Exploring Differential Equations with Interactive Jupyter Notebooks

Adam Spiegler
University of Colorado Denver
adam.spiegler@ucdenver.edu

Department of Mathematical and Statistical Sciences
COLLEGE OF LIBERAL ARTS AND SCIENCES
UNIVERSITY OF COLORADO **DENVER**

https://aspiegler.github.io/Exploring-Differential-Equations/

COLORADO
Department of Higher Education

Outline

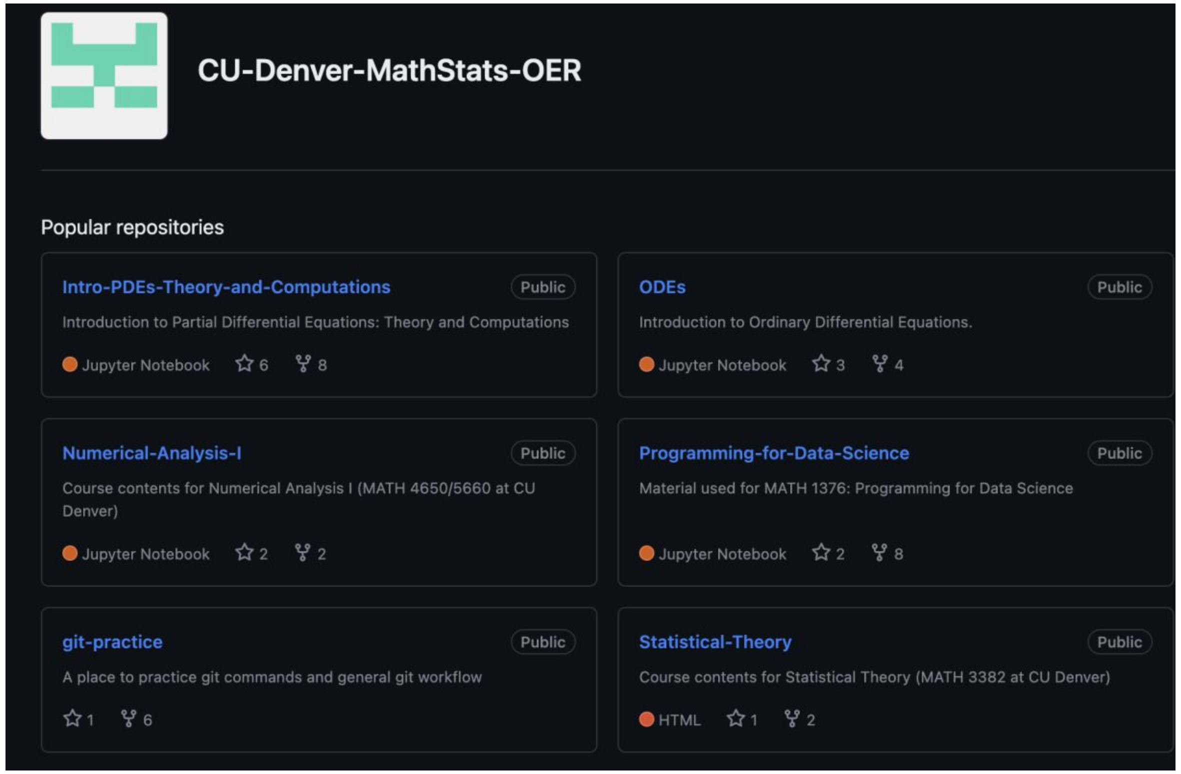
- Give a brief background of the program redesign at CU Denver.
- Demo the course materials (and tour Google Colab).
- Provide access to the materials.
- Share observations about using the materials in the classroom.

“What would you like to see improved in our program?”

- “The improvement that **the mathematics major desperately needs is a heavier focus on coding in python and R.** … There absolutely needs to be a required in department coding course that goes over the fundamentals of python and R.”
- “Maybe an intro to Python class or something. **My only exposure to Python in the math department was through Math Clinic** and that goes 0-60 faster than a sports car.”
- “**Integrate coding into earlier courses**. I hadn't learned python from any course and was expected to use it in Math Clinic for an extremely challenging project. The best course experiences I had were courses in which coding played a large role.”

