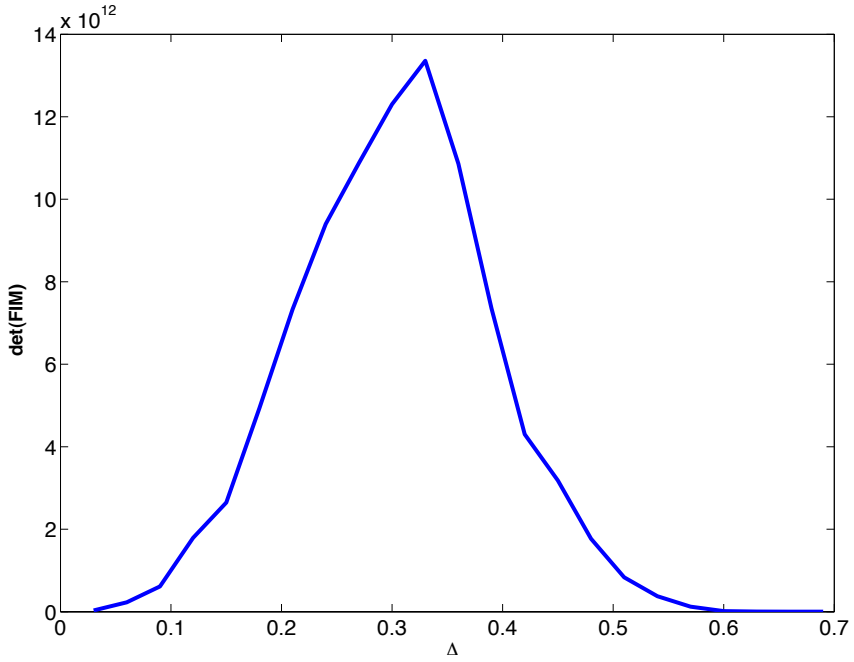

Figure 8: Determinant of the FIM for the LV system as a function of the time interval between observation Δ, computed for the fixed number of measurements $n = 15$.

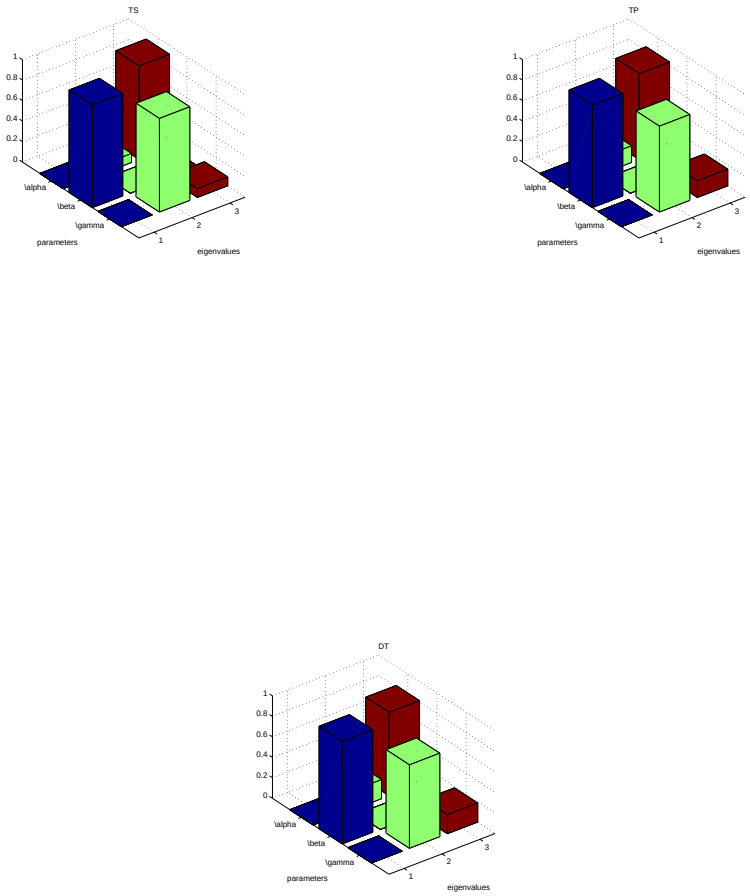

Figure 9: Output of the function *decomp()* for Lotka-Volterra model.