M.*Poochelvi*

Assistant professor in mathematics

Department of mathematics[SF]

C.P.A.COLLEGE

Differential Equations and a few helpful ways to solve them

Differential Equations

- In this class we will focus on solving ordinary differential equations that represent the physical processes we are interested in studying. With perhaps a few exceptions the most complicated differential equation we will look at will be second order, which means it will look something like

$$\frac{\partial}{\partial x}\left(D(x)\frac{\partial C(x)}{\partial x}\right)+v(x)\frac{\partial C(x)}{\partial x}=f(x,C)$$

- Sometimes you may be able to solve the equation by hand. Other times not. Here we will cover some useful methods to help you solve them as you need to.

SOS MATH

- Great website that has so much helpful stuff on it and is pretty accessible and easy to follow, unlike so many others.
- http://sosmath.com/diffeq/diffeq.html
- A lot of the differential equations we are interested are solved there. I have always found it useful as it helps me learn to solve these things by hand.

Mathematic

- As a student at Notre Dame you can download and have access to Mathematic, which is an incredibly powerful and amazing tool for solving all sorts of mathematical problems. If it cannot solve the differential equation we are looking at then it is unlikely you can either.
- The specific command for solving a differential equation is
- Solve

Example

- Solve the following differential equation in Mathematica assuming that *D(x)* and λ(x) are constants.

$$\frac{\partial}{\partial x}\left(D(x)\frac{\partial C(x)}{\partial x}\right) = / \,(x)C(x)$$

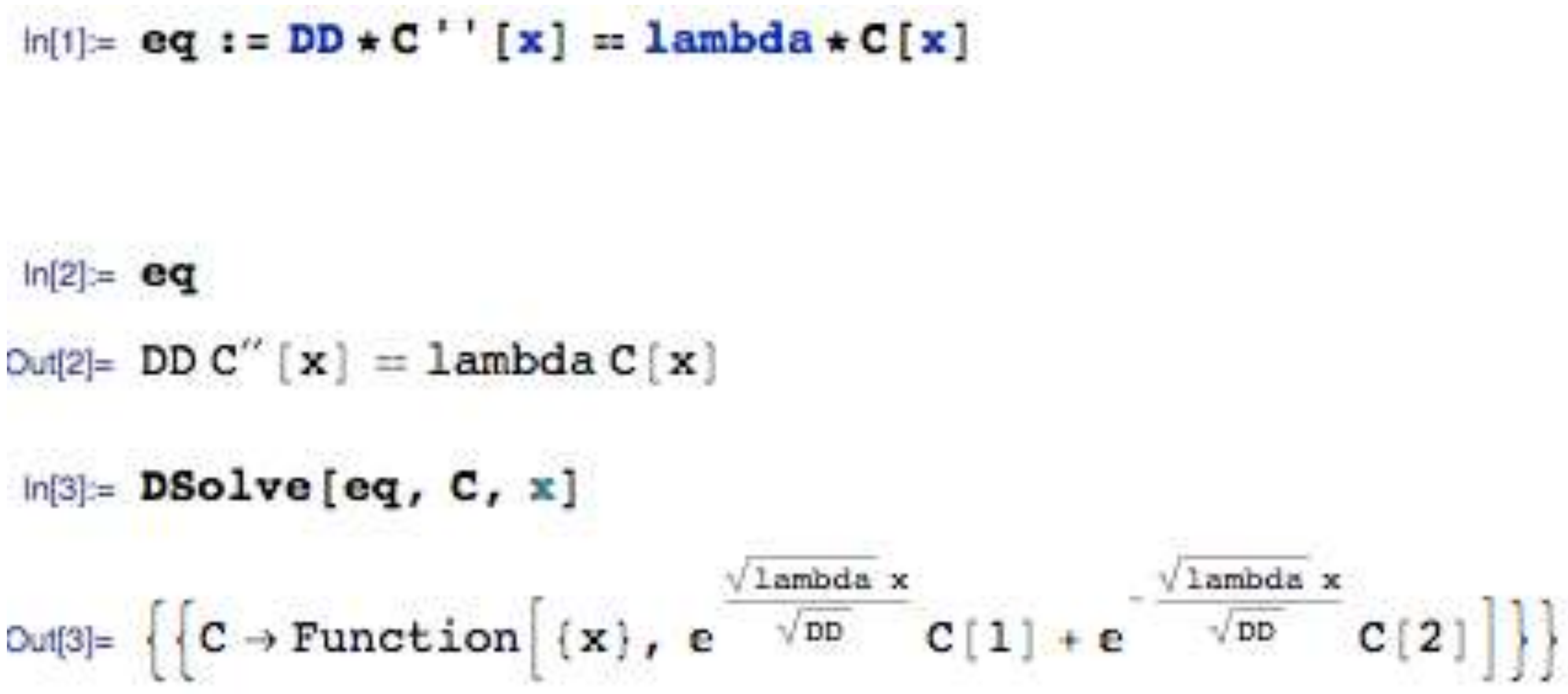
- Rearrange (we will assume D is constant, but may not always be)

