International Journal of Engineering & Technology, 7 (4.10) (2018) 773-779

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



MATLAB Programming of Nonlinear Equations of Ordinary Differential Equations and Partial Differential Equations

S.Balamuralitharan[1*]

[1*]*Department of Mathematics, Faculty of Engineering & Technology, SRM Institute of Science and Technology, Kattankulathur - 603 203, Tamil Nadu, INDIA.*
**Corresponding author E-mail: balamurali.maths@gmail.com*

Abstract

My idea of this paper is to discuss the MATLAB program for various mathematical modeling in ordinary differential equations (ODEs) and partial differential equations (PDEs). Idea of this paper is very useful to research scholars, faculty members and all other fields like engineering and biology. Also we get easily to find the numerical solutions from this program.

Keywords: *Matlab Program, nonlinear ODE and PDE*

1. Introduction

The MATLAB is the very powerful tool in mathematics and all other engineering, as well as medicine. Many of the nonlinear ODE and PDE analytical solutions is very difficult. So we find the numerical solution and then compare to the real life problems. Hence we will use the MATLAB codes for all nonlinear programs such as ODE and PDE type problems [1]. Furthermore this MATLAB code is very useful to research scholars and faculty members for various mathematics department, engineering and medicine also [2].

In our work, mostly I have discussed many of the papers of mathematical modeling and mathematical biological phenomena [3, 4]. In this regard, we used many models from biology like, HIV model, SIR and SIS models, predator prey models, Lotka Volterra models, second order nonlinear ODE and PDE etc [5, 6, 7]. Hence these all models MATLAB codes are very useful for research scholars and beginners of the research areas in all application oriented field of biology, Engineering and pure mathematics etc [8, 9, 10].

In the HIV model is the best example of the mathematical modeling of biological area [9, 13]. Then its MATLAB code is good for other nonlinear ODE. So we will use the same code for all these types of problems in the system of equations [11]. Not only the nonlinear ODE, we discussed the nonlinear PDE that it gives fantastic Matlab codes [12].

Finally, it is given for one beautiful example of the numerical simulation of predator prey model that MATLAB code is valuable for many mathematical modeling areas and uses it to all other simulation like delay also [13]. The PDE MATLAB codes are the mathematical modeling of all areas such as valid for numerical simulations and many real life problems [14, 15]. Hence I conclude that this paper is useful for all researchers and same it will be used in many fields [16]. Already I have published many papers in international journals [5-16].

2. Mathematical models of nonlinear equations of ODE and PDE

In this section, we gave some important mathematical models in current research area based equations of the biology, engineering and other fields. It is collected from different areas of research in ODE and PDE as below:

(i) HIV model

$$\frac{da}{dt} = s + ra\left(1 - \frac{a+b+c}{T_{\max}}\right) - \mu a - k_1 ae,$$

$$\frac{db}{dt} = k_1 ae - \mu b - k_2 b,$$

$$\frac{dc}{dt} = k_2 b - \beta b,$$

$$\frac{de}{dt} = N\beta b - k_1 ea - \alpha e.$$

(ii) Enzyme Kinetics

$$\frac{da}{dt} = Aa + Bb,$$

$$\frac{db}{dt} = Aa - (B+C)b,$$

$$\frac{dc}{dt} = Cb.$$

(iii) General SIR model

$$\frac{dS}{dt} = -rSI,$$

$$\frac{dI}{dt} = rSI - aI,$$

$$\frac{dR}{dt} = aI.$$

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

(iv) three-species predator–prey problem

$$\frac{da}{dt} = -a + ab,$$

$$\frac{db}{dt} = -b + 2bc - ab,$$

$$\frac{dc}{dt} = 2c - c^2 - bc.$$

(v) SIR model

$$\frac{dS}{dt} = -rSI + R(t-1),$$

$$\frac{dI}{dt} = rSI - aI,$$

$$\frac{dS}{dt} = aI - R(t-1).$$

(vi) Original Lotka–Volterra equations

$$\frac{dx}{dt} = rx - axy$$

$$\frac{dy}{dt} = -my + bxy$$

(vii) Second order nonlinear ODE

$$\frac{d^2a}{dx^2} = M^2ab$$

$$\frac{d^2b}{dx^2} = \frac{M^2}{rq}ab$$

(viii) **SIR new model**

$$\frac{ds}{dt} = (1-p)\pi - \beta si - \pi s$$

$$\frac{di}{dt} = \beta si - (\gamma + \pi)i$$

$$\frac{dr}{dt} = p\pi + \gamma i - \pi r$$

(ix) Second order nonlinear ODE

$$y'' + \lambda \exp(y) = 0,\ t \in [0,1]$$

(x) Second order nonlinear PDE 1

$$\frac{\partial^2 a}{\partial x^2} = Dab$$

$$\frac{\partial^2 b}{\partial x^2} = \gamma Cab - C(1-b)$$

(xi) Second order nonlinear PDE 2

$$\left(\frac{\partial^2 c}{\partial x^2} + \frac{2}{x}\frac{\partial c}{\partial x}\right) - \alpha^2 \frac{c}{1 + c + \frac{c^2}{\beta}} = 0$$

