

APPM 2460
SOLVING SYSTEMS OF EQUATIONS

1. INTRODUCTION

We know how to use `ode45` to solve a single ODE. A few weeks ago, we solved the problem

$$y'(t) - y(t) \cos(t + y(t)) = 0, \quad y(0) = 1$$

using this method, which is basically a pumped-up version of Euler's method.

Now we need to work on more complicated situations. For example, we might have a second-order ODE:

$$y''(t) + \sin(y(t)) = 0, \quad y(0) = 1, \quad y'(0) = 1$$

We might also have a *system* of first-order ODEs:

$$\begin{aligned} x'(t) &= y^2(t), & x(0) &= 1 \\ y'(t) &= -x(t) - \cos(y(t)), & y(0) &= 2 \end{aligned}$$

We can use `ode45` in either of these cases; we just need to learn how to adapt ourselves. First let's review the relationship between second order ODEs and systems.

2. SOLVING A SECOND-ORDER ODE WITH `ODE45`

2.1. Second-Order ODEs as Systems. Consider the so-called van der Pol equation,

$$(1) \quad x''(t) - x'(t)(1 - x^2(t)) + x(t) = 0, \quad x(0) = 2, \quad x'(0) = 0.$$

This is a second-order ODE, but we will now recast it as a *system* of first-order ODEs. We'll use $v(t)$ to denote $x'(t)$. Then, we can write (1) as

$$\begin{aligned} x'(t) &= v(t) \\ v'(t) &= v(t)(1 - x^2(t)) - x(t) \end{aligned}$$

where $x(0) = 2$ and $v(0) = 0$. We thus see that second-order ODEs (and indeed, any order ODE) can be rewritten as a system of first-order ODEs. This first-order form is what we need in order to use `ode45`. Note that we could treat the two elements of our system (x and v) as two elements of a vector, that is to say our vector y would be

$$(2) \quad y(t) = \begin{pmatrix} x(t) \\ v(t) \end{pmatrix}$$

So in this case, $x = y_1$ and $v = y_2$. Note that $y'(t) = (x'(t), v'(t))$. We could then write our ODE as a **vector-valued** function, i.e.

$$(3) \quad y'(t) = f(t, y) \quad \text{where} \quad f(t, y) = \begin{pmatrix} y_2 \\ y_2(1 - y_1^2) - y_1 \end{pmatrix}$$

To summarize: we have rewritten the ODE (1) in the form $y'(t) = f(t, y)$, but now y is a vector, with $y = (x, v)$. With our ODE in this form, we can put it into MATLAB.

2.2. Inputting the ODE into MATLAB. Now, we must figure out how to input this into Matlab. Our first task is to write a function $f(t,y)$ that takes in a time t (i.e. a number) and a vector y and outputs another vector $f(t,y)$. Well, we can do it as follows:

```
function yout = odefun(t,y)
yout = zeros(2,1); % initialize yout as a column vector of length 2
yout(1) = y(2);
yout(2) = y(2)*(1-y(1)^2) - y(1);
```

The above function generates an output $yout$ that is a 2×1 column vector. We can now call `ode45` like so:

```
y0 = [2 0]; % since y = (x,v) and x(0)=2, v(0)=0
tspan = [0 20]; % time runs from 0 to 20
[t,y] = ode45(@odefun,tspan,y0);
plot(t,y(:,1),t,y(:,2)); % plot the two columns of the result
```

We have plotted both $y_1(t) = x(t)$ and $y_2(t) = v(t)$ against time. Are there other ways to visualize solutions to systems of ODEs? Next we will look at another way of plotting the result.