$$\cancel{\frac{\P D(x)}{\P x}}\frac{\P C(x)}{\P x} + D(x)\frac{\P^2 C(x)}{\P x^2} = / \,(x)C(x)$$

In Mathematica

- Open a clean sheet
- Define you equation
  - Type eq:=DD*C"[x}==lambda*C[x]
  - Hit shift+return
- Confirm it is correct
  - Type eq
  - Hit shift +Return
- Solve the differential equation
  - Type Dsolve[eq,C,x]
  - Hit shift+return

Example



```
In[1]:= eq := DD * C''[x] == lambda * C[x]

In[2]:= eq

Out[2]= DD C''[x] == lambda C[x]

In[3]:= DSolve[eq, C, x]
```

Out[3]= $\left\{\left\{C \to \text{Function}\left[\{x\},\ e^{\frac{\sqrt{\text{lambda}}\ x}{\sqrt{\text{DD}}}}\ C[1] + e^{-\frac{\sqrt{\text{lambda}}\ x}{\sqrt{\text{DD}}}}\ C[2]\right]\right\}\right\}$

Now you try

- Solve the following differential equation

$$v\frac{\P C(x)}{\P x} = - \, / \, C^{2}(x)$$

You can also specify boundary conditions

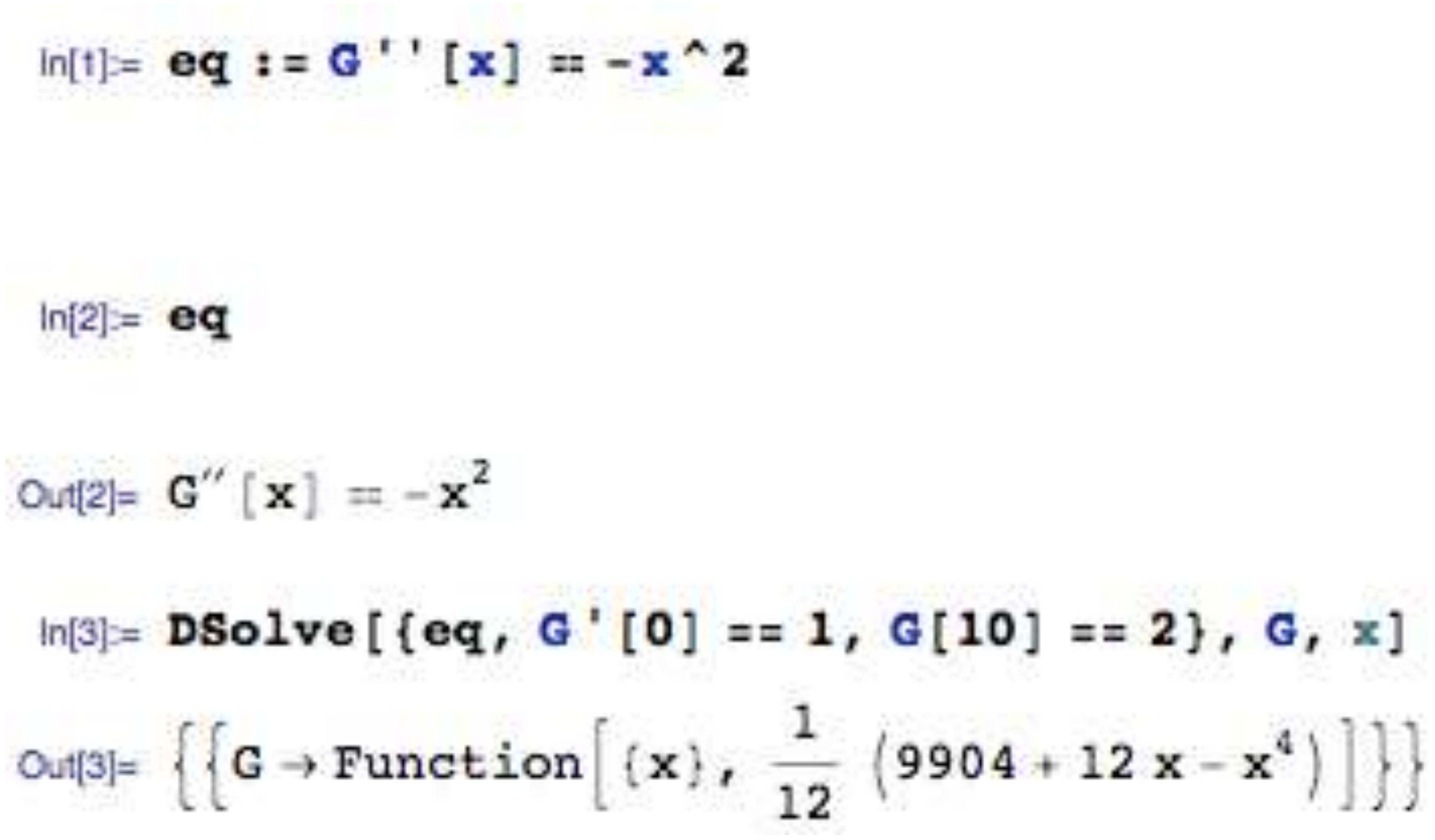
- Solve

$$\frac{\P^2 C(x)}{\P x^2} = - x^2$$

- Subject to boundary conditions

$$\left.\frac{\P C}{\P x}\right|_{x=0} = 1 \qquad C(x = 10) = 2$$

Example



```
In[1]:= eq := G''[x] == -x^2

In[2]:= eq

Out[2]= G''[x] == -x^2

In[3]:= DSolve[{eq, G'[0] == 1, G[10] == 2}, G, x]