3. MATLAB program

This section we discussed MATLAB code for above mentioned equations. Each model obtained the current research. The following MATLAB programs are executed for as below:

(i). function xprime = file name (t, x)
s = 1;
r = 0.05;
max = 1600;
mu = 0.01;
b = 17;
a = 1;
m1 = 0.000034;
m2 = 0.005;
N = 10000;
xprime = [s-mu*x1+r*x1.*(1-(x1+x2+x3)/max)-m1*x4.*x1;
m1*x4.*x1-mu*x2-m2*x2;
m2*x2-b*x3;
N*b*x3-m1*x4.*x1-a*x4];
[t, x] = ode23('file name',[0 1000], [10000; 0; 1; 1]);
Plot (t, x)

(ii). function xprime = file name (t, x)
mm1 = 1;
m2 = 1;
m1 = 1;
xprime = [-m1*x1+m1*m2;
m1*x1-(mm1+m2)*x2; m2*x2];
a = 0.5;
b = 0.0001;
c = 0.0001;
[t1,x1] = ode23 ('file name',[0 10],[a; b; c]);
plot(t1,x1);
[t10, x10] = ode23 ('file name',[0 10],[a; b; c]);
Plot (t10, x10)

(iii). function xprime = file name (t, x);
r = 4;
a = 5;
x prime = [-r*x1.*x2;
r*x1.*x2-a*x2;
a*x2];
[t, x] = ode45 ('file name',[0 100], [0.7; .1; 0]);
Plot (t, x)

(iv). function x prime = file name (t, x);
x prime = [-x1+x1.*x2;
-x2+2*x2.*x3-x1.*x2;
2*x3-x3.*x3-x2.*x3];
[t, x] = ode23 ('file name',[0 100],[0; .4; 2]);
Plot (t, x)

(v). n = 1000;
delt =1/n;
tf = 4;
nf = (tf+1)*n+1;
for
I = 1 : n
x1 = 0;
x2 = 0;
x3 = 0;
end
x1(n+1) = 4;
x2(n+1) = 0.1;
x3(n+1) = 0;
for
i = n+1 : nf-1
y = delt*[-r*x1*x3+x1(i-n);
r*x2*x3-a*x3;
a*x3-x1(i-n)];
x2(i+1) = x2+y1;
x3(i+1) = x3+y2;
x1(i+1) = x1+y3;
end
t = -1:delt : tf;
Plot (t, x1, t , x2 ,t , x3)

(vi). function xprime = file name (t, x)
a = 1.04;
b = 1.25;
c = 1.45;
d = 1.23;
xprime = [a*x1-b*x1.*x2;
-c*x2+d*x1.*x2];
[t, x] = ode('file name',[0 100],[1 ; 1]);
plot(t, x)

(vii). function pdex
a1= 0;
x = linspace (0,10);
t = linspace (0,10000);
function [a, b, c] = pdex (x, t, v , dvdx)

```
a = [2; 2];
b = [2; 2] .* dudx;
a1 = 0.7;
b1 = 90;
c1 = 2;
f = -a1^2*v1*v2;
f1 = -(a1^2)*v1*v2/(b1*c1);
c  = [f; f1];
function v = pdex
v = [2; 2];
function [c1, c2, c3, c4] = pdex (xl, vl, xr, vr, t)
a1 = 0.6;
b1 = 67;
c1 = 2;
a2 = 0.4;
b2 = 0.2;
c2 = 100;
d2 = 0.04;
v = 0.0001;
c1 = [v1-1; -a2*(1-b2)*v2];
c2 = [0; 1];
c3 = [a1^2*(c2-1)*v1+d2*(v1-v0) v2)-1];
c4 = [1; 0];
```

(viii). function main = odex

```
x0 = [0 ; 0.1, 1];
tspan = [0 1000];
tic [t, x] = ode45 (tf, tspan, x0, options);
plot(t, x)
return
function [dxdt]= tf (t, x)
a = 10.45;
b = 10.50;
c = 10.70;
d = 10.019;
dxdt1 = (1-a)*b-c*x1*x2-b*x1;
dxdt2  = c*x1*x2-(b+d)*x2;
dxdt3  =a*b+d*x2-b*x3;
dxdt  = dxdt;
 return
```

(ix). function pdex

```
m = 0;
x = linspace (0, 10);
t = linspace (0, 10000);
plot (x, v)
function [d, e, f] = pdex (x, t, v, dvdx)
d  = 9;
e  = dvdx;
la = 4;
h =la*exp(v);
f = h;
function v = pdex
v = 12;
function [c1,c2,c3,c4] = pdex (xl, vl, xr, vr, t)
c1 = vl;
c2 = 0.004;
c3= vr-0.007;
c4 = 0.001;
```