Initial Courses Targeted

- 4000/5000 levels
  - Applied Regression Analysis (Year 1, R)
  - Machine Learning Methods (Year 2, Python)
  - Partial Differential Equations (Year 1, Python)
  - Numerical Analysis I (Year 1, Python)
- 3000 level
  - Linear Algebra (Year 2, Python)
  - Data Wrangling and Visualization (Year 0, R)
  - Elementary Differential Equations (Year 1, Python)
  - Statistical Theory (Year 2, R)
- 1000/2000 level
  - Intro to Prog for Data Science (Year 0, Python)



CU-Denver-MathStats-OER

Popular repositories

Intro-PDEs-Theory-and-Computations Public
Introduction to Partial Differential Equations: Theory and Computations
Jupyter Notebook ☆ 6 ⑂ 8

ODEs Public
Introduction to Ordinary Differential Equations.
Jupyter Notebook ☆ 3 ⑂ 4

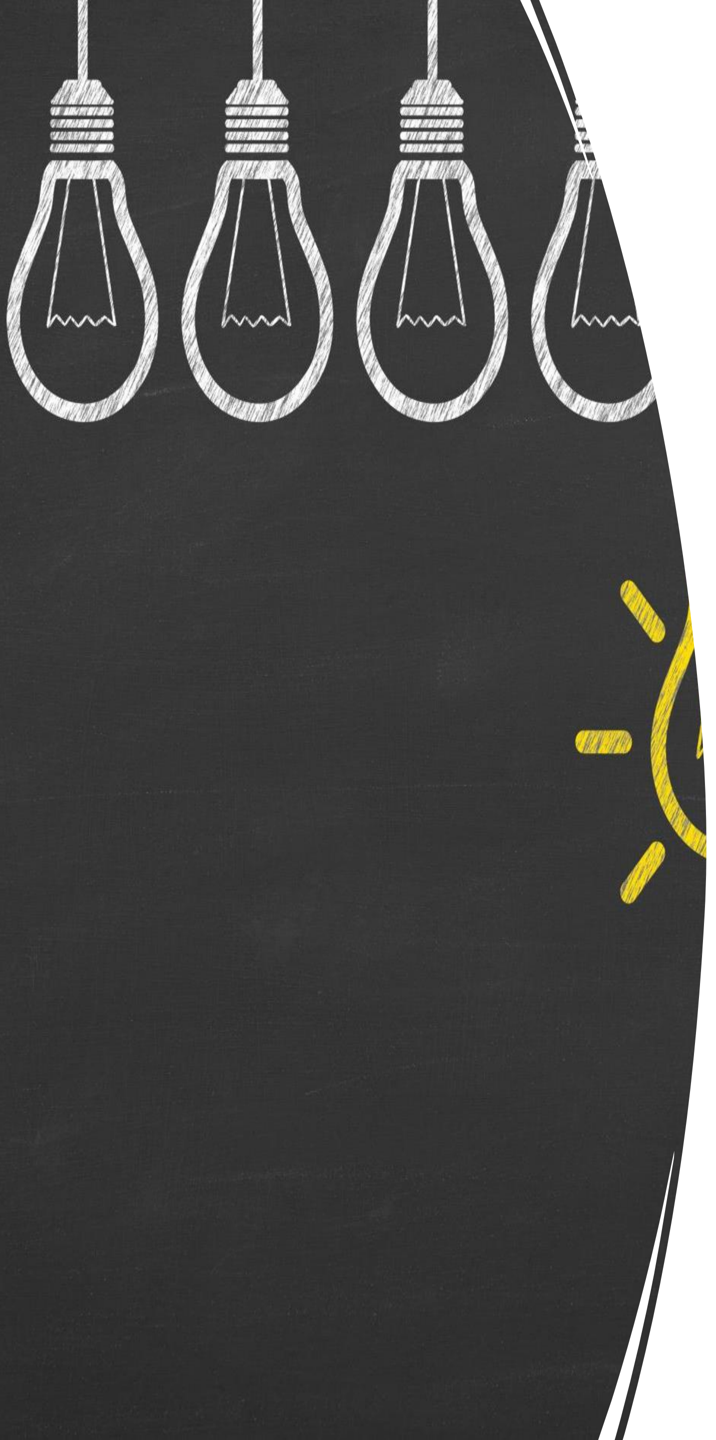
Numerical-Analysis-I Public
Course contents for Numerical Analysis I (MATH 4650/5660 at CU Denver)
Jupyter Notebook ☆ 2 ⑂ 2

