

Implementation and Numerical Aspects of the Matlab Solver Designed For the Solution of Low Protein Model

Meraihi Mouna

ABSTRACT: In this paper, we discuss the implementation and numerical aspects of the Matlab solver designed for the solution of Low Protein Model (LPD) defined by:

$$\frac{dL}{dt} = \lambda_i - \mu DL, \quad \forall i = 1, 2, 3, 4. \tag{0.1}$$

$$\frac{dM}{dt} = \mu L - \delta M, \tag{0.2}$$

subject to an initial condition

$$L(0) = 0 \quad \text{and} \quad M(0) = 0. \tag{0.3}$$

The code is based on Euler’s Method and several technics of programmation.

Key Words: Low Protein Model, Euler’s method, analytical solution, mathematical model, code in Matlab.

Contents

1 Introduction	1
2 Low Protein Model	3
2.1 Solution of the linear equation (2.1)	3
3 Euler Methods	4
4 Computational Results	5
5 Program implements the euler’s method	6

1. Introduction

Global Infectious Diseases Epidemiology Network (GIDEON) is a web-based program for decision support and informatics in the fields of Infectious Diseases and Geographic Medicine. As of 2005, more than 300 generic infectious diseases occur haphazardly in time and space and are challenged by over 250 drugs and vaccines. 1,500 species of pathogenic:

- bacteria,
- viruses,
- parasites and fungi have been described. Printed media can no longer follow the dynamics of diseases, outbreaks and epidemics in "real time."

Bacteria (singular: bacterium) are a large domain of single-celled, prokaryote microorganisms. Typically a few micrometres in length, bacteria have a wide range of shapes, ranging from spheres to rods and spirals. Bacteria are ubiquitous in every habitat on Earth, growing in soil, acidic hot springs, radioactive waste, [10] water, and deep in the Earth’s crust, as well as in organic matter and the live bodies of plants and animals. There are typically 40 million bacterial cells in a gram of soil and a million bacterial cells in a millilitre of fresh water; in all, there are approximately five nonillion (5-1030) bacteria on Earth, [12] forming a biomass on Earth, which exceeds that of all plants and animals. [13] Bacte-

2010 *Mathematics Subject Classification*: 35B40, 35L70.

Submitted January 16, 2020. Published April 29, 2021

ria are vital in recycling nutrients, with many steps in nutrient cycles depending on these organisms, such as the fixation of nitrogen from the atmosphere and putrefaction. However, most bacteria have not been characterised, and only about half of the phyla of bacteria have species that can be grown in the laboratory. [14] The study of bacteria is known as bacteriology, a branch of microbiology.