(x). function pdex

```
a = 0;
x = linspace (0,1);
t = linspace (0,1000);
function [d, e, f] = pdex(x, t, v, dvdx)
d = [2; 2];
e = [2; 2] .* dvdx;
x = v1 * v2;
x1=v1*v3;
a = 0.0011;
b = 0.012;
h=0.24;
la =0.00019;
g  = (-la*x-x1);
g1 = (-b*x);
g2 = (-a*b*x)-h + h*v2;
f=[g1; g2];
function v = pdex
v  = [0; 100];
function[c1,c2,c3,c4] = pdex(xl, v1, xr, vr, t)
c1 = [v1-1; 0];
c2 = [0; 1];
c3 = [v1; v2-1];
c4 = [0,0];
```

(xi). function pdex

```
a = 4;
x = linspace (0,1);
t =  linspace (0,1000);
function [d, e, f] = pdex (x, t, v, dvdx)
d = 2;
e = dvdx;
q = 0.7;
b = 100;
y = -q^2*b*v1/(b+b*v1+v1^2);
f = y;
function v = pdex
v = 2;
function [c1, c2, c3, c4]=pdex (xl, vl, xr, vr, t)
c1 = 0.0018;
c2 = 24;
c3 = vr-21;
c4 = 0.00419;
```

4. Numerical Example

This section I have discussed MATLAB code for predator prey model and to find the numerical solution of the system. It gives the population species in Fig 1. & predator prey for Fig 2, error for Fig 3, Fig 10. Also the population of predator prey percentages is in Fig 4-9 & Fig 11-14.

```
File name
Parameters    = structure ();
Initial States = structure ();
Ts = 0;
nlgr = idnlgrey();
disp ('parameter estimation');
p true = [0; 1; 0; 1];
disp(' initial values');
x true = [0.2; 4];
z = id data(y, 0.1, 'File Name', 'predator -prey system');
Set ();
fprintf(‘1.4f   1.4', [p true]);
```

Similarly, we develop the remaining all diagrams by using the same way to use MATLAB codes in the same systems. Hence it is a code for all the populations and predictions in different percentages in this program.

