Analytical and Numerical Simulation of Epidemic Models using Maple and Sage

Verónica Orjuela Contreras
Engineering Physics, Universidad EAFIT, Medellín, Antioquia, Colombia
Microengineering Group, Logic and Computation Group, vorjuela@eafit.edu.co

Abstract **-** *The simulation of some epidemic models was made using computational software such as Maple and Sage, and the results were analytical or numerical solutions. The performance of each program was compared and these results could be applied to model the current spread of disease in the world. The obtained graphics show the behavior of the models in some hypothetic cases.*

Keywords: Epidemic Models, Computational Software: Maple and Sage, SI model, SIR model, SEIR model, Vector Borne Model.

1 Introduction

Currently, the epidemics reach high level of spread of disease as a result of the globalization, and sometimes the behavior is unknown and the control can be very challenging. Due to this, epidemic models have been developed to predict the outbreak and the proliferation of the infection.

Nevertheless, these models are not lineal and the solution cannot be obtained analytically, that is why only numerical solutions are gotten through the use of computational software such as Maple and Sage. In previous studies, the nonlinear models have been considered only from the point of view of computation of the so called basic reproductive number, using computer algebra[1,2,3,4,5].

The objective of this paper is to compare the performance of these two software: Maple[6] and Sage[7], modeling the SI model, SIR model, SEIR model and Vector Borne model, and contrast the results and the benefits of each one.

2 Problem

2.1 SI Model

In the SI Model the equations are:

$$\frac{d}{dt}x(t) = -\beta\, x(t)\, y(t) \quad (1) \qquad \frac{d}{dt}y(t) = \beta\, x(t)\, y(t) \quad (2)$$

Where, β is the contact or infection rate of the disease, and x(t) and y(t) are susceptible and infected individuals respectively.

2.2 SIR Model

In the SIR Model the equations are:

$$\frac{d}{dt}x(t) = -\beta\, x(t)\, y(t) \quad (3)$$

$$\frac{d}{dt}y(t) = \beta\, x(t)\, y(t) - g\, y(t) \quad (4)$$

$$\frac{d}{dt}z(t) = g\, y(t) \quad (5)$$

Where, β is the contact or infection rate of the disease, *g* represents the mean recovery rate; x(t), y(t) and z(t) are susceptible, infected and recovered individuals respectively.

2.3 SEIR Model

In the SEIR Model the equations are:

$$\frac{d}{dt}x(t) = -\beta\, x(t)\, z(t) \quad (6)$$

$$\frac{d}{dt}y(t) = \beta\, x(t)\, z(t) - \sigma\, y(t) \quad (7)$$

$$\frac{d}{dt}z(t) = \sigma\, y(t) - g\, z(t) \quad (8)$$

$$\frac{d}{dt}w(t) = g\, z(t) \quad (9)$$

Where, β is the contact or infection rate of the disease, σ is the transition rate of the exposed individuals to the infected one, *g* represents the mean recovery rate; *x(t), y(t), z(t)* and *w(t)* are susceptible, exposed, infected and recovered individuals respectively.

2.4 Vector Borne Model

In Vector Borne Model the equations are:

$$\frac{d}{dt}x_h(t) = -\beta_{m,h}\, x_h(t)\, y_m(t) \quad (10)$$

$$\frac{d}{dt}y_h(t) = \beta_{m,h}\, x_h(t)\, y_m(t) - g_h y_h(t) \quad (11)$$

$$\frac{d}{dt}z_h(t) = g_h y_h(t) \quad (12)$$

$$\frac{d}{dt}x_m(t) = -\beta_{h,m}\, x_m(t)\, y_h(t) \quad (13)$$

$$\frac{d}{dt}y_m(t) = \beta_{h,m}\, x_m(t)\, y_h(t) - g_m y_m(t) \quad (14)$$

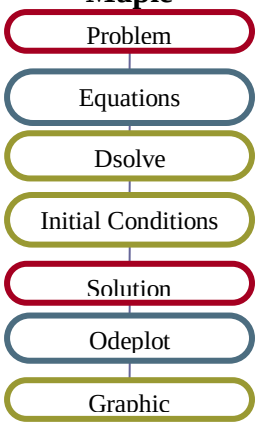
$$\frac{d}{dt}z_m(t) = g_m y_m(t) \quad (15)$$

Where, $\beta_{m,h}$ is the contact or infection rate of the disease due to mosquitoes over humans, $\beta_{h,m}$ is the contact or infection rate of the disease in humans when mosquitoes spread the infection, g_h represents the mean recovery rate in humans, and g_m the mean recovery rate in mosquitoes; $x_h(t)$, $y_h(t)$ and $z_h(t)$ are susceptible, infected and recovered individuals respectively, and $x_m(t)$, $y_m(t)$ and $z_m(t)$ are susceptible, infected and removed mosquitoes respectively.