Out[3]= {{G -> Function[{x}, 1/12 (9904 + 12 x - x^4)]}}
```

$$G''[x] = -x^2$$

$$\left\{\left\{G \to \text{Function}\left[\{x\}, \frac{1}{12}\left(9904 + 12x - x^4\right)\right]\right\}\right\}$$

You try

- Solve

$$\frac{\P C(x)}{\P x} = - e^{ax} C^4(x)$$

- Subject to

$$C(0) = 10$$

OK

- Now you're in good shape to solve pretty much any of the ordinary differential equations that you will face in this class
- You will have a few of these for homework just to make sure that you feel comfortable with it all.

Numerical Solutions

- Coupled Ordinary Differential equations.
- Imagine I have two (or more) concentrations that are coupled to each other such that

$$\frac{\P C_1(t)}{\P t} = -\,0.1C_1(t) + 0.2C_2(t) \qquad C_1(t=0)=1$$

$$\frac{\P C_2(t)}{\P t} = +0.1C_1(t) -\,0.2C_2(t) \qquad C_2(t=0)=0$$

Plot the concentrations over 100 times units

Before we solve this

- Look at the equations carefully.
- What do you think the solution is going to look like
- Do a few things
  - What is Steady State/Equilibrium
  - Where do both start, where do they end?
  - Draw what you think solution will look like.