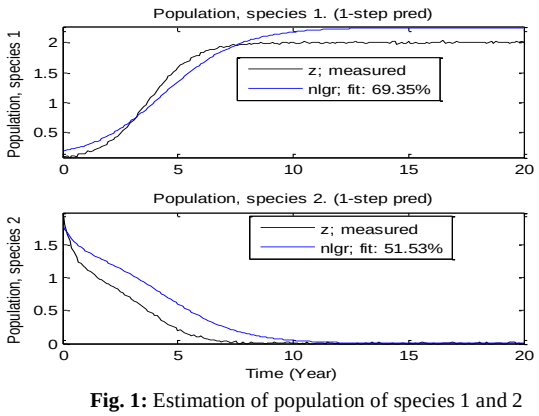


Fig. 1: Estimation of population of species 1 and 2

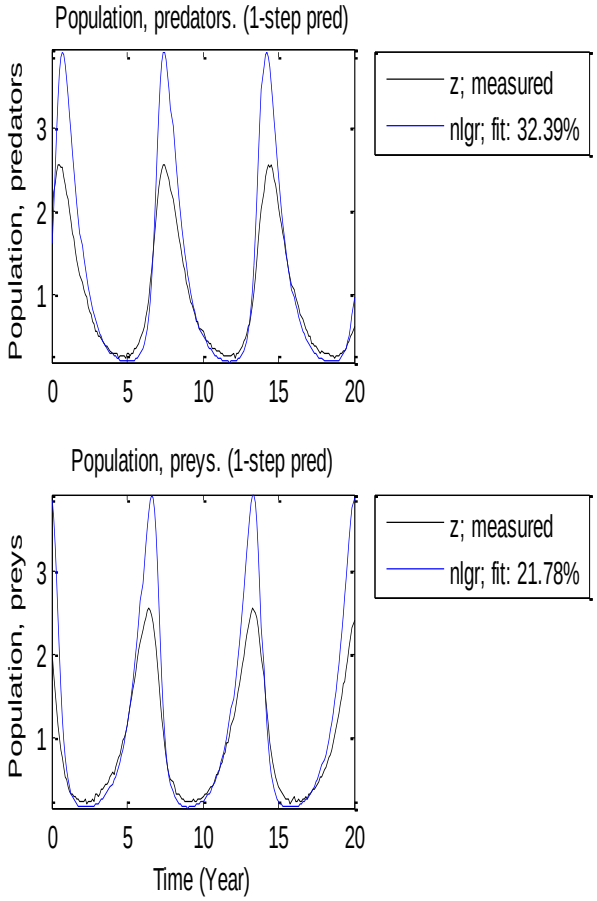


Fig. 2: Estimation of population of predators and preys in 21.78%

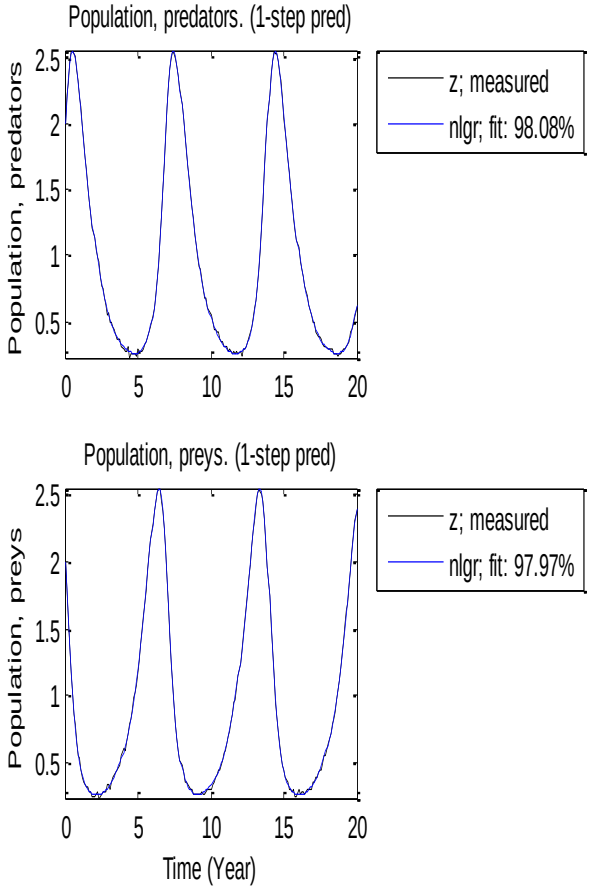


Fig. 4: Estimation of population predators and populations preys in 97.97%

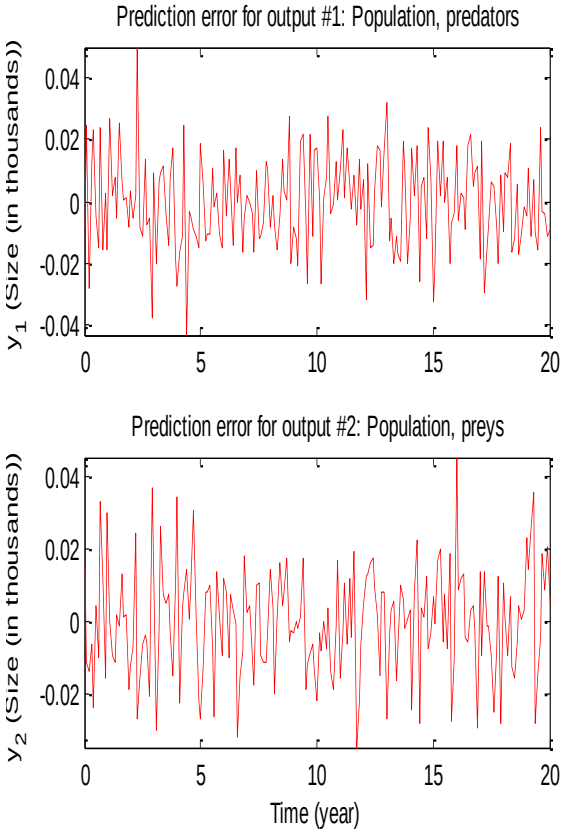


Fig. 3: Estimation of population of prediction error for output #1 and #2

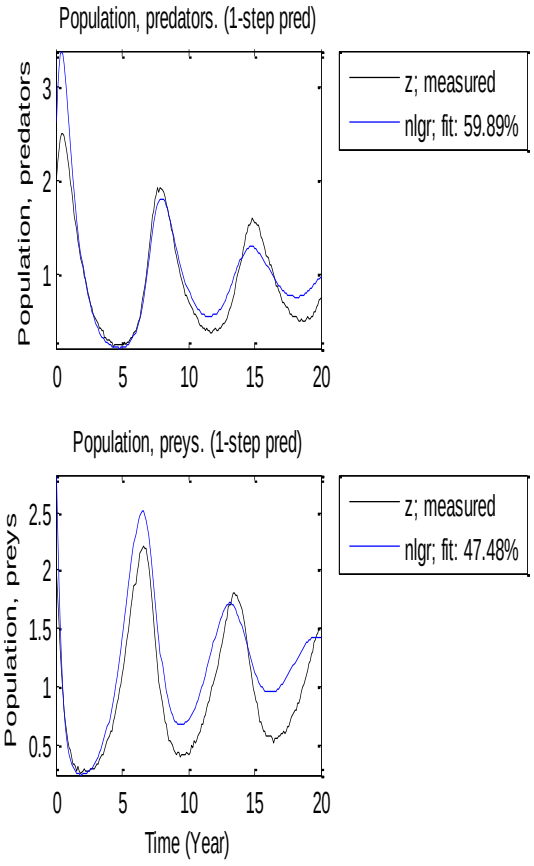


Fig. 5: Estimation of population of predator prey in 47.78%

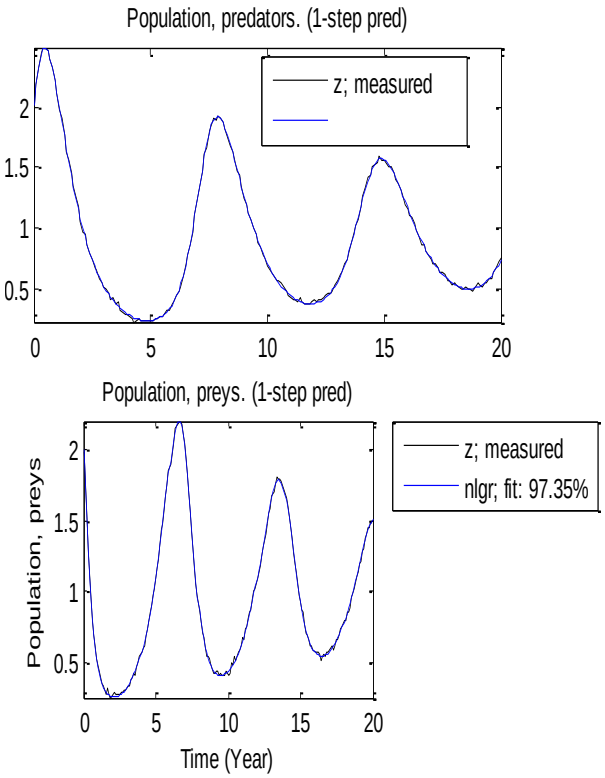


Fig. 6: Estimation of population of predator prey in 97.35%

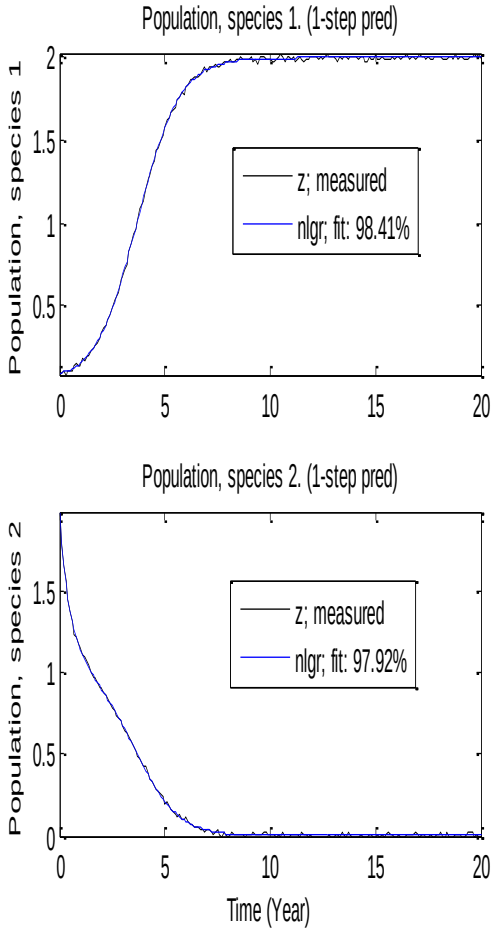


Fig. 7: Estimation of population of predator prey in 97.92%

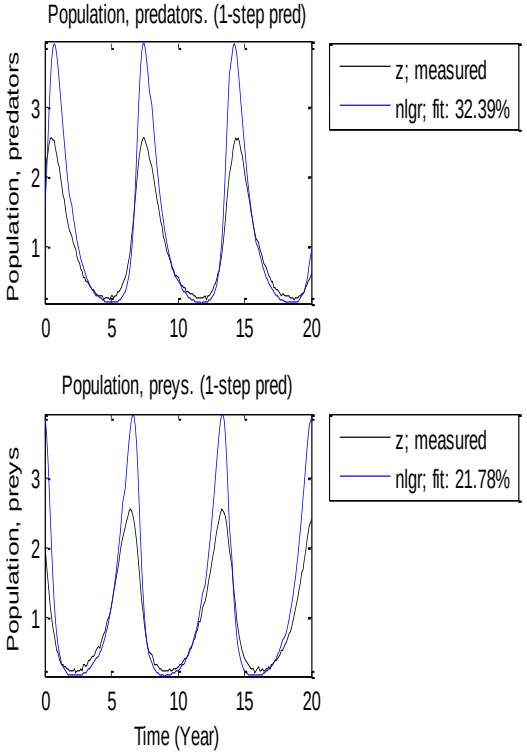


Fig. 8: Estimation of population of predator prey in 21.78%

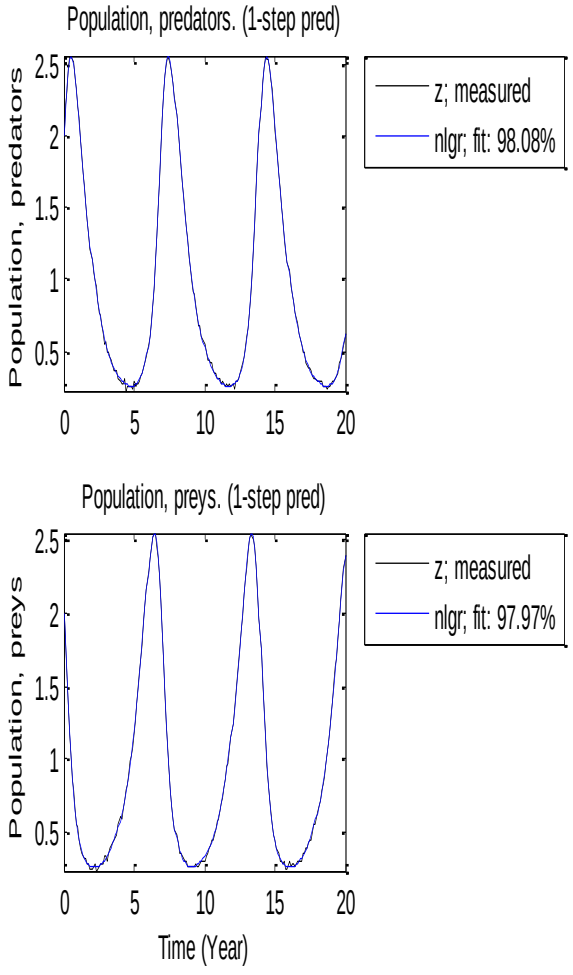