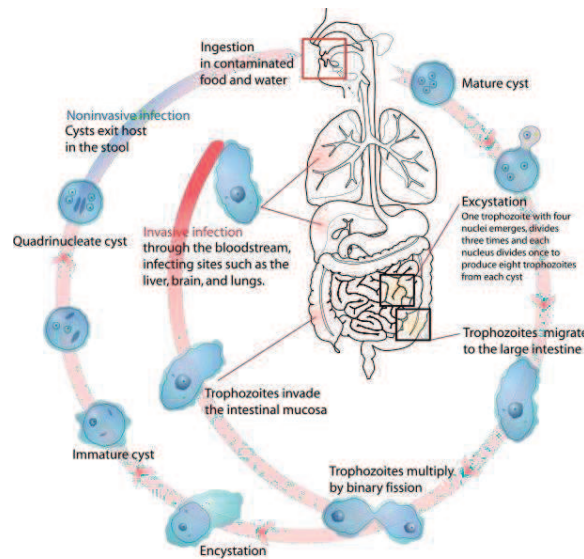


Figure 1: *Entamoeba histolytica* life cycle

- A virus is a small infectious agent that can replicate only inside the living cells of organisms. Most viruses are too small to be seen directly with a light microscope. Viruses infect all types of organisms, from animals and plants to bacteria and archaea. [10] Since Dmitri Ivanovsky's 1892 article describing a non-bacterial pathogen infecting tobacco plants, and the discovery of the tobacco mosaic virus by Martinus Beijerinck in 1898, [2] about 5,000 viruses have been described in detail, [13] although there are millions of different types. [4] Viruses are found in almost every ecosystem on Earth and are the most abundant type of biological entity. [12]–[14] The study of viruses is known as virology, a sub-speciality of microbiology.
- Parasitism is a type of symbiotic relationship between organisms of different species where one organism, the parasite, benefits at the expense of the other, the host. Traditionally parasite referred to organisms with lifestages that went beyond one host (e.g. *Taenia solium*), which are now called macroparasites (typically protozoa and helminths). Parasites can now also refer to microparasites, which are typically smaller, such as viruses and bacteria and can be directly transmitted between hosts of one species. Unlike predators, parasites are generally much smaller than their host, although both are special cases of consumer-resource interactions. [11] Parasites show a high degree of specialization for their mode of life, and reproduce at a faster rate than their hosts. Classic examples of parasitism include interactions between vertebrate hosts and diverse animals such as tapeworms, flukes, the *Plasmodium* species, and fleas. Parasitism is differentiated from the parasitoid relationship, though not sharply, by the fact that parasitoids generally kill or sterilize their hosts. Parasitoid occurs in about as many classes of organism as parasitism does.

The harm and benefit in parasitic interactions concern the biological fitness of the organisms involved. Parasites reduce host fitness in many ways, ranging from general or specialized pathology (such as parasitic castration), impairment of secondary sex characteristics, to the modification of host behavior. Parasites increase their fitness by exploiting hosts for resources necessary for the parasite's survival: (i.e. food, water, heat, habitat, and dispersal).

Although the concept of parasitism applies unambiguously to many cases in nature, it is best considered part of a continuum of types of interactions between species, rather than an exclusive category.

Particular interactions between species may satisfy some but not all parts of the definition. In many cases, it is difficult to demonstrate that the host is harmed. In others, there may be no apparent specialization on the part of the parasite, or the interaction between the organisms may be short-lived.

Our paper is structured as follows. In section one, we present a low protein model defined by two equations and their initial conditions. In section two, we describe the Euler's method applied to our model defined in section one. In the rest of this paper, we give Computational Results and Program implements the Euler's method.

2. Low Protein Model

Let us start by considering the parasite dynamics of the LPG. The parasites, harbored by a host population of constant size, are subdivided into two categories: larvae in the wall of the small intestine, and adult worms in the gut lumen. We model the dynamics of the mean number of larvae, L , per host by:

$$\frac{dL}{dt} = \lambda_i - \mu DL, \quad \forall i = 1, 2, 3, 4. \quad (2.1)$$

subject to an initial condition

$$L(0) = 0, \quad (2.2)$$

where $\lambda_i, i = 1, 2, 3, 4$ refer to the experimentally controlled infection rates, for example, 5, 10, 20 and 40 larvae per mouse per 2 weeks as in the experiments. Here $\frac{1}{D} = CL$ denotes the proportion of larvae developing into adult worms after a developmental time delay t_L here denoted by $\frac{1}{\mu}$. For the parasite *Heligmosoides polygyrus*, $t_L = \frac{1}{\mu} = 8$ days. Next, we estimate $C_L = 0.64$. We can now evaluate the net loss rate of the larval population per host as $\mu D = 0.195/day$ which implies an effective life span of a larval worm of $\frac{1}{\mu D} = 5.12/days$, and the natural larval mortality rate $\mu_0 = \mu(D - 1) = 0.07day$.

We model the dynamics of the mean adult worm burden, M , by

$$\frac{dM}{dt} = \mu L - \delta M, \quad (2.3)$$

subject to an initial condition

$$M(0) = 0. \quad (2.4)$$

where δ denotes the natural deathrate of the adult worms in the absence of competitive or immunological constraints. We estimate $\delta = 5.6 \times 10^{-3}/day$ from the experimental results of a single infection, which implies an adult worm life span of approximately 25 weeks.

2.1. Solution of the linear equation (2.1)

Let a first order linear non-homogeneous differential equation is

$$a \frac{df(x)}{dx} + bf(x) + c = 0 \quad (2.5)$$

subject to an initial condition

$$f(x_0) = f_0, \quad (2.6)$$

where a, b and c are constants.

Having a non-zero value for the constant c is what makes this equation non-homogeneous, and that adds a step to the process of solution. The path to a general solution involves finding a solution to the homogeneous equation (i.e., drop off the constant c), and then finding a particular solution to the non-

homogeneous equation (i.e., find any solution with the constant c left in the equation). The solution to the homogeneous equation is

$$f(x) = Ae^{-\frac{bx}{a}} \quad \text{homogeneous solution.} \quad (2.7)$$

By substitution you can verify that setting the function equal to the constant value $-\frac{c}{b}$ will satisfy the non-homogeneous equation.

