3. Using MATLAB to solve differential equations

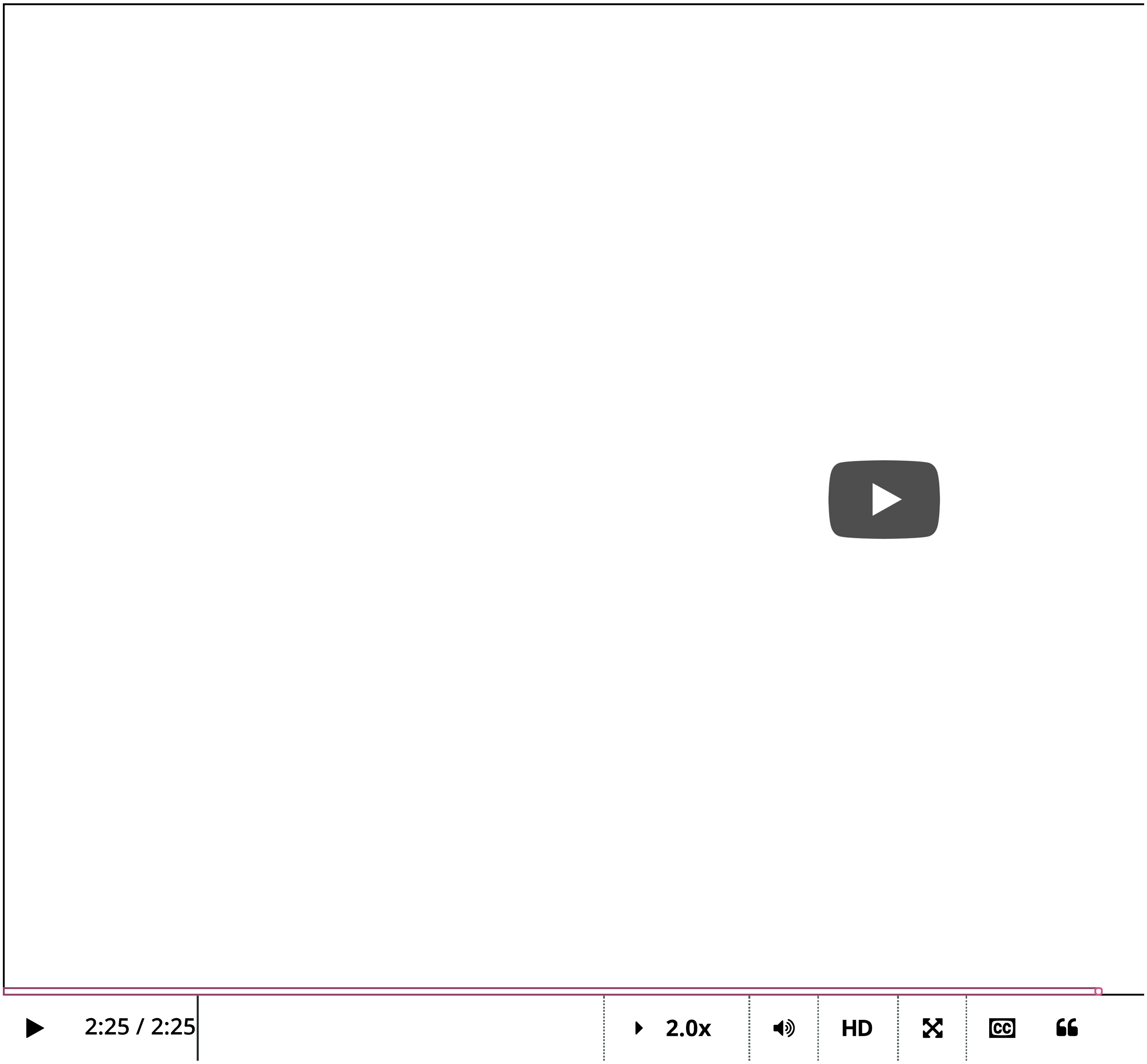
So far in this course, we have learned various techniques to solve systems of differential equations (DEs) by hand. However, DEs describing real-world systems are often very complex and are either too time-consuming to solve analytically, or have no closed-form solution at all! Luckily, many numerical algorithms have been developed so that approximate solutions to DEs can be quickly obtained using a computer.