Fig. 9: Estimation of population of predator prey in 97.97%

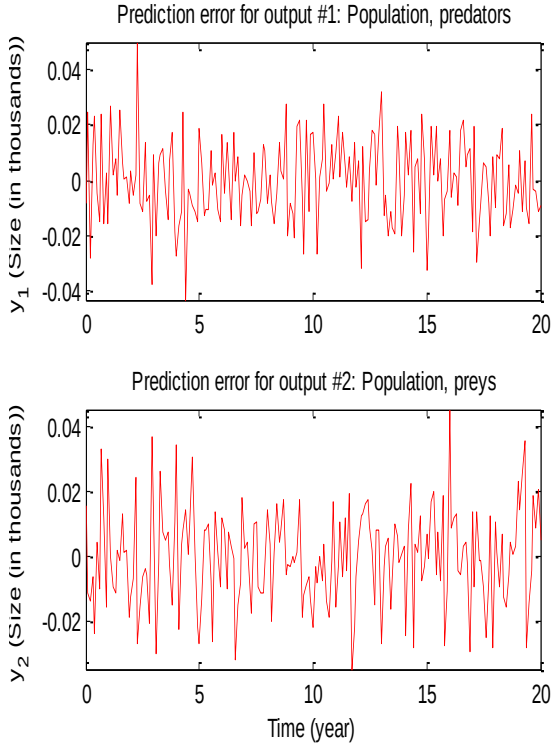


Fig. 10: Estimation of population of predator prey error

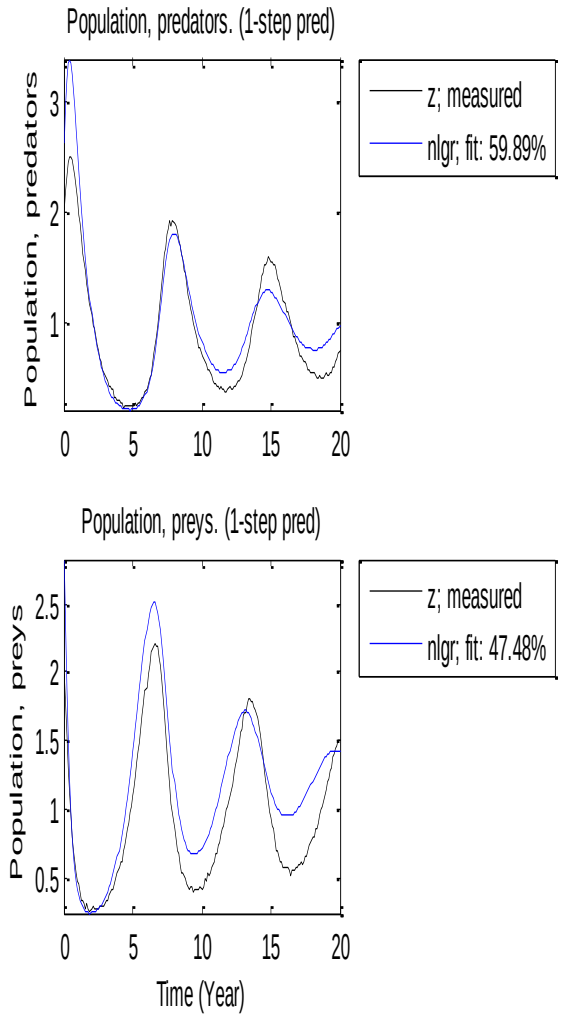


Fig. 11: Estimation of population of predator prey in 47.48%

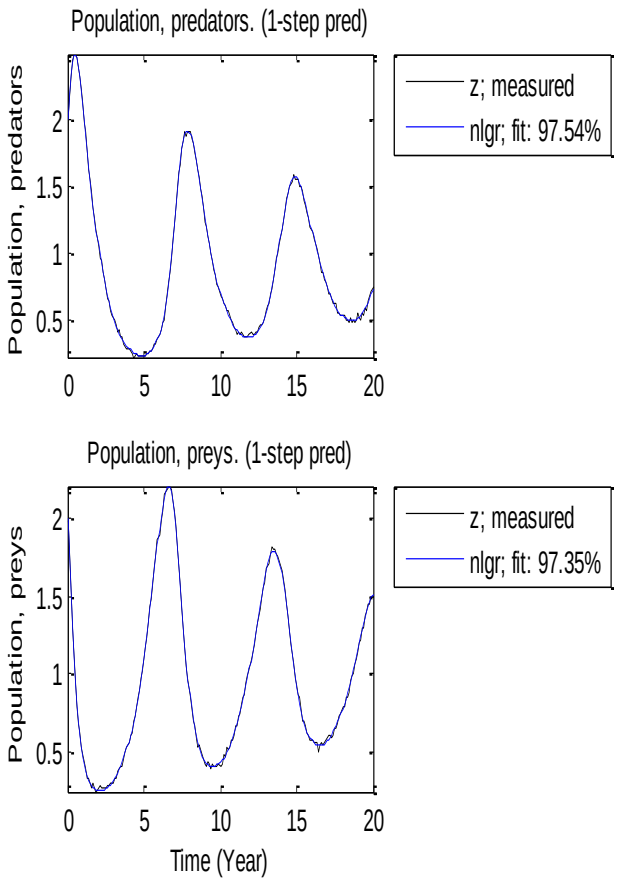


Fig. 12: Estimation of population of predator prey in 97.35%

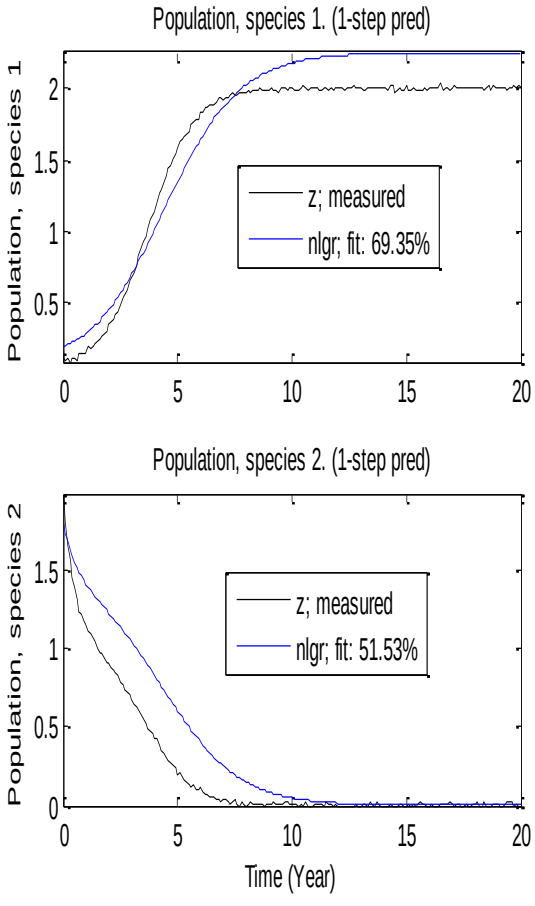


Fig. 13: Estimation of population of predator prey in 51.53%

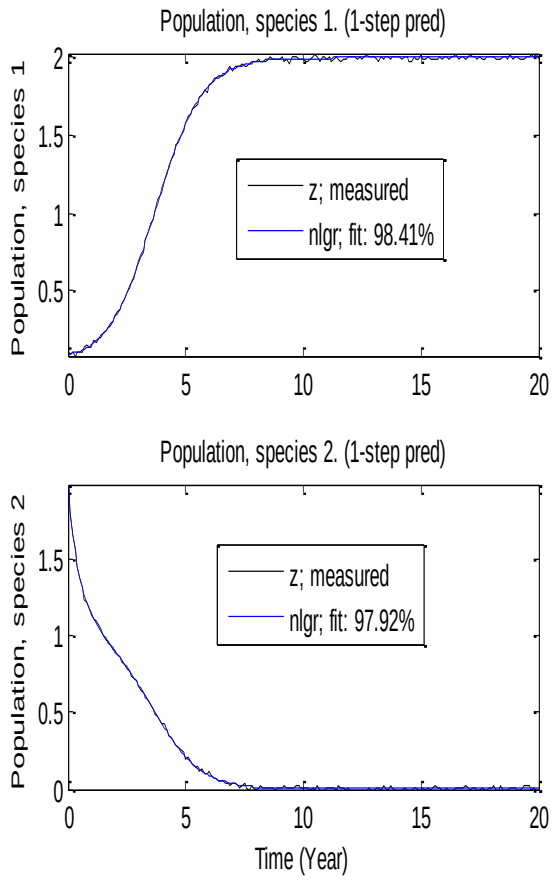


Fig. 14: Estimation of population of predator prey in 97.92%

5. Conclusion

My work of this paper is to create the MATLAB program for various mathematical modeling of ODEs and PDEs such as very important to pure and applied mathematics. This MATLAB code is very useful to find the numerical solutions and approximate solutions of nonlinearity problems.

References

[1] L. J. Gross (1994), Quantitative training for life-science students, *Biosci*., 44-2, 59.