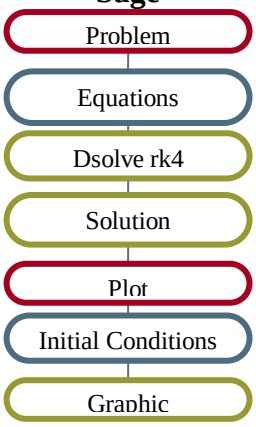
3 Method

In this section, the algorithms using Maple are presented in the first column, and in the second one, using Sage.

3.1 Algorithms using Maple



3.2 Algorithms using Sage



4 Results

With the previous algorithms it was possible to obtain the following results using Maple and Sage:

4.1 Results using Maple

4.1.1 SI Model

For the case of a closed population of constant size N=n+*a*, where n is the initial number of susceptible and *a* is the initial number of infected individuals, we have the following analytic solution

$$x(t) = \frac{e^{-t\beta(a+n)}\, n(a+n)}{a + e^{-t\beta(a+n)}\, n} \quad (16)$$

$$y(t) = \frac{(a+n)\, a}{a + e^{-t\beta(a+n)}\, n} \quad (17)$$

For the case of a closed population with variable size $N(t)=N+\mu\cdot t$, we have the analytic solution, n equation 18.

$$y(t) = -\frac{\sqrt{-2\mu\beta}\; a e^{\frac{1}{2}\beta t(2N+\mu t)}}{\beta\sqrt{\pi}\, e^{-\frac{1}{2}\frac{\beta N^2}{\mu}} \operatorname{erf}\left(\frac{\beta(N+\mu t)}{\sqrt{-2\mu\beta}}\right) a - \sqrt{-2\mu\beta} - \beta\sqrt{\pi}\, e^{-\frac{1}{2}\frac{\beta N^2}{\mu}} \operatorname{erf}\left(\frac{\beta N}{\sqrt{-2\mu\beta}}\right) a} \quad (18)$$

where a is the initial number of infected individuals, and erf represents the errors function defined as

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt. \quad (19)$$

A numerical illustration, for the case with closed population with constant size is made possible using Maple with the following commands

```
sys1:=diff(x(t),t)=-beta*(x(t))*(y(t)),diff(y(t),t)=beta*(x(t)
*y(t)):
 fcns1:={x(t),y(t)}:
p1:=dsolve({sys1,x(0)=45400,y(0)=2100},fcns1,type=num
eric,method=classical):
odeplot(p1,[[t,x(t)],[t,y(t)]],0..300);
```

And the result is showed in the illustration 1

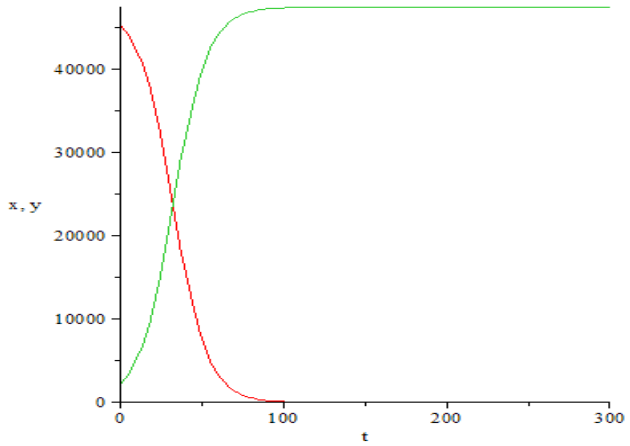


Illustration 1. Result SI Model, closed population with constant size

A numerical illustration for the case of population with variable size is as follows in equation 20:

$$y(t) = -\frac{\sqrt{-2\mu\beta}\; a e^{\frac{1}{2}\beta t(2N+\mu t)}}{\beta\sqrt{\pi}\, e^{-\frac{1}{2}\frac{\beta N^2}{\mu}} \operatorname{erf}\left(\frac{\beta(N+\mu t)}{\sqrt{-2\mu\beta}}\right) a - \sqrt{-2\mu\beta} - \beta\sqrt{\pi}\, e^{-\frac{1}{2}\frac{\beta N^2}{\mu}} \operatorname{erf}\left(\frac{\beta N}{\sqrt{-2\mu\beta}}\right) a} \quad (20)$$