Hopefully you remember from JJW

- You can approximate the derivative by (for small Δt)

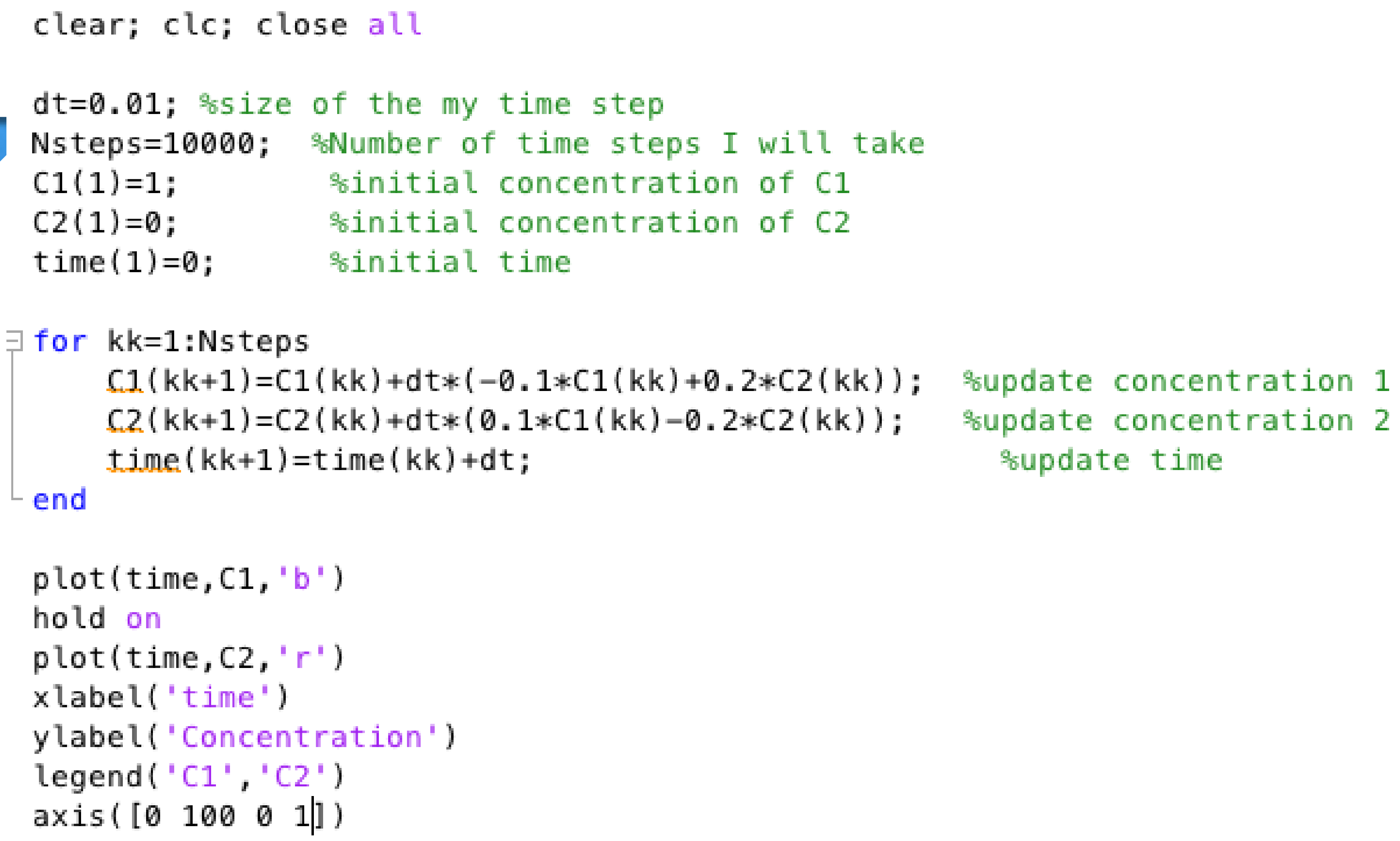
$$\frac{\partial C(t)}{\partial t} \approx \frac{C(t+\Delta t) - C(t)}{\Delta t}$$

