A SYSTEM FOR THE AUTOMATION OF ONE-AT-A-TIME SENSITIVITY ANALYSIS IN ODE SYSTEMS

Matheus Fajardo Galvão¹

Marcelo Lobosco²

Abstract: Sensitivity analysis is a widely used tool in computing modeling, being used for distinct purposes, such as a better understanding of the relationship between parameter values in a mathematical model and its results, and the identification of parameters whose change in their values implies in a larger variation in the results of a model, among others purposes. Several approaches can be used to perform sensitivity analysis. One of the simplest is one-at-a-time, which analyzes the impact of changing a single parameter on a model while keeping the others one fixed. Depending on the number of parameters and the variations adopted, the sensitivity analysis of a large model may generate a large volume amount of results and it can be hard to identify the most sensitive parameters of the model. This paper presents a tool that automates the identification of these parameters.

Keywords: Sensitivity analysis. One-at-a-time. Ordinary differential equations.

Resumo: A análise de sensibilidade é uma ferramenta muito utilizada em modelagem computacional, servindo para distintas finalidades, como o melhor entendimento da relação entre os valores dos parâmetros em um modelo matemático e os seus resultados, e a identificação dos parâmetros cuja alteração de valores implica em uma maior variação nos resultados de um modelo, dentre outros. Diversas abordagens podem ser utilizadas para realizar a análise de sensibilidade. Uma das mais simples é a um por vez, em que se analisa o impacto da alteração de um único parâmetro em um modelo, mantendo-se os demais fixos. Dependendo do número de parâmetros de um modelo e de variações adotadas, a análise de um grande volume de resultados pode ser necessária para a identificação dos parâmetros mais sensíveis de um modelo. Neste trabalho é apresentada uma ferramenta que automatiza a identificação destes parâmetros em um sistema de EDOs.

Palavras-chave: Análise de Sensibilidade Um por vez. Equações diferenciais ordinárias.

¹Graduando em Ciência da Computação, FISIOCOMP - Laboratório de Fisiologia Computacional e Computação de Alto Desempenho Universidade Federal de Juiz de Fora, matheusfajardo@ice.ufjf.br
²Professor Orientador, FISIOCOMP - Laboratório de Fisiologia Computacional e Computação de Alto Desempenho Universidade Federal de Juiz de Fora, marcelo.lobosco@ice.ufjf.br
1 INTRODUCTION

Sensitivity analysis is a tool used for understanding a mathematical model, more specifically, the influence that each parameter will have on the results of a model. In this way, it is possible to identify its most sensitive parameters, that is, those whose changes have a more significant impact on the results and, for this reason, should have their reference values assigned more carefully.

Although the basic principle that guides sensitivity analysis is extremely simple, changing the values used in a model and observing its new behavior, some complications may arise in this process, especially when the OAT (one-at-a-time) sensitivity analysis approach is employed. In particular, a large number of experiments may be necessary, and therefore the analysis of a large number of results is necessary to identify the most sensitive parameter(s).

The objective of this work is to assist this process: a tool was developed to carry out local sensitivity analysis automatically using the OAT approach. The tool allows the user to describe a mathematical model composed of ODE (Ordinary Differential Equations) and choose, for each parameter, its reference value, lower and upper limits and ranges of variation. At the end of its execution, the tool identifies the most sensitive parameters of the model. This paper describes the computational implementation of this tool, as well as a case study of its use in a model that describes the immune response against the yellow fever vaccine.

The rest of this paper is organized as follows. Section 2 presents a brief review of sensitivity analysis and section 3 presents the implementation of the tool. The case study is presented in section 4. Finally, conclusions and plans for future work are presented in the last section.

2 SENSITIVITY ANALYSIS

According to the literature (SALTELLI et al., 2008), the sensitivity analysis allows to determine how the uncertainty in an output of a mathematical model can be attributed to different uncertainties in its inputs. There are several reasons to perform the sensitivity analysis of a model, among which we can highlight: a) to better understand the relationships between the parameters of a model and its results; b) to identify the parameters of a model that are the main causes of variations in the results and that, therefore, should be adjusted more carefully; c) to identify errors in the model, resulting from unexpected relationships between the input parameters and the responses produced by them; and d) to simplify the model, excluding parameters that have little effect on its results.
There are several approaches to carry out the sensitivity analysis of a mathematical model. One of the simplest and most commonly used is one-at-a-time (OAT), which analyzes the impact of changing a single parameter on the outputs produced by the model, keeping the other parameters fixed. This process is repeated several times for the same parameter, until all values of interest are evaluated. In the next step, this procedure is repeated for each parameter of the model. The sensitivity can then be measured by monitoring changes in the results, since any change that is observed could only be caused by the change in the parameter value. Despite its simplicity, this approach does not fully explore the entire input space, since it does not take into account the simultaneous variation of two or more parameters, thus it is not possible to identify correlations between them, which is a limitation of this approach.

3 A COMPUTATIONAL TOOL TO AUTOMATE THE SENSITIVITY ANALYSIS

The tool proposed in the scope of this work aims to automate the sensitivity analysis of ODE systems, using the OAT approach.