Programming-for-Data-Science Public
Material used for MATH 1376: Programming for Data Science
Jupyter Notebook ☆ 2 ⑂ 8

git-practice Public
A place to practice git commands and general git workflow
☆ 1 ⑂ 6

Statistical-Theory Public
Course contents for Statistical Theory (MATH 3382 at CU Denver)
HTML ☆ 1 ⑂ 2

https://github.com/CU-Denver-MathStats-OER



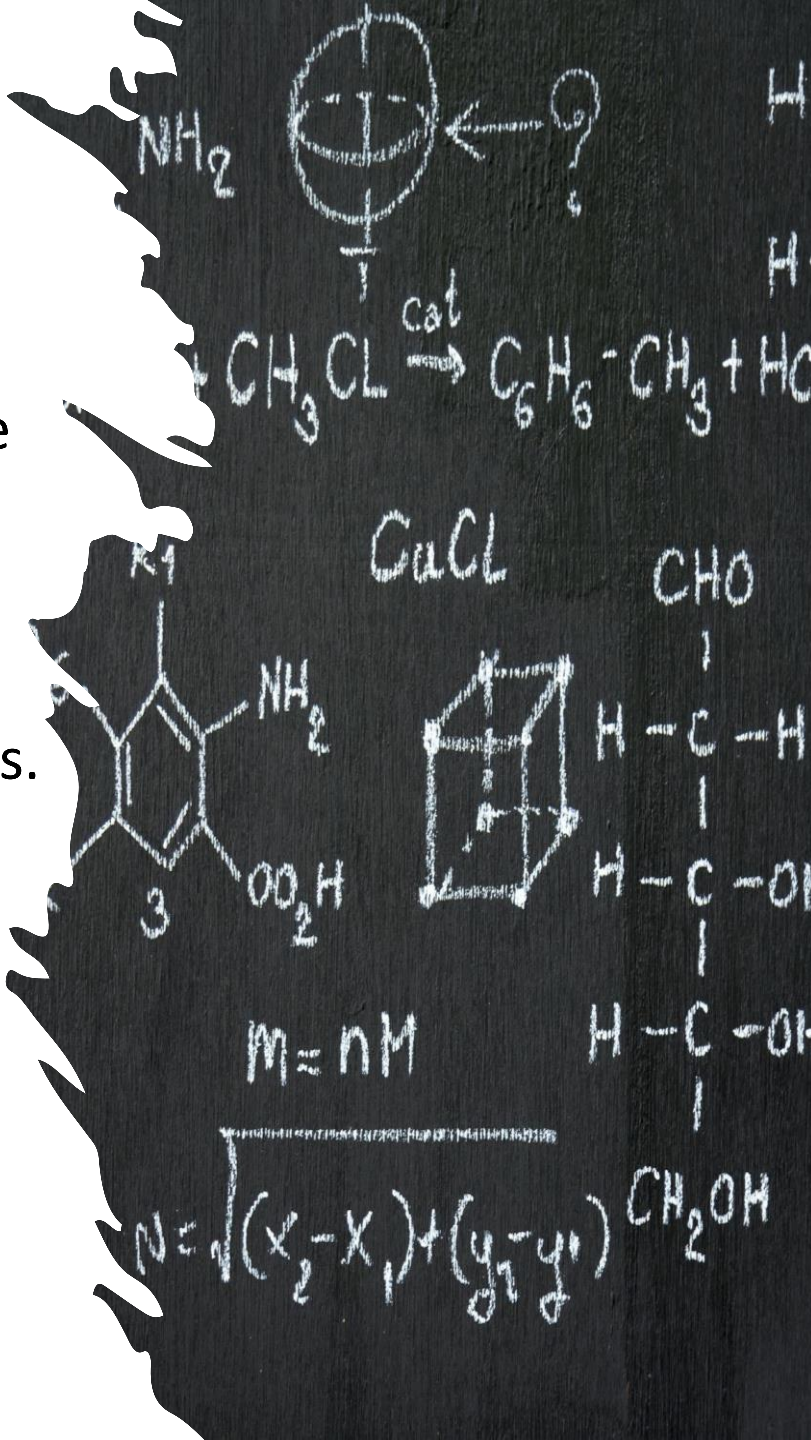
Project Motivation and Vision

- Creation and dissemination of free knowledge.
- Equip students with knowledge & skills necessary to compete in a global marketplace.
- Train students as “mathematical experimentalists” across multiple levels of math, data science and statistics curricula.
- Utilize “virtual laboratories” to enable an active computational learning environment.

Experimenting with Mathematics

The objectives of experimental mathematics are generally to make mathematics **more tangible, lively and fun**.

- Discover patterns and relationships.
- Using graphical displays to investigate mathematical concepts.
- Developing and testing conjectures.
- Exploring results to help construct proofs.
- Confirm analytically derived results.
- Gain insight and intuition.
- **Bridge the divide between theory and practice.**



Jupyter, Equity in Computing, and Colab

https://jupyter.org/

https://colab.research.google.com/

Equity in Computing

We want to ensure the computing technology and educational resources we develop are accessible and usable **no matter the socioeconomic circumstances of the user or institution**.

Jupyter

- is free software for interactive computing.
- supports over 40 programming languages including Python, R, and Julia.
- is a dynamic environment to weave together narrative text, executable code, widgets, videos, and more.

Google Colaboratory (Colab)

- is a free cloud-based version of Jupyter that any institution or user can utilize.
- levels the playing field and reduces barriers to access.

What's novel about our approach?

Provides instructors and students an environment to **weave together** narrative text, executable code, interactive visualizations, and/or videos all in one document.

Materials are **FREE**! No additional technology required. Course materials are delivered in Google Colaboratory, a free, cloud-based application.

The **flexibility** to design materials for a variety of course formats.

Create a **dynamic** learning environment to bridge the divide between theory and practice.

Structure of the course

- Math, Engineering, Economics and Physics students
- No prior programming experience needed (or required).
- 40% Math BS students first generation.
- 58% Math BS students transfer students.
- Class met twice a week for 1 hr 15 min each class.
- Students watch short video with quiz prior to each class.
- Active learning working in groups for much of the class time.
- The technology requirement is a tablet/computer/phone that can connected to the internet.

Accessing Differential Equations Materials

HTML text:

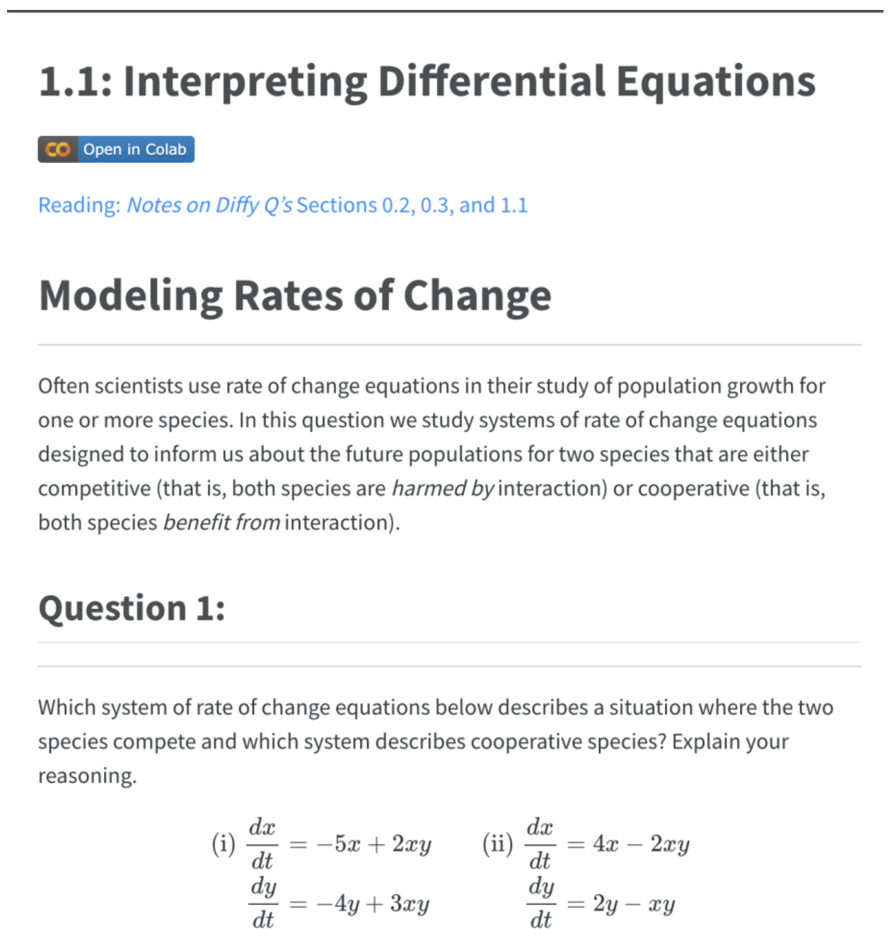
https://aspiegler.github.io/Exploring-Differential-Equations/

GitHub:

https://github.com/CU-Denver-MathStats-OER/ODEs

Open in Colab

Static HTML Text



1.1: Interpreting Differential Equations

Reading: *Notes on Diffy Q's* Sections 0.2, 0.3, and 1.1

Modeling Rates of Change

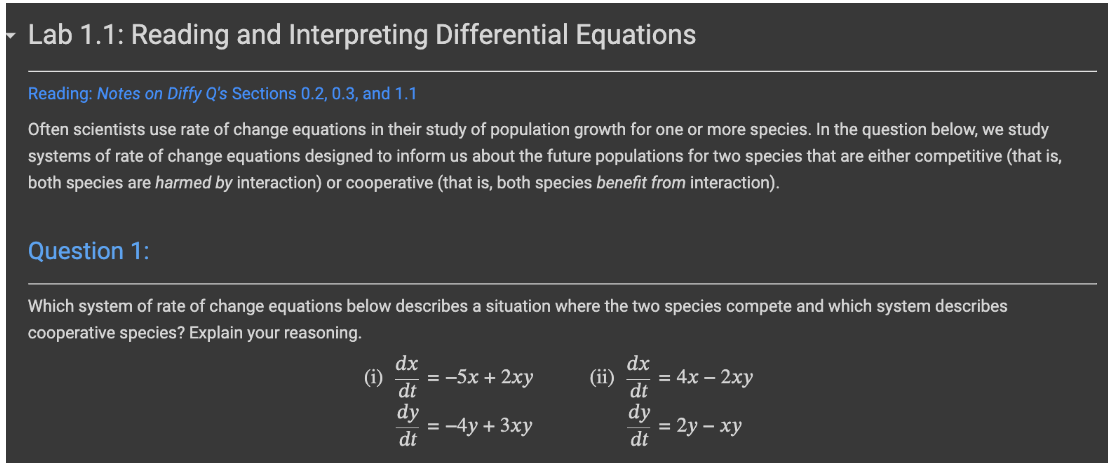
Often scientists use rate of change equations in their study of population growth for one or more species. In this question we study systems of rate of change equations designed to inform us about the future populations for two species that are either competitive (that is, both species are *harmed by* interaction) or cooperative (that is, both species *benefit from* interaction).