It is the nature of differential equations that the sum of solutions is also a solution, so that a general solution can be approached by taking the sum of the two solutions above. The final requirement for the application of the solution to a physical problem is that the solution fits the physical boundary conditions of the problem.

The most common situation in physical problems is that the boundary conditions are the values of the function $f(x)$ and its derivatives when $x = 0$. Boundary conditions are often called "initial conditions". For the first order equation, we need to specify one boundary condition. For example:

$$\text{Find solution } f(x) = Ae^{-\frac{bx}{a}} - \frac{c}{b} \quad (2.8)$$

subject to

$$f(x) = B \quad \text{at } x = 0.$$

Substituting at $x = 0$ gives:

$$f(0) = B = A - \frac{c}{b} \quad (2.9)$$

and the solution $f(x)$ is given by :

$$f(x) = B.e^{-\frac{bx}{a}} + \frac{c}{b}.[e^{-\frac{bx}{a}} - 1] \quad (2.10)$$

In our case, $a = 1$, $b = \mu D$ and $c = -\lambda_i$, $i = 1, 2, 3, 4$. Then the solutions of the linear equations (2.1) with the initial conditions $L(0) = 0$, is simply

$$L(t) = \frac{\lambda_i}{\mu D}.(1 - e^{-\mu D t}) \quad \forall i = 1, 2, 3, 4. \quad (2.11)$$

Now, we use (2.11) in differential equation (2.3) we obtain

$$\frac{dM}{dt} = \frac{\lambda_i}{D}.(1 - e^{-\mu D t}) - \delta M \quad \forall i = 1, 2, 3, 4. \quad (2.12)$$

with the initial conditions $M(0) = 0$.

$$\frac{dM}{dt} = \frac{\lambda_i}{D}.(1 - e^{-\mu D t}) - \delta M \quad \forall i = 1, 2, 3, 4. \quad (2.13)$$

In the next section we present Euler Method to solve equation (2.13).

3. Euler Methods

In this section, we will consider a numerical method for a basic initial value problem, that is, for

$$\frac{dM}{dt} = f(t, M), \quad (3.1)$$

subject to an initial condition

$$M(0) = 0. \quad (3.2)$$

We will use a simplistic numerical method called Euler's method. Because of the simplicity of both the problem and the method, the related theory is relatively transparent and will be provided in detail. Though we will not do so, the theory developed in this chapter does extend to the more advanced methods to be introduced later, but only with increased complexity. With respect to (3.1), we assume that a unique

solution exists, but that analytical attempts to construct it have failed. The basic idea is as follows. By the definition of a derivative,

$$M'(t) = \lim_{h \rightarrow 0} \frac{M(t+h) - M(t)}{h}. \quad (3.3)$$

For small $h > 0$, then, (3.3) implies that a reasonable difference quotient approximation for $M'(t)$ is

$$M'(t) = \frac{M(t+h) - M(t)}{h}. \quad (3.4)$$

Substitution of (3.4) into (3.1) Yields the difference equation

$$\frac{M(t+h) - M(t)}{h} = f(t, M). \quad (3.5)$$

which approximates the differential equation (3.1). However, (3.5) can be rewritten as

$$M(t+h) = M(t) + hf(t, M). \quad (3.6)$$

which enables one to approximate $M(t+h)$ in terms of $M(t)$ and $f(t, M(t))$. Equation (3.6) is the cornerstone of Euler's method, which is described precisely as follows.

Since a computer cannot calculate indefinitely, let $t \geq 0$ be replaced by $0 \leq t \leq T_{max}$, in which T_{max} is a positive constant. The value of T_{max} is usually determined by the physics of the phenomenon under consideration. If the phenomenon occurs over a short period of time, then L can be chosen to be relatively small. If the phenomenon is long lasting, then T_{max} must be relatively large. In either case, T_{max} is a fixed, positive constant. The interval $0 \leq t \leq T_{max}$ is then divided into n equal parts, each of length h , by the points $t_i = ih, i = 0, 1, 2, \dots$. The value $h = \frac{(T_{max}-t_0)}{n}$ is called the grid size. The points t_i are called grid points. Let $M_i = M(t_i), i = 0, 1, 2, \dots$, so that initial condition (3.2) implies $M(0) = M_0 = 0$. Next, at each of the grid points t_0, t_1, t_2 and t_{n-1} , approximate the differential equation by (3.5) in the notation