The corresponding graphic is showed in illustration 2

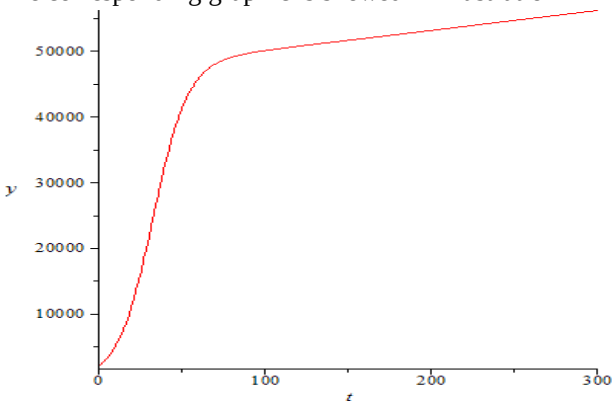


Illustration 2. Result SI Model population with variable size

4.1.2 SIR Model

A numerical illustration, for the case with closed population with constant size is illustrated as it follows

sys:=diff(x(t),t)=-beta(x(t))*(y(t)),*
diff(y(t),t)=beta(x(t))*(y(t))-g*(y(t)),diff(z(t),t)= g*(y(t)):*
fcns:={x(t),y(t),z(t)}:
p:=dsolve({sys,x(0)=45400,y(0)=2100,z=2500},fcns,type=numeric,method=classical):
odeplot(p,[t,x(t)],0..300,numpoints=25);

And the results are showed in the illustration 3 and 4

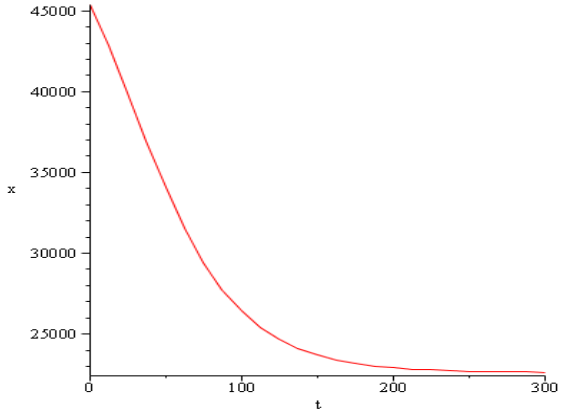


Illustration 3. Result SIR Model, susceptible individuals

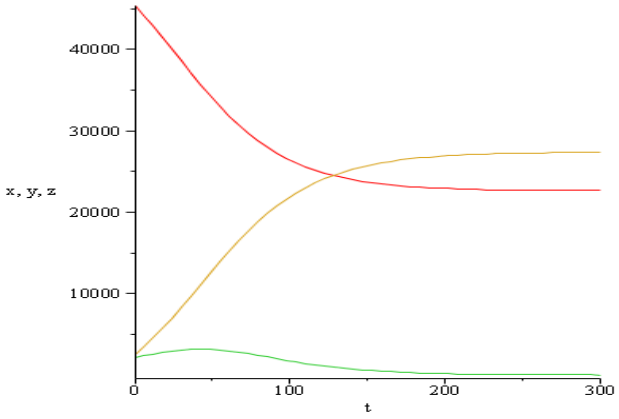


Illustration 4. Result SIR Model

4.1.3 SEIR Model

A numerical illustration, for the case with closed population with constant size is made using the next command

sys2:=diff(x(t),t)=-beta(x(t))*(z(t)),diff(y(t),t=beta*x(t))-sigma*(y(t),diff(z(t),t)=sigma*(y(t))-g*(z(t)),diff(w(t),t)=g*(z(t)):*
fcns2:={x(t),y(t),z(t),w(t)}:
p2:=dsolve({sys,x(0)=45400,y(0)=2100,z=2500,w(0)=10000},fcns2,type=numeric,method=classical):
odeplot(p2,[t,x(t)],[t,y(t)],[t,z(t)],[t,w(t)]0..300);