The tool allows the user to define, at run time, the ODE system, either through a file or a graphical interface. Additionally, for each parameter and initial condition, it is possible to define its default value, the percentage of variation, and the upper and lower limits. For example, for a given parameter $A$, the following values can be defined: initial value equals to 50, the percentage of variation equals to 10%, lower limit equals to 10% and upper limit equals to 20%. Thus, the model would be evaluated using the values 45, 50, 55 and 60 for the parameter $A$. All parameters of the model can be studied in the same way.

At the end of the sensitivity analysis process, the tool informs the user the variations in the outputs observed by changing each parameter, allowing he/she to identify the most sensitive parameters.

3.1 Computational Implementation

The tool was developed using the Python programming language, which is available in distinct hardware and operating systems platforms. Python has also the advantage of having a vast number of libraries that offer tools to facilitate the development of graphical interfaces, the numerical resolution of EDOs, and the generation of graphs with the results obtained.

For the development of the graphical interface, the standard GUI library for Python, Tkinter, was used. For the resolution of the ODE system, the library chosen was scipy (ODEINT, 2014), which has a package called `integrate` with some functions for differential
and integral computations. In particular, this package has a function called `odeint` that implements the numerical resolution of ODE systems. The function `odeint` chooses the numerical method to solve the ODE system automatically, depending on its characteristics. For example, the BDF (Backward Differentiation Formulae) or Adams(LEVEQUE, 2007) methods can be used, depending whether stiff equations are identified or not, respectively.

Initially, the user must define the number of parameters and equations that his/her model has, as shown in Figure 1. In a second screen, shown in Figure 2, the user types the default values of each parameter, the set of equations, the percentage of variation that will be used in each step and the lower(%) and upper(%) limits. There is also the possibility of entering this data through a file, which facilitates the process for more complex ODE systems. Initial conditions are also required for all variables.

Figure 1: The first screen asks the user for the number of parameters and equations.

Figure 2: In the second screen, the values of the parameters, the percentage of variation, their lower and upper limits, as well as the set of equations, are requested.

After the user has entered all data, he/she can choose to solve the system of equations (by clicking on the “Calculate” button), or to perform the sensitivity analysis (by clicking on the “Analysis” button). In the case the user chooses to solve the system of equations, the lower and upper limits do not need to be informed, and the tool will solve it using the function `odeint`.

In the case the user chooses the sensitivity analysis, the system will automatically vary, one at a time, each input parameter, between its minimum and maximum values, with intervals given by the rate of change. The results obtained for each equation over time will be stored in a vector. This process is done repeatedly for each variation of each parameter. The values
obtained in each of the variations are, finally, compared to identify the largest variations found, which are then printed.

4 CASE STUDY

This section presents the results of the sensitivity analysis of a mathematical model of the immune response against the yellow fever vaccine (BONIN et al., 2018). The mathematical model is composed by ten ODE used to represent the main populations related to the immune response against the vaccine, such as the virus used in the vaccine, antigen presenting cells, B and T cells, and antibodies (BONIN et al., 2018).

Since the focus of this work is the sensitivity analysis tool, the ODE system will be not described in details. Some of its equations will be reproduced here with the purpose of describing how to represent the equations in the tool. The complete model, as well as the detailed description of each of its parameters, can be found in the reference paper (BONIN et al., 2018).

The Equation 1 represents the vaccine virus (V):

$$\frac{d}{dt} V = \pi_v V - \frac{c_{v1} V}{c_{v2} + V} - k_{v1} V A - k_{v2} V T_E.$$  \hspace{1cm} (1)

The virus cannot reproduce itself, so it infects a cell and uses it as a factory for new virus. The first term represents represents the multiplication of the virus in the body, with a production rate $\pi_v$; the second term, $\frac{c_{v1} V}{c_{v2} + V}$, denotes the non-specific viral clearing done by the innate innate system. The third term, $k_{v1} V A$, denotes the viral cleaning due to the opsonization of the virus by antibodies ($A$). The last term, $k_{v2} V T_E$, denotes specific viral clearing due to the induction of apoptosis of infected cells by effector T cells, represented by $T_E$.

The equation 2 represents antibodies ($A$):

$$\frac{d}{dt} A = \pi_{AS} P_S + \pi_{AL} P_L - \delta_A A$$  \hspace{1cm} (2)

The first and second terms represent, respectively, the production of antibodies by short-lived ($P_S$) and long-lived ($P_L$) plasma cells. The third term denotes the natural decay of these molecules in the body.