- So I can write my equations as

$$\frac{C_1(t+\Delta t) - C_1(t)}{\Delta t} = -0.1C_1(t) + 0.2C_2(t)$$

$$\frac{C_2(t+\Delta t) - C_2(t)}{\Delta t} = 0.1C_1(t) - 0.2C_2(t)$$

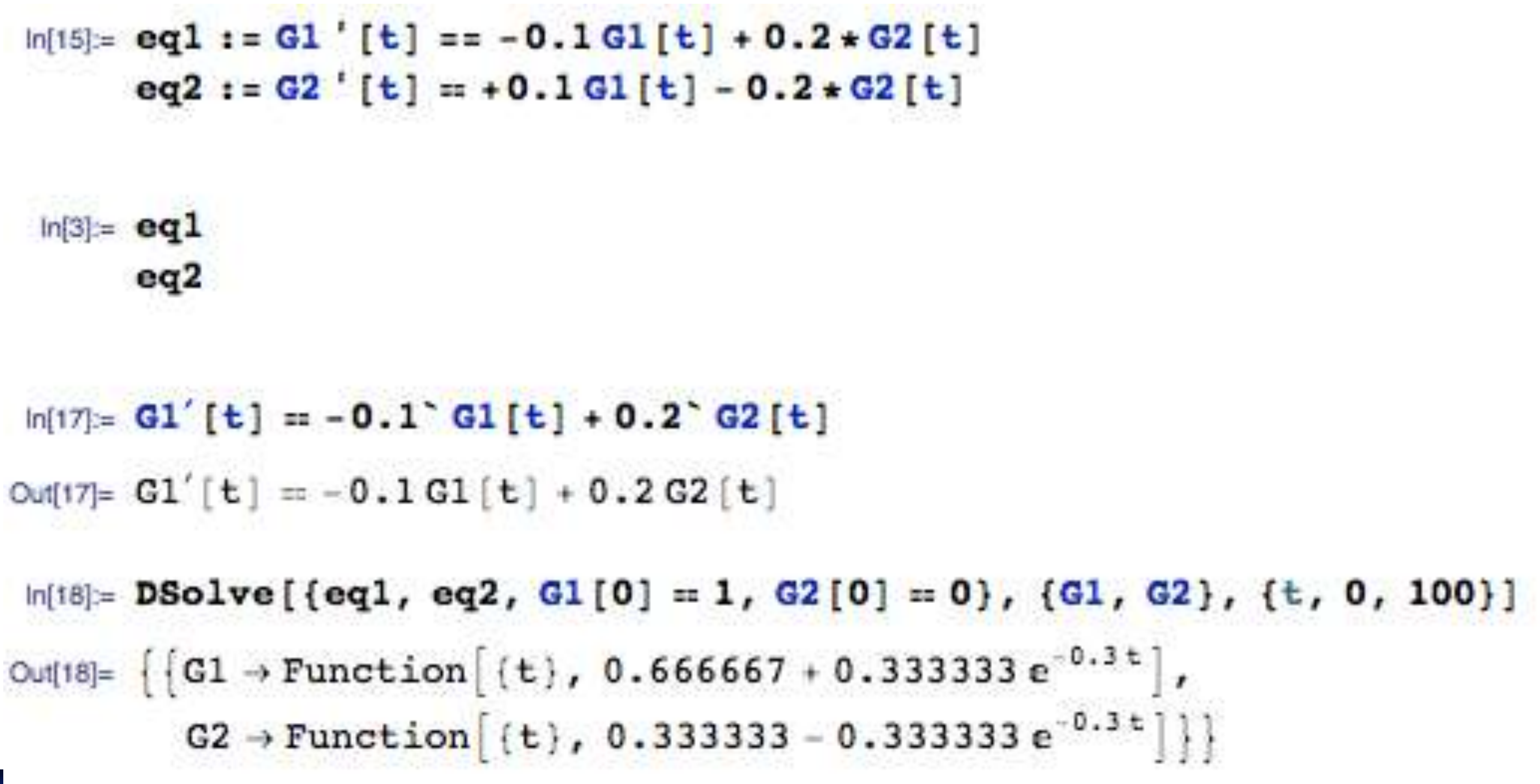
With these I can write a loop to solve for C1 and C2 at whatever time I want



```
clear; clc; close all

dt=0.01; %size of the my time step
Nsteps=10000;  %Number of time steps I will take
C1(1)=1;        %initial concentration of C1
C2(1)=0;        %initial concentration of C2
time(1)=0;      %initial time

for kk=1:Nsteps
    C1(kk+1)=C1(kk)+dt*(-0.1*C1(kk)+0.2*C2(kk));  %update concentration 1
    C2(kk+1)=C2(kk)+dt*(0.1*C1(kk)-0.2*C2(kk));   %update concentration 2
    time(kk+1)=time(kk)+dt;                         %update time
end

plot(time,C1,'b')
hold on
plot(time,C2,'r')
xlabel('time')
ylabel('Concentration')
legend('C1','C2')
axis([0 100 0 1])
```