And the results are presented in illustration 5

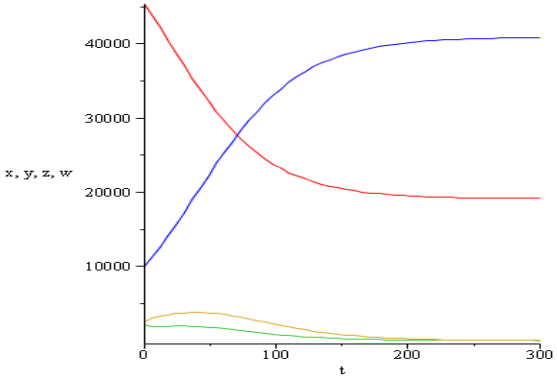


Illustration 5. Result SEIR Model

4.1.4 Vector Borne Model

A numerical result, for the case with closed population with constant size of individuals and mosquitoes is made through the following command

$$sys := diff(x[h](t),t) = -\beta[m,h]\cdot x[h](t)\cdot y[m](t),\ diff(y[h](t),t) = \beta[m,h]\cdot x[h](t)\cdot y[m](t) - g[h]\cdot y[h](t),\ diff(z[h](t),t) = g[h]\cdot y[h](t),\ diff(x[m](t),t) = -\beta[h,m]\cdot x[m](t)\cdot y[h](t),\ diff(y[m](t),t) = \beta[h,m]\cdot x[m](t)\cdot y[h](t) - g[m]\cdot y[m](t),\ diff(z[m](t),t) = g[m]\cdot y[m](t):$$

$$fcns := \{x[h](t), y[h](t), z[h](t), x[m](t), y[m](t), z[m](t)\}:$$

$$p := dsolve(\{sys, x[h](0) = 1000, y[h](0) = 330, z[h](0) = 170, x[m](0) = 500, y[m](0) = 112, z[m](0) = 50\}, fcns, type = numeric, method = classical):$$

$$odeplot(p, [[t, x[h](t)], [t, y[h](t)], [t, z[h](t)], [t, x[m](t)], [t, y[m](t)], [t, z[m](t)]], 0..150, color = [red, blue, green, black, yellow, brown]);$$

And the results can be observed in illustration 6

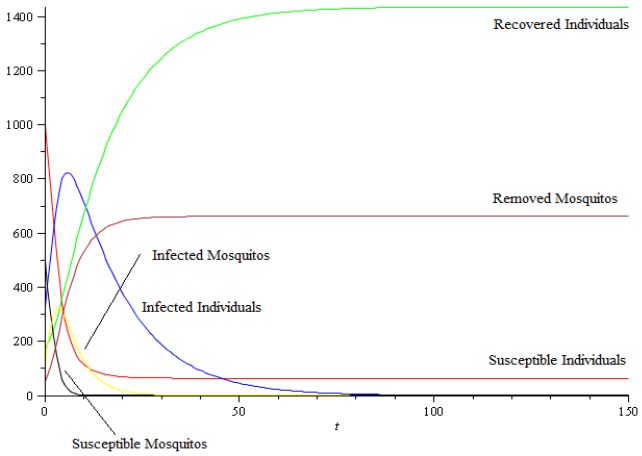


Illustration 6. Result Vector Borne Model

4.2 Results using Sage

4.2.1 SI Model

In Sage the algorithm for a numerical illustration, in the case with closed population with constant size, is the following one

```
def si_ode(S_init, I_init,     # initial values
           endtime,            # time to model
           beta = 2/10^6,   # model parameters
           colors=['red','blue'], thickness=3 # plotting controls
           ):
    var('t S I')
    system = [-beta*S*I, beta*S*I]

    # P will be a set of 3-tuples of the form (t,S,I)
    P = desolve_system_rk4(system,[S,I,], ivar=t,
                           ics=[0, S_init, I_init],
                           step=1, end_points=endtime)

    plotS = plot(line([(time,s) for time,s,i in P],
                 linestyle = '--', rgbcolor=colors[0], thickness=thickness))
    plotI = plot(line([(time,i) for time,s,i in P],
                 linestyle = '-.', rgbcolor=colors[1], thickness=thickness))
    return plotS+plotI
```

And the initial conditions provides the illustration 7
Si_ode(45400, 2100, 300)