[2] F. S. Kantor (1994),, Disarming Lyme disease, *Sci. Amer*., 271, 34–39.

[3] P. Raeburn (2009), Chaos and the catch of the day, *Sci. Amer*., 300-2, 76–78.

[4] A. S. Perelson (1993), D. E. Kirschner, and R. J. De Boer, The dynamics of HIV infection of CD4+ T cells, *Math. Biosci.*, 114, 81–125.

[5] Balamuralitharan S, D.Seethalakshmi (2016), Numerical Simulations for Parameter Estimation Model of SIR with Replacement Numbers, *Global Journal of Pure and Applied Mathematics*, 12, 118-123.

[6] Balamuralitharan S, S.Geethamalini (2016), Within Host Virus Models with Sensitive and Resistant Strain in EIAV, *Global Journal of Pure and Applied Mathematics*, 12, 329-334.

[7] S. Geethamalini and S. Balamuralitharan (2016), Homotopy Perturbation Method for Solving A Model for EIAV Infection, *International Journal of Control Theory and Applications*, 9(28), 439-446.

[8] S.Balamuralitharan and M.Radha (2017), Stability Analysis of Cholera - Carrier Dependent Infectious Disease, *International Journal of Pure and Applied Mathematics*, 113, 234 - 242.

[9] S.Balamuralitharan and V.Geetha (2017), Analytical Approach to solve the Model for HIV Infection of $Cd4^{+}$T Cells Using Ladm, *International Journal of Pure and Applied Mathematics*, 113, 243 - 251.

[10] S.Balamuralitharan and S.Geethamalini (2017), Parameter Estimation of Model for Eiav Infection Using Hpm, *International Journal of Pure and Applied Mathematics*, 113, 196 - 204.

[11] G. Arul Joseph and S. Balamuralitharan (2018), A Mathematical Modeling and Simulation of Non Linear Ordinary Differential Equations using Hpm, *ARPN Journal of Engineering and Applied Sciences*, 13, 2685- 2689.

[12] S. Balamuralitharan, M.Radha (2018), Bifurcation Analysis in SIR Epidemic Model with Treatment, *IOP Conf. Series: Journal of Physics: Conf. Series* 1000 doi :10.1088/1742-6596/1000/1/012169

[13] V.Geetha And S. Balamuralitharan (2018), Stability Analysis of Host Dynamics For HIV, *IOP Conf. Series: Journal of Physics: Conf. Series* 1000 doi :10.1088/1742-6596/1000/1/012022

[14] S.Balamuralitharan and S.Geethamalini (2018), Solutions Of The Epidemic of EIAV Infection by HPM, *IOP Conf. Series: Journal of Physics: Conf. Series* 1000 doi :10.1088/1742-6596/1000/1/012023

[15] G. Arul Joseph and S.Balamuralitharan (2018), A Nonlinear Differential Equation Model of Asthma Effect Of Environmental Pollution using LHAM, *IOP Conf. Series: Journal of Physics: Conf. Series* 1000 doi :10.1088/1742-6596/1000/1/012043

[16] Geethamalini S. and Balamuralitharan S (2018), Equine infectious anemia virus dynamics and stability analysis: The role of agar gel immune diffusion test and enzyme immune absorbent assay, *Research Journal of Biotechnology*, 13, 28- 33.