$$\frac{M(i+1) - M_i}{h} = f(t_i, M_i). \quad i = 0, 1, 2, \dots \quad (3.7)$$

Or, in explicit recursive form

$$M(i+1) = M_i + hf(t_i, M_i) \quad i = 0, 1, 2, \dots \quad (3.8)$$

Then, beginning with

$$M(0) = M_0 = 0 \quad (3.9)$$

We set $i = 0$ in (3.8) and determine M_1 . Knowing M_1 , set $i = 1$ in (3.8) and determine M_2 . Knowing M_2 , set $i = 2$ in (3.8) and determine M_3 , and so forth, until, finally, M_n is generated. The resulting discrete function $M_0, M_1, M_2, \dots, M_n$ is called the numerical solution.

A generic algorithm for Euler's method is given as follows.

4. Computational Results

In this section Low's Protein equation was treated, and it is solved by using Euler's method. Input: end points $t_0, \mu, D, h, M_0, \delta$ and λ_i integer i , and n , tolerance TOL, maximum number of variation of i . In our case, we take $i = 1, 2, 3, 4$. Output: Approximation M_i and plots $M(t)$ for each $i = 1, 2, 3, 4$ for the first 12 weeks.

Euler's algorithm

Step 1. Set a counter $k = 1$.

Step 2. Set a time step h .

- Step 3. Set an initial time t .
- Step 4. Set initial value M .
- Step 5. Calculate $K0 = M$ and $K1 = hf(t, M)$.
- Step 6. Calculate M at $t + h$ by $M(t + h) = (K0 + K1)$.
- Step 7. Increase the counter from k to $k + 1$.
- Step 8. Set $M = M(t + h)$, $t = t + h$.
- Step 9. Repeat Steps 5{8}.
- Step 10. Continue until $k = 160$.

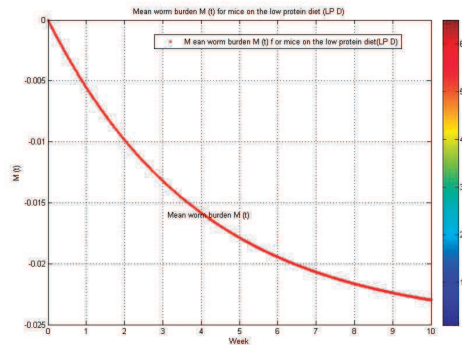


Figure 2: Entamoeba histolytica life cycle

5. Program implements the euler's method

In this part, We present a simple program implements the Euler's method. The files below can form the basis for the implementation of Euler's method using Matlab. They include `EULER.m`, which runs Euler's method; `f.m`, which defines the function $f(t, M(t))$; `ME.m`, which contains the exact analytical solution (computed independently). In order to solve a particular differential equation, you will need to define the function $f(t, M(t))$ in the file `f.m`, and also the exact solution in `ME.m`, if needed. The program computes the step size h from the initial and final t values a and b , and the number of steps N .

The file `EULER.m`

The solution is returned in an array M . You may wish to compute the exact solution using `ME.m` and plot this solution on the same graph as M , for instance by modifying the second-to-last line to read `plot(t,M,'-',t,ME(t),'-')`

The file `f.m`

The file `f.m` contains the function $f(t, M(t))$ for the general differential equation (2.5) above; the particular form of $f(t, M(t))$ corresponds to the equation

$$M' = \mu L - \delta M \quad (5.1)$$

To solve a different differential equation with `EULER.m` or another solver, you need only change this file.

```
function f=f(t,M(t))
f =  $\mu L - \delta M$ ;
```

The file ME.m

The file ME.m contains the exact solution $M(t)$ of equation (2.5), corresponding to the above function $f(t, M)$ defined in the file **f.m**. If you solve a different differential equation with **EULER.m** or one of the other numerical methods described below, and you wish to compare with an analytical expression for the exact solution, you should modify the file **ME.m** as well as **f.m**.

```
function ME = ME(t)
ME = 2 * ones(size(t)) + t - exp(-t);
```