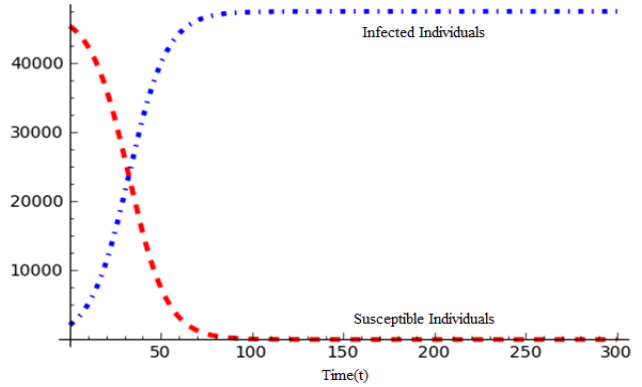


Illustration 7. Result SI Model

4.2.2 SIR Model

The algorithm for a numerical illustration, in the case with closed population with constant size is

```
def sir_ode(S_init, I_init, R_init,     # initial values
            endtime,                      # time to model
            beta = 2/10^6, gamma=1/14, # model parameters
            colors=['black','red','blue'], thickness=3 # plotting controls
            ):
    var('t S I R')
    system = [-beta*S*I, beta*S*I- gamma * I, gamma *I]

    # P will be a set of 4-tuples of the form (t,S,I,R)
    P = desolve_system_rk4(system,[S,I,R], ivar=t,
                           ics=[0, S_init, I_init, R_init],
                           step=1, end_points=endtime)

    plotS = plot(line([(time,s) for time,s,i,r in P],
                  linestyle = '-', rgbcolor=colors[0]), thickness=thickness)
    plotI = plot(line([(time,i) for time,s,i,r in P],
                  linestyle = '--', rgbcolor=colors[1], thickness=thickness))
    plotR = plot(line([(time,r) for time,s,i,r in P],
                  linestyle = '-.', rgbcolor=colors[2], thickness=thickness))
    return plotS+plotI+plotR
```

The initial conditions are changed and the results are showed in illustration 8 and 9.
sir_ode(45400, 2100, 2500, 300)

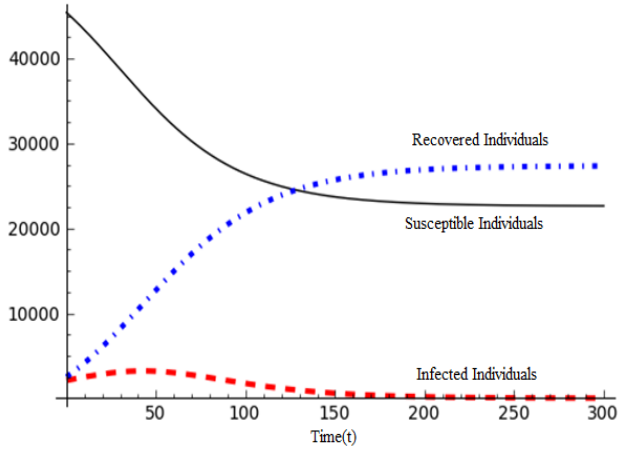


Illustration 8. Result SIR Model

```
sir ode(45400,2100,2500,300,beta=1/50000)
```

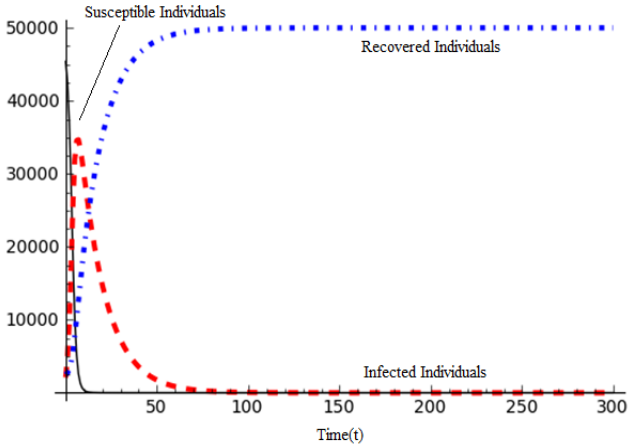


Illustration 9. Result SIR Model with different beta

```
plot1 = sir_ode(45400, 2100, 2500, 150)
plot2 = sir_ode(45400, 2100, 2500, 150,
gamma=1/10^2, colors=['gray','purple','orange'])
show(plot1 + plot2)
```

The result is showed in the illustration 10.