Verify this solution

- With Mathematica you can easily show that the theoretical solution to this problem is



```
In[15]:= eq1 := G1 '[t] == -0.1 G1[t] + 0.2*G2[t]
         eq2 := G2 '[t] == +0.1 G1[t] - 0.2*G2[t]


 In[3]:= eq1
         eq2


In[17]:= G1′[t] == -0.1` G1[t] + 0.2` G2[t]

Out[17]= G1′[t] == -0.1 G1[t] + 0.2 G2[t]

In[18]:= DSolve[{eq1, eq2, G1[0] == 1, G2[0] == 0}, {G1, G2}, {t, 0, 100}]

Out[18]= {{G1 → Function[{t}, 0.666667 + 0.333333 e^-0.3 t],
           G2 → Function[{t}, 0.333333 - 0.333333 e^-0.3 t]}}
```

Ok you try this slightly harder one

- Solve

$$\frac{\P C_1(t)}{\P t} = -0.2C_1(t) + 0.3C_2(t) \qquad C_1(t=0) = 1$$

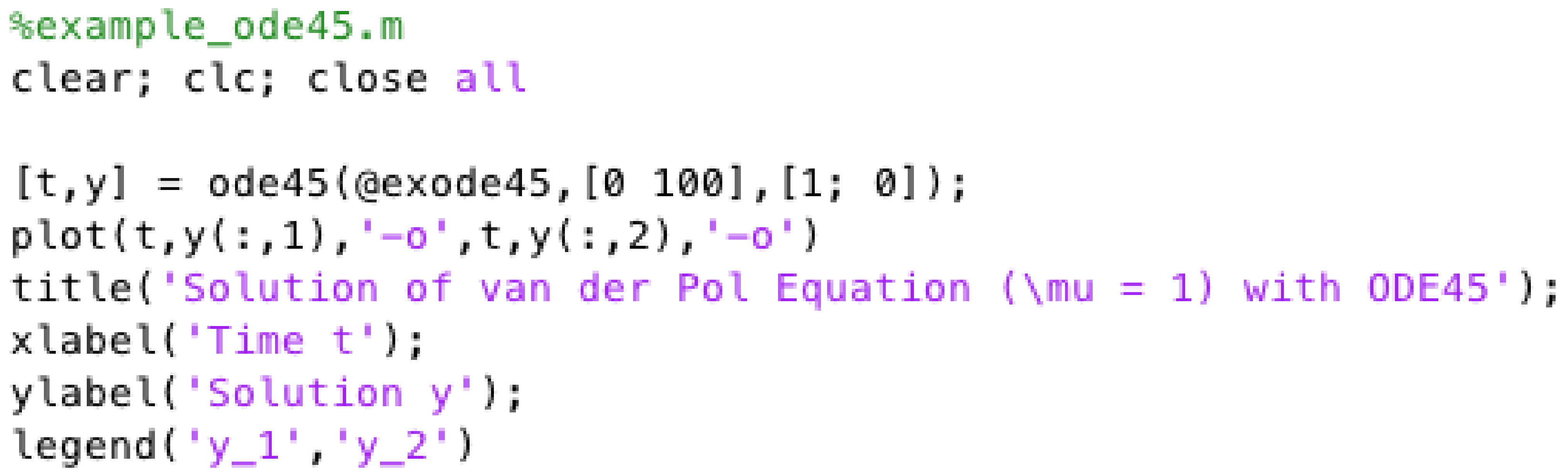
$$\frac{\P C_2(t)}{\P t} = +0.2C_1(t) - 0.4C_2(t) + 0.1C_3(t) \qquad C_2(t=0) = 1$$