You may choose whether or not to type these comments into your program, but if you include the comments in your file you must include the following program:

```
clear all ;
close all ;
format long
t0 = input('Enter the initial time-');
t1 = input('Enter the final time-');
M0 = input('Enter the initial condition-') ;
h = input('Enter the time step-');
np1 = t1/h;
tp1 = linspace(t0, t1, np1 + 1);
tolerance = 1.e - 04;
M = zeros(1, np1 + 1);
M(1) = M0;
t = t0;
mu = input('Enter the parameter value of mu-');
D = input('Enter the parameter value of D-');
delta = input('Enter the parameter value of delta-');
lambda = input('Enter the parameter value of lambda-');
a = lambda/D;
for i = 1 : np1
M(i + 1) = h * (a * (1 - exp(-mu * D * h))) + (1 - h * delta) * M(i);
t = t + h;
end
plot(tp1, M, 'r - .', tp1, 'sole', 'b-', 'LineWidth' 2);
grid
legend('Mean worm burden M (t) for mice on the low protein diet (LP D)', 'The analytical solution')
xlabel('Week');
ylabel('M (t)');
title('Mean worm burden M (t) for mice on the low protein diet (LP D)')
```

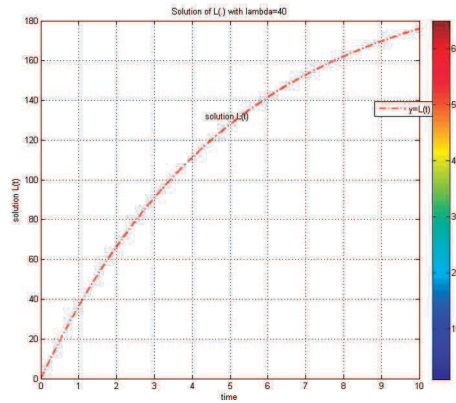


Figure 3: Entamoeba histolytica life cycle

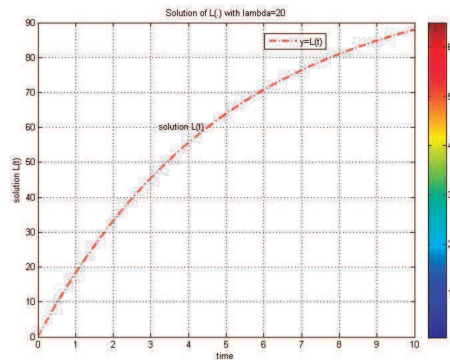


Figure 4: Entamoeba histolytica life cycle

This part contains program used to approximation M_i and plots $M(t)$ for each $i = 1, 2, 3, 4$ for the first 12 weeks. Mean worm burden $M(t)$ for mice on the low protein diet obtained from the analytical solution (2.11) of the model (2.1). The curves correspond to the different larvae infection rates λ_i for each $i = 1, 2, 3, 4$ larvae per mouse per 2 weeks. Parameter values $\mu = 0.125/\text{day}$, $D = 1.56$, $\delta = 5.6 \times 10 - 3/\text{day}$. These curves correspond to those superimposed on Figure 1, 2, 3 and 4.

Remark

The procedure is the same way as for the previous equation (2.1)-(2.2), and therefore we can write the program in matlab for the function $L(t)$ as follows:

```
clear all clear all;
close all;
clf;
clc;
format long;
t0 = input('Enter the initial time-');
t1= input('Enter the final time-');
L0 = input('Enter the initial condition-');
```



```

h = input('Enter the time step-');
np1 = t1/h;
tp1 = linspace(t0, t1, np1 + 1);
tolerance = 1.e - 04;
L = zeros(1, np1 + 1);
L(1) = L0;
t = t0;
mu = input('Enter the P arameter value of mu-');
D = input('Enter the P arameter value of D-');
lambda = input('Enter the P arameter value of lambda-');
a= (1 - mu * D * h);
for i = 1 : np1;
L(i + 1) = a * L(i) + lambda * h;
t= t + h;
end
plot(tp1, L,'r -.', 'LineWidth', 2);