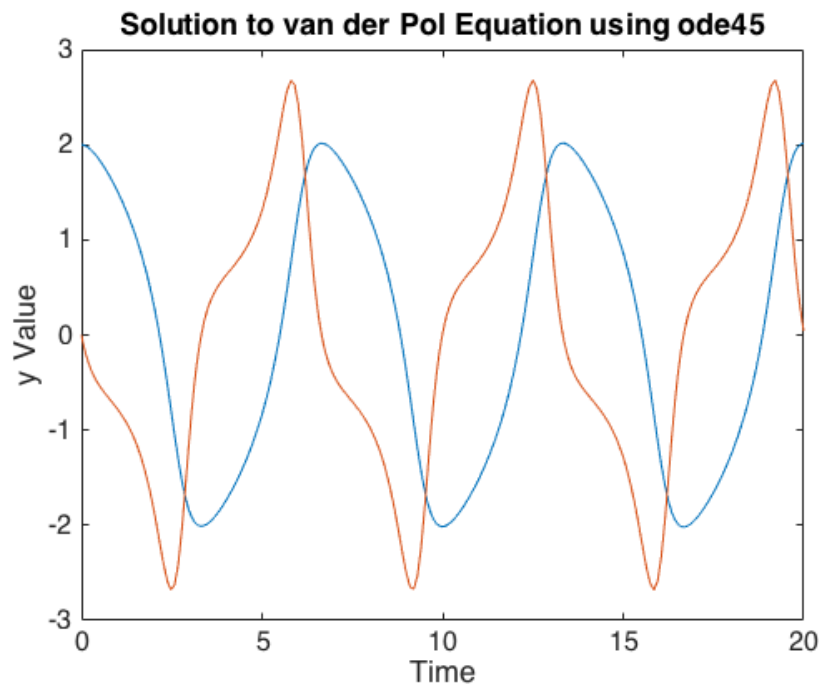


FIGURE 1. Solution to the van der Pol equation, using supplied code.

3. WORKING IN PHASE SPACE

Physicists often like to visualize systems in what they call **phase space**. Recall that we have a vector that depends on time, $y(t) = (x(t), v(t))$. To plot in phase space is simply to plot the parametric curve given by $(x(t), v(t))$. We can modify the above code to plot in phase space as follows:

```
y0 = [2 0]; % since y = (x,v) and x(0)=2, v(0)=0
tspan = [0 20]; % time runs from 0 to 20
[t,y] = ode45(@odefun,tspan,y0);
plot(y(:,1),y(:,2)); % plot the two columns of the result AGAINST ONE ANOTHER
```

The resulting figure is shown in Figure 2.

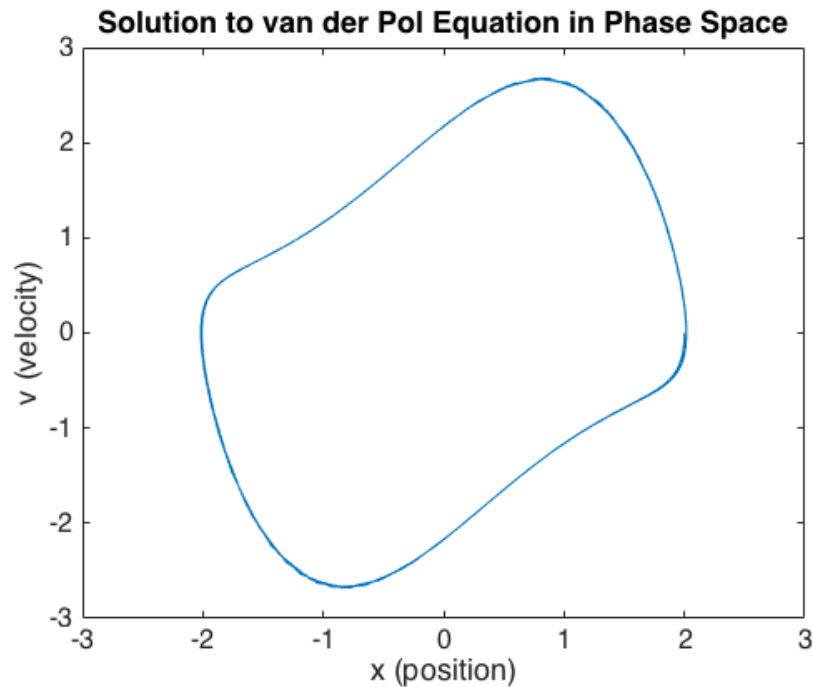


FIGURE 2. Phase space visualization of van der Pol solution.

4. HOMEWORK

Write a script that finds a numerical solution to the second-order IVP

$$x(t)'' - \mu(1 - x^2(t))x'(t) + x(t) - A \sin(\omega t) = 0$$

using parameters $\mu = 8.53$, $A = 1.2$, and $\omega = 2\pi/10$. Use the initial conditions $x(0) = 2$ and $x'(0) = 0$. (Note that the above equation reduces to the van der Pol equation we solved above if we set $\mu = 1$ and $A = 0$. Adding in nonzero A adds **forcing** to the system, i.e. makes it non-homogeneous.) Plot for times $t = 0$ to $t = 100$. Include both a plot of x and v against time, as well as a phase space plot of your solutions. Plots should be appropriately labelled and titled.

If you solve the system correctly, you will see some very strange behavior! In particular, there will be a pattern that almost repeats itself, but not exactly. This is an example of **chaotic** dynamics, which we will explore more thoroughly in a few weeks.