$$\frac{\P C_3(t)}{\P t} = +0.1C_2(t) - 0.1C_3(t) \qquad C_3(t=0) = 2$$

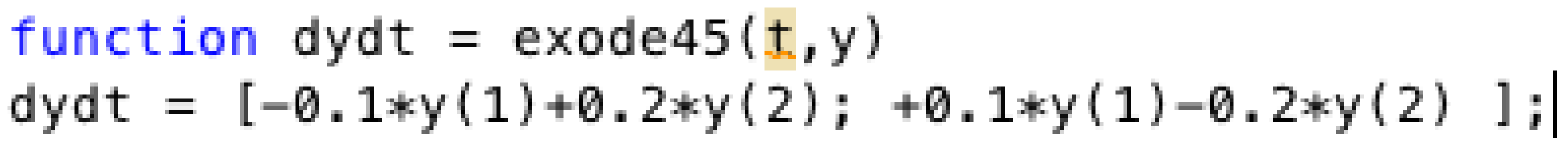
Alternative

- If you prefer not to write your own solver and use a more efficient one you can use Matlab's inbuilt ode45 function (https://www.mathworks.com/help/matlab/ref/ode45.html)
- Or
- You can use the numerical solver NDSolve in Mathematics (http://reference.wolfram.com/language/tutorial/NumericalSolutionOfDifferentialEquations.html) - I do not personally love this
- On the following slide I show you how to solve some of the problems we have looked at with both, but leave it to you to invest time and learn how to use them.