In this lecture, we will practice using one of MATLABs built-in ODE solvers, called **ODE45**. The numeric solver **ODE45** can solve any first order system of the form,

$$\dot{\mathbf{x}} = \mathbf{f}\left(t, \mathbf{x}\right),$$

and is an accurate and highly optimized numerical solver. Remember that any higher order DE can always be converted into an equivalent system of first order DEs. **ODE45** uses a combination of 4th and 5th order Runge-Kutta methods to approximate the solution of a DE and estimate the error made at each time-step. It also uses a variable time-step to minimize the number of computations while keeping the error of the numerical solution below a desired threshold.

Numerically solving first order ODEs



Problem1 (External resource) (1.0 points possible)

A first problem using ODE45

Let's see an example of how we can use **ODE45** to numerically solve the simple DE,

$$\dot{x} = tx^2, \quad x(0) = -2,$$

on the interval $t \in [0, 2]$. You can verify that the analytic solution of this IVP is $x(t) = -\frac{2}{1+t^2}$. The script below shows the code that is needed to run **ODE45**.

It is missing two components that you must fill in yourself. Specifically, you must:

1. Define the value `x_0` which specifies the value $x(0)$.
2. Define a 1×2 vector `tspan` whose first and second elements are the start and end times of the time interval on which to solve the DE.

If your script runs correctly, you will see a plot comparing the analytic solution to the numerical solution produced by **ODE45**, a plot showing the relative error between the two solutions, and the time it took **ODE45** to compute the numerical solution. The commands `tic` and `toc` time how long it took the function **ODE45** to run. You might have noticed that the entire MATLAB script took much longer to run than the time required for **ODE45** to run. This is because you are writing the MATLAB script on an internet browser and the code must be sent to, executed on, and sent back from a remote server. Note the small error of the numerical solution and how little time the solver required!

Your Script

Save Reset MATLAB Documentation (https://www.mathworks.com/help/)

```
%Numerically solve DE and time how long it takes
x0 = -2;                                %The initial condition, x(0)
tspan = [0,2];                         %The time interval, tspan
tic;                                   %Start timer
[t,x] = ode45(@(t,x) t*x^2,tspan,x0); %Use ODE45 to compute numerical solution
timeElapsed = toc;                     %Stop timer
disp(['It took ODE45 ',num2str(timeElapsed,3), ' seconds to compute the solution'

%Enter analytic solution
xTrue = -2./(1+t.^2);

%Plot results
%You do not need to edit any of the code below which creates the plots for you.
figure(1)
plot(t,x,'bo','markersize',10); hold on;
plot(t,xTrue,'r','linewidth',3);
legend('Numerical Solution','Exact Solution','location','northwest');
```

```
legend('Numerical Solution','Exact Solution','location','northwest');
xlabel('$t$','interpreter','latex'); ylabel('$x(t)$','interpreter','latex')
title('Comparison of Solutions','interpreter','latex')
set(gca,'fontsize',25)

figure(2)
plot(t,abs(x-xTrue)./xTrue,'ro-','linewidth',3,'markersize',10);
xlabel('$t$','interpreter','latex'); ylabel('$|x(t)-x_{true}|/x_{true}$','interpr
title('Relative Error','interpreter','latex')
set(gca,'fontsize',25)

```

▶ Run Script ? ()

Assessment: Correct

Submit ? ()

- Correct definition of x0
- Correct definition of tspan
- Correct definition of xTrue

3. Using MATLAB to solve differential equations

Hide Discussion

Topic: Unit 3: Solving systems of first order ODEs using matrix methods / 3. Using MATLAB to solve differential equations

Add a Post

Show all posts ▼ by recent activity ▼

the problem is missing
It seems that something should be uploaded. The script is missing. 3

Learn About Verified Certificates