In order to represent the ODE system in the tool, each model variable is associated with a position in a vector that represents the system of equations. This vector will be referred to as $f$. For example, in the equation 1, $V$ will be associated with $f[0]$, $A$ will be associated with
and \( T_E \) with \( f[3] \). In the same way, each model constant is associated with a position in a constant vector, which will be referred to as \( y \). For example, \( \pi_v \) will be associated with \( y[0] \), \( c_{v1} \) a \( y[1] \), \( c_{v2} \) a \( y[2] \), and so on. Thus, the ODE system that describes the immune response against the yellow fever vaccine (BONIN et al., 2018) will be represented as follows:

\[
\begin{align*}
    f[0] &= y[0] \cdot f[0] - (y[1] \cdot f[0]) / (y[2] + f[0]) - y[3] \cdot f[0] \cdot f[7] - y[4] \cdot f[0] \cdot f[3] \\
    f[1] &= y[5] \cdot (y[28] - f[1]) - y[6] \cdot f[1] \cdot (y[7] + \text{np.tanh}(f[0] - y[8])) \\
    f[2] &= y[6] \cdot f[1] \cdot (y[7] + \text{np.tanh}(f[0] - y[8])) - y[9] \cdot f[2] \\
    f[3] &= y[11] \cdot f[2] \cdot f[9] + y[12] \cdot f[2] \cdot f[3] - y[13] \cdot f[3] \\
    f[4] &= y[14] \cdot (y[31] - f[4]) + y[15] \cdot f[2] \cdot f[4] - y[16] \cdot f[2] \cdot f[4] - y[17] \cdot f[2] \cdot f[4] - y[18] \cdot f[2] \cdot f[4] \\
    f[5] &= y[16] \cdot f[2] \cdot f[4] - y[19] \cdot f[5] \\
    f[6] &= y[17] \cdot f[2] \cdot f[4] - y[20] \cdot f[6] + y[21] \cdot f[8] \\
    f[7] &= y[24] \cdot f[5] + y[25] \cdot f[6] - y[26] \cdot f[7] \\
    f[8] &= y[18] \cdot f[2] \cdot f[4] + y[22] \cdot f[8] \cdot (1 - f[8] / y[23]) - y[21] \cdot f[8] \\
    f[9] &= y[10] \cdot (y[36] - f[9]) - y[11] \cdot f[2] \cdot f[9]
\end{align*}
\]

After the representation of the model, its sensitivity analysis was performed. According to the work that proposed the model (BONIN et al., 2018), the three most sensitive parameters of the model are: \( \pi_v \), \( k_{v1} \), and \( k_{v2} \), respectively, the virus replication rate, the viral neutralization (opsonization) rate, and the viral clearing rate by the action of effector cells, the same parameters found by the tool.

Figure 3 shows the viremia and antibodies curves (the last one using two distinct simulation time, 50 and 4,000 days) generated by the tool, considering different values for the multiplication of the virus in the body (\( \pi_v \)).

Figure 4 shows the viremia and antibody curves (the last one using two distinct simulation time, 50 and 4,000 days) generated by the tool, considering different values for the viral clearance rate due to antibody signaling (\( k_{v1} \)).

Finally, figure 5 shows the viremia and antibody curves (the last one using two distinct simulation time, 50 and 4,000 days) generated by the tool, considering different values for the viral clearance rate due to the induction of apoptosis of cells infected by the YF virus (\( k_{v2} \)).
Figure 3: Results obtained by the automatic sensitivity analysis performed by the tool. In this figure, the values for the virus replication rate ($\pi_v$) were changed, while other values were kept constant: (a) viremia curve 15 days after vaccination; (b) antibody curve 50 days after vaccination and (c) antibody curve 4,000 days after vaccination.
Figure 4: Results obtained by the automatic sensitivity analysis performed by the tool. In this figure, the values for the viral clearance rate due to antibody signaling ($k_v$) were changed, while other values were kept constant: (a) viremia curve 15 days after vaccination; (b) antibody curve 50 days after vaccination and (c) antibody curve 4,000 days after vaccination.
Figure 5: Results obtained by the automatic sensitivity analysis performed by the tool. In this figure, the values for the viral clearance rate due to the induction of apoptosis of cells infected by the YF virus ($k_v$) were changed, while other values were kept constant: (a) viremia curve 15 days after vaccination; (b) antibody curve 50 days after vaccination and (c) antibody curve 4,000 days after vaccination.
5 CONCLUSION AND FUTURE WORK

This work presented a tool, developed in the Python programming language, to perform the sensitivity analysis in ODE systems using the one-at-a-time approach. As a case study, with the help of the tool, the sensitivity analysis of a mathematical-computational model that describes the response of the human immune system against the Yellow Fever vaccine was performed. The tool was able to automatically identify the most sensitive variables of the model.

As future work, the tool will be extended, adding other methods to perform the sensitivity analysis. Approaches such as the Sobol method, which can be used for a global sensitivity analysis, will be implemented in future versions of the tool.

ACKNOWLEDGMENTS

The authors would like to thank UFJF, CNPq and FAPEMIG for funding this research. Matheus is a BIC/UFJF fellow.

REFERENCES

BONIN, C. R. et al. A qualitatively validated mathematical-computational model of the immune response to the yellow fever vaccine. BMC immunology, BioMed Central, v. 19, n. 1, p. 15, 2018.

LEVEQUE, R. J. Finite difference methods for ordinary and partial differential equations - steady-state and time-dependent problems. [S.l.]: SIAM, 2007. I-XV, 1-341 p.

ODEINT. Odeint’s Homepage. 2014. Disponível em: <http://docs.scipy.org>.

SALTELLI, A. et al. Global Sensitivity Analysis: the Primer. [S.l.]: Wiley, New York, 2008.