Solving nonlinear ODE (Van Der Pol) numerically using Matlab's ODE45

Nasser M. Abbasi

October 27, 2025 Compiled on October 27, 2025 at 8:23pm

Contents

1 Introduction

Report for course EGME 511 (Advanced Mechanical Vibration). California state university, Fullerton, CA. Van der Pol differential equation is given by

$$x''(t) - c\big(1 - x^2(t)\big)\, x'(t) + kx(t) = 0$$

In this analysis, we will consider the case only for positive c, k. The above equation will be solved numerically using Matlab's ODE45 for different initial conditions, and the phase portrait (velocity vs. displacement) is plotted to show the limit cycle for different initial conditions.

2 Matlab implementation

To use ODE45, one must first convert the above second order ODE to two ODE's, each of which is first order. Letting $x_1 = x, x_2 = x'(t)$ results in

$$\left.\begin{array}{c} x_1 = x \\ x_2 = x'(t) \end{array}\right\} \quad \left.\begin{array}{c} x_1' = x'(t) = x_2 \\ x_2' = x''(t) = c(1-x^2)\,x'(t) - kx(t) \end{array}\right\} \quad \left.\begin{array}{c} x_1' = x_2 \\ x_2' = c(1-x_1^2)\,x_2 - kx_1 \end{array}\right\}$$

The system of equations to be solved by ODE45 is the following

$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{pmatrix} x_2 \\ c(1-x_1^2)\,x_2 - kx_1 \end{pmatrix}$$

Subject to initial conditions $x_1(0) = x(0)$ and $x_2(0) = x'(0)$. In the Matlab implementation below, the values of c, k and the initial conditions are defined at the top of the code. This needs to be modified to change the initial conditions before running the program again.

3 Results and output

The program was run for a number of different initial conditions (different $x(0)$ and different $v(0)$). In all cases, $c = 1, k = 1$ was used. It is noticed that one limit cycle is reached in all cases. Both the $x(t)$ and the phase plot where shown. The following are output from 3 different runs made. The title on the plots show the initial conditions used.

3.1 Test case 1

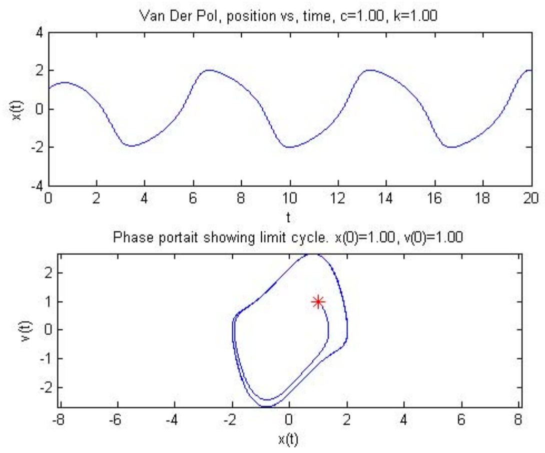

Figure 1: Test case 1

3.2 Test case 2

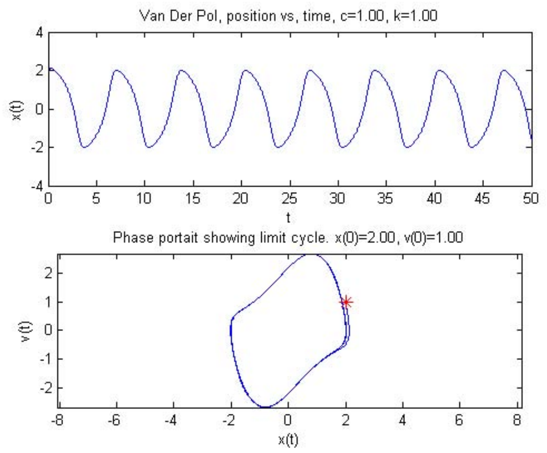

Figure 2: Test case 2

3.3 Test case 3

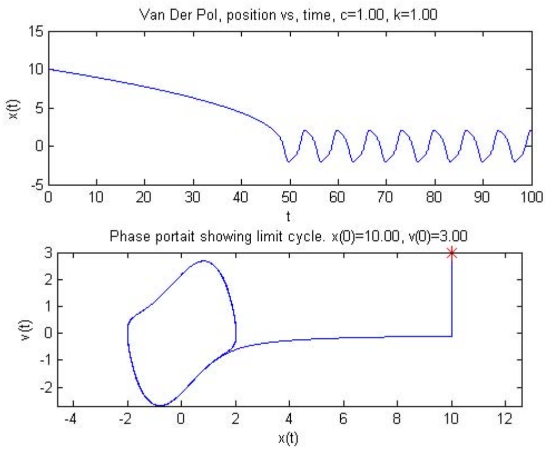

Figure 3: Test case 3

4 Conclusion

A non-linear second order ODE was solved numerically using Matlab's ode45. The solution to the Van Der Pol was found to contain a limit cycle in the phase portrait when starting from any initial conditions.

5 Source code

```
function nma_project_511()
% This function solves the Van Der Pol nonlinear ode
% numerically using Matlab's ode45.
%
% Final project for 511, CSUF, spring 2009. Instructor: Dr Oh, Sang June
%
% Written by : Nasser M. Abbasi
% Van Der Pol ode is  x''(t) - c (1-x^2(t)) x'(t) + k x(t) = 0

t  = 0:0.001:100;   % time scale
c  = 1;
k  = 1;
x0 = 10;
v0 = 3;

[t,x] = ode45( @rhs, t, [x0,v0]);

subplot(2,1,1);
plot(t,x(:,1));
xlabel('t'); ylabel('x(t)');
title(sprintf('Van Der Pol, position vs, time, c=%3.2f, k=%3.2f',c,k));

subplot(2,1,2);
plot(x(:,1),x(:,2));
xlabel('x(t)'); ylabel('v(t)');
hold on;
plot(x0,v0,'*r','MarkerSize',10);
title(sprintf('Phase portait showing limit cycle. x(0)=%3.2f, v(0)=%3.2f',x0,v0));
axis equal
hold off;

    function dxdt=rhs(t,x)
        dxdt_1 = x(2);
        dxdt_2 = c*(1-x(1)^2)*x(2)-k*x(1);

        dxdt = [dxdt_1 ; dxdt_2];
    end
end
```

The file to download is nma_project_511.m