```

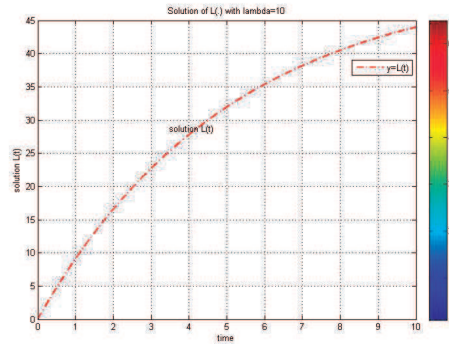


Figure 5: Entamoeba histolytica life cycle

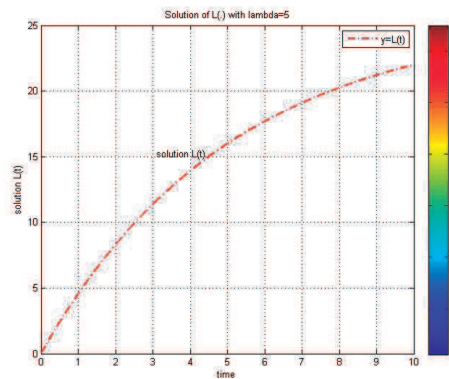


Figure 6: Entamoeba histolytica life cycle

This part too contains program used to approximation L_i and plots $M(t)$ for each $i = 1, 2, 3, 4$ for the first 12 weeks. The curves correspond to the different larvae infection rates λ_i for each $i = 1, 2, 3, 4$ larvae per mouse per 2 weeks. Parameter values: $\mu = 0.125/\text{day}$, $D = 1.56$ and $\lambda = 5, 10, 20 \text{ and } 40$. These curves

correspond to those superimposed on *Figure 5, 6, 7* and 8.

We show that figures 5, 6, 7 and 8 plots $L(t)$ for $i = 1, 2, 3, 4$ for the first 12 weeks using the above estimates for the parameter values. There is very good quantitative agreement. We estimate $\delta = 5.6 \times 10^{-03}/\text{day}$ from the experimental results of a single infection.

References

1. Lions, J. L., Exact controllability, stabilizability, and perturbations for distributed systems, *Siam Rev.* 30 (1988), 1-68.
2. Aikman, D. and Hewitt, G., An experimental investigation of the rate and form of dispersal in grasshoppers, *J. Appl. Ecol.* (1972), p.p. 807-817.
3. Beck, M.T. and Vàradi, Z.B., One, two and three-dimensional spatially periodic chemical reactions, *Nature.* (1972), p.p. 15-16.
4. Britton, N.F., *Reaction-Diffusion Equations and their Applications to Biology*, Academic Press, New York. (1986).
5. Getz, W., Biomass transformation webs provide a unified approach to consumer resource modelling, *Ecology Letters*. doi: 10.1111/j.1461-0248.2010.01566.x. (2011).
6. Hliddell, H. G. and Scott, R., *A Greek-English Lexicon*, on Perseus Digital Library, (2002).
7. Sibony, M. and Mardon, J.C.L., *Approximation et quations differentielles*, (1982).
8. Diekmann, O. and Heesterbeek, J. A. P., *Mathematical Epidemiology of Infectious Diseases: Model Building, Analysis and Interpretation*. John Wiley, New York. (2000).
9. Capasso, V. and Pavari-Fontana, S. L., A mathematical model for the 1973 cholera epidemic in the European Mediterranean region, *Rev. Epidemic. et Santé Publ.* (1979), p.p. 121-132.
10. Rook, G.A.W., The hygiene hypothesis and the increasing prevalence of chronic inflammatory disorders, *Transactions of the Royal Society of Tropical Medicine and Hygiene*. Vol. 101 (11): 1072-4, (2007).
11. Falconer, K.J., *Fractal Geometry, Mathematical foundations and Applications*. John Wiley and sons, Ithaca, New York. (1990).
12. Fife, P.C. and McLeod, J.B., The approach of solutions of nonlinear diffusion equations to travelling wave solutions, *Archiv. Rat. Mech. Anal.* (1977), Vol. 65, p.p. 335-361.
13. Lively, C. M. and Mark F. Dybdahl., Parasite adaptation to locally common host genotypes, *Nature*. Vol. 405. 8 June (2000).
14. Jean L. M., *Analyse numeérique avec MATLAB*, (2007).
15. Frerichs, R.R. and Prawda, J., A computer simulation model for the control of rabies in an urban area of Colombia, *Management Science.* (1975), Vol. 22, p.p. 411-421.

*Department of Mathematics,
Faculty of Exact Sciences,
Frères Mentouri Constantine University,
B.P. 325 Route Ain El Bey City: Constantine,
State Zip code: 25017, Country: Algeria.
Laboratory: Applied Mathematics and Modeling Laboratory
E-mail address: waelkhodja@hotmail.fr*