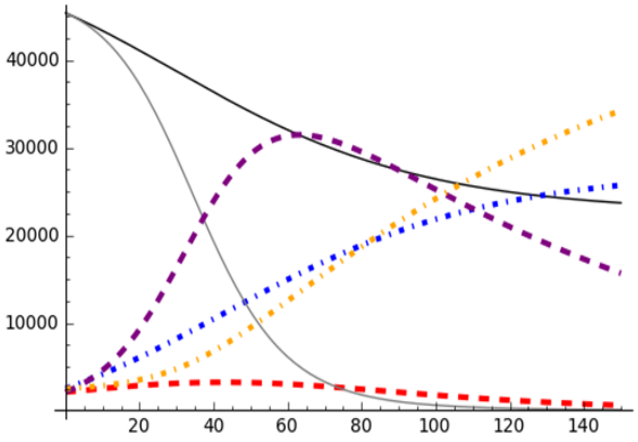


Illustration 10. Result SIR Model for both cases

4.2.3 SEIR Model

For the case of a closed population with constant size, it's possible to obtain a solution through the following algorithm

```
def seir_ode(S_init, E_init, I_init, R_init,          # initial values
             endtime,                                 # time to model
             beta = 2/10^6, gamma=1/14, sigma=1/10, # model parameters
             colors=['black','red','blue','green'], thickness=3 # plotting controls
             ):
    var('t S E I R')
    system = [-beta*S*I, beta*S*I- sigma*E, sigma*E- gamma*I, gamma*I]

    # P will be a set of 5-tuples of the form (t,S,E,I,R)
    P = desolve_system_rk4(system,[S,E,I,R], ivar=t,
                           ics=[0, S_init, E_init, I_init, R_init],
                           step=1, end_points=endtime)

    plotS = plot(line([(time,s) for time,s,e,i,r in P],
                 linestyle = '-', rgbcolor=colors[0]), thickness=thickness)
    plotE = plot(line([(time,e) for time,s,e,i,r in P],
                 linestyle = '--', rgbcolor=colors[1]), thickness=thickness)
    plotI = plot(line([(time,i) for time,s,e,i,r in P],
                 linestyle = '--', rgbcolor=colors[2], thickness=thickness))
    plotR = plot(line([(time,r) for time,s,e,i,r in P],
                 linestyle = '-.', rgbcolor=colors[3], thickness=thickness))
    return plotS+plotE+plotI+plotR
```

The initial conditions give the results showed in the illustration 11

```
seir_ode(45400, 9000, 3100, 2500, 300)
```