Matlab ode45

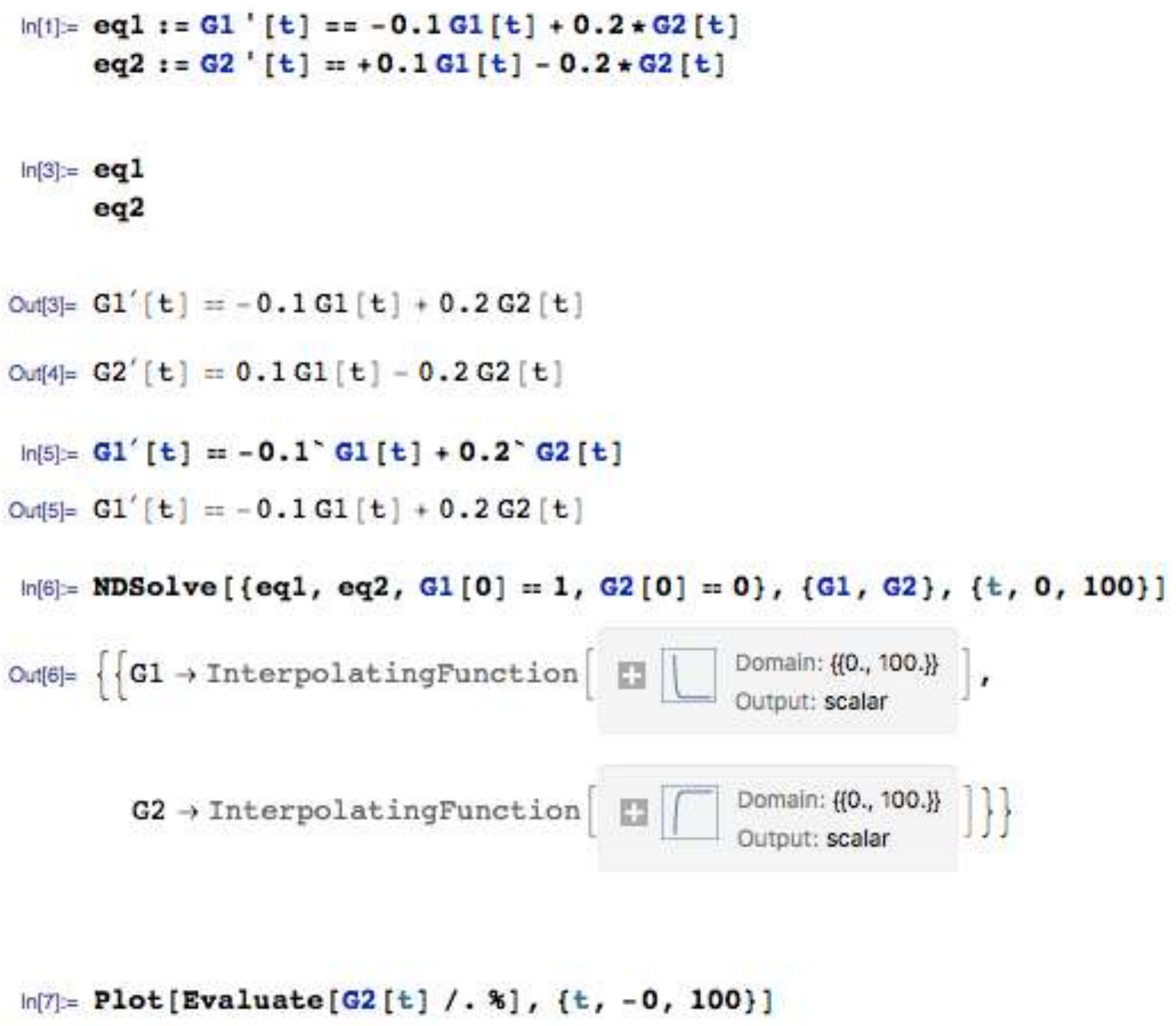


```matlab
%example_ode45.m
clear; clc; close all

[t,y] = ode45(@exode45,[0 100],[1; 0]);
plot(t,y(:,1),'-o',t,y(:,2),'-o')
title('Solution of van der Pol Equation (\mu = 1) with ODE45');
xlabel('Time t');
ylabel('Solution y');
legend('y_1','y_2')
```



```matlab
function dydt = exode45(t,y)
dydt = [-0.1*y(1)+0.2*y(2); +0.1*y(1)-0.2*y(2) ];
```

Mathematic AND Solve



```
In[1]:= eq1 := G1 '[t] == -0.1 G1[t] + 0.2 * G2[t]
        eq2 := G2 '[t] == +0.1 G1[t] - 0.2 * G2[t]

In[3]:= eq1
        eq2

Out[3]= G1′[t] == -0.1 G1[t] + 0.2 G2[t]

Out[4]= G2′[t] == 0.1 G1[t] - 0.2 G2[t]

In[5]:= G1′[t] == -0.1` G1[t] + 0.2` G2[t]

Out[5]= G1′[t] == -0.1 G1[t] + 0.2 G2[t]

In[6]:= NDSolve[{eq1, eq2, G1[0] == 1, G2[0] == 0}, {G1, G2}, {t, 0, 100}]

Out[6]= {{G1 → InterpolatingFunction[ Domain: {{0., 100.}}
                                      Output: scalar ],
          G2 → InterpolatingFunction[ Domain: {{0., 100.}}
                                      Output: scalar ]}}

In[7]:= Plot[Evaluate[G2[t] /. %], {t, -0, 100}]
```

THANK YOU