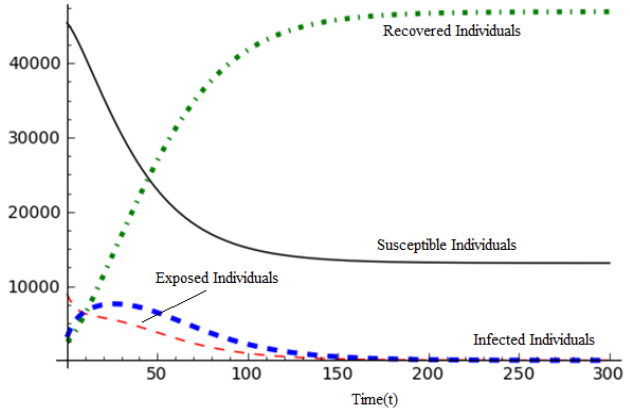


Illustration 11. Result Vector Borne Model

4.2.4 VECTOR BORNE MODEL

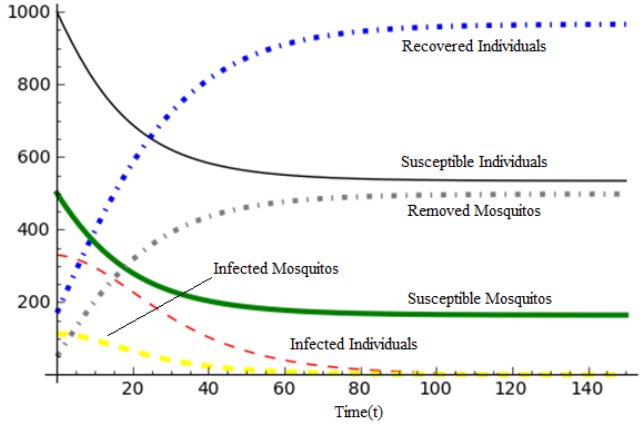
A numerical result, for the case with closed population with constant size of individuals and mosquitoes is made through the following algorithm

```
def vectbor_ode(Sh_init, Ih_init, Rh_init, Sm_init, Im_init, Rm_init,  # initial values
             endtime,                                  # time to model
             betamh=2/10^4, gammah=1/14, betahm=1/10^4, gammam=1/7,  # model parameters
             colors=['black','red','blue','green','yellow','gray'], thickness=3 # plotting controls
             ):
    var('t Sh Ih Rh Sm Im Rm')
    system = [-betamh*Sh*Im, betamh*Sh*Im- gammah*Ih,gammah*Ih, -betahm*Sm*Ih, betahm*Sm*Ih- gammam*Im,
gammam*Im]

    # P will be a set of 7-tuples of the form (t,Sh,Ih,Rh,Sm,Im,Rm)
    P = desolve_system_rk4(system,[Sh,Ih,Rh,Sm,Im,Rm], ivar=t,
                           ics=[0, Sh_init, Ih_init, Rh_init, Sm_init, Im_init, Rm_init],
                           step=1, end_points=endtime)

    plotSh = plot(line([(time,sh) for time,sh,ih,rh,sm,im,rm in P],
                 linestyle = '-', rgbcolor=colors[0]), thickness=thickness)
    plotIh = plot(line([(time,ih) for time,sh,ih,rh,sm,im,rm in P],
                 linestyle = '--', rgbcolor=colors[1]), thickness=thickness)
    plotRh = plot(line([(time,rh) for time,sh,ih,rh,sm,im,rm in P],
                 linestyle = '-.', rgbcolor=colors[2], thickness=thickness))
    plotSm = plot(line([(time,sm) for time,sh,ih,rh,sm,im,rm in P],
                 linestyle = '-', rgbcolor=colors[3], thickness=thickness))
    plotIm = plot(line([(time,im) for time,sh,ih,rh,sm,im,rm in P],
                 linestyle = '--', rgbcolor=colors[4], thickness=thickness))
    plotRm = plot(line([(time,rm) for time,sh,ih,rh,sm,im,rm in P],
                 linestyle = '-.', rgbcolor=colors[5], thickness=thickness))
    return plotSh+plotIh+plotRh+plotSm+plotIm+plotRm
```

The initial conditions produce the following graph



5 Conclusions

In this paper, several epidemic models were considered and the solutions were obtained through the application of two different software: Maple and Sage. The mathematical calculus and the speed, besides the computational language, were better in Maple. However the graphics of Sage gave a better view of the behavior of the models, even though Sage is very strict about its syntax. On the other side, Maple has a lot of commands to improve the graphics, therefore with a basic understanding, its performance is way better than Sage.
In addition, Maple is software which needs a license for its use, and Sage is free software that is available to everyone who wants it, only with an internet connection. Even so, Sage still has a lot of problems with the server, and not always is working good.
The epidemic models studied in this work could be useful to model some of the epidemics worldwide, due to easier incensement of the illness at the present time, and we hope that this model can predict the conduct and prevent the proliferation of the epidemics.

References

[1] Clarita Saldarriaga Vargas, Mathematical Model for Dengue Epidemics with Differential Susceptibility and Asymptomatic Patients Using Computer Algebra, Lecture Notes in Computer Science, Vol. 5743, pags 284-298, Springer 2009.

[2] Davinson Castaño Cano, Computer Algebra and Mechanized Reasoning in Mathematical Epidemiology, Proceedings of the World Congress on Engineering and Computer Science 2009 Vol I. WCECS 2009, October 20-22, 2009, San Francisco, USA

[3]Jeyver André Morales Taborda, Epidemic Thresholds via Computer Algebra, Proceedings of the 2008 International Conference on Modeling, Simulation & Visualization Methods, MSV 2008, Las Vegas, Nevada, USA, July 14-17, 2008, CSREA Press, 2008.

[4] Doracelly Hincapié palacio et al, The epidemic threshold theorem with social and contact heterogeneity, Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2008, Belur V. Dasarathy, Editors, 69730A.

[5]Juan Ospina et al, Epidemic Thresholds in SIR and SIIR Models Applying an Algorithmic Method, Biosurveillance and Biosecurity, International Workshop, BioSecure 2008 Vol 5354, Springer, NC, USA.

[6]Math Software for Engineers, Educators & Students | Maplesoft, www.maplesoft.com

[7]Sage: Open Source Mathematics Software, www.sagemath.org