Question 1:

Which system of rate of change equations below describes a situation where the two species compete and which system describes cooperative species? Explain your reasoning.

$$\text{(i)}\ \begin{aligned}\frac{dx}{dt} &= -5x + 2xy \\ \frac{dy}{dt} &= -4y + 3xy\end{aligned} \qquad \text{(ii)}\ \begin{aligned}\frac{dx}{dt} &= 4x - 2xy \\ \frac{dy}{dt} &= 2y - xy\end{aligned}$$

Interactive Colab Version



Lab 1.1: Reading and Interpreting Differential Equations

Reading: *Notes on Diffy Q's* Sections 0.2, 0.3, and 1.1

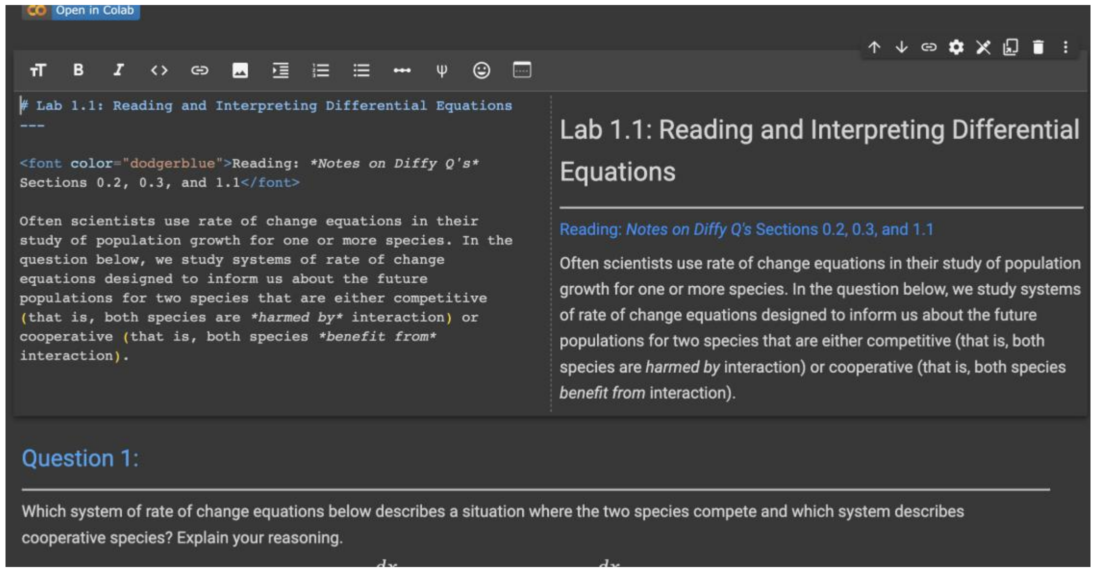
Often scientists use rate of change equations in their study of population growth for one or more species. In the question below, we study systems of rate of change equations designed to inform us about the future populations for two species that are either competitive (that is, both species are *harmed by* interaction) or cooperative (that is, both species *benefit from* interaction).

Question 1:

Which system of rate of change equations below describes a situation where the two species compete and which system describes cooperative species? Explain your reasoning.

$$\text{(i)}\ \begin{aligned}\frac{dx}{dt} &= -5x + 2xy \\ \frac{dy}{dt} &= -4y + 3xy\end{aligned} \qquad \text{(ii)}\ \begin{aligned}\frac{dx}{dt} &= 4x - 2xy \\ \frac{dy}{dt} &= 2y - xy\end{aligned}$$

Editing Text Cells



```
# Lab 1.1: Reading and Interpreting Differential Equations
---

<font color="dodgerblue">Reading: *Notes on Diffy Q's*
Sections 0.2, 0.3, and 1.1</font>

Often scientists use rate of change equations in their
study of population growth for one or more species. In the
question below, we study systems of rate of change
equations designed to inform us about the future
populations for two species that are either competitive
(that is, both species are *harmed by* interaction) or
cooperative (that is, both species *benefit from*
interaction).
```

Lab 1.1: Reading and Interpreting Differential Equations

Reading: *Notes on Diffy Q's* Sections 0.2, 0.3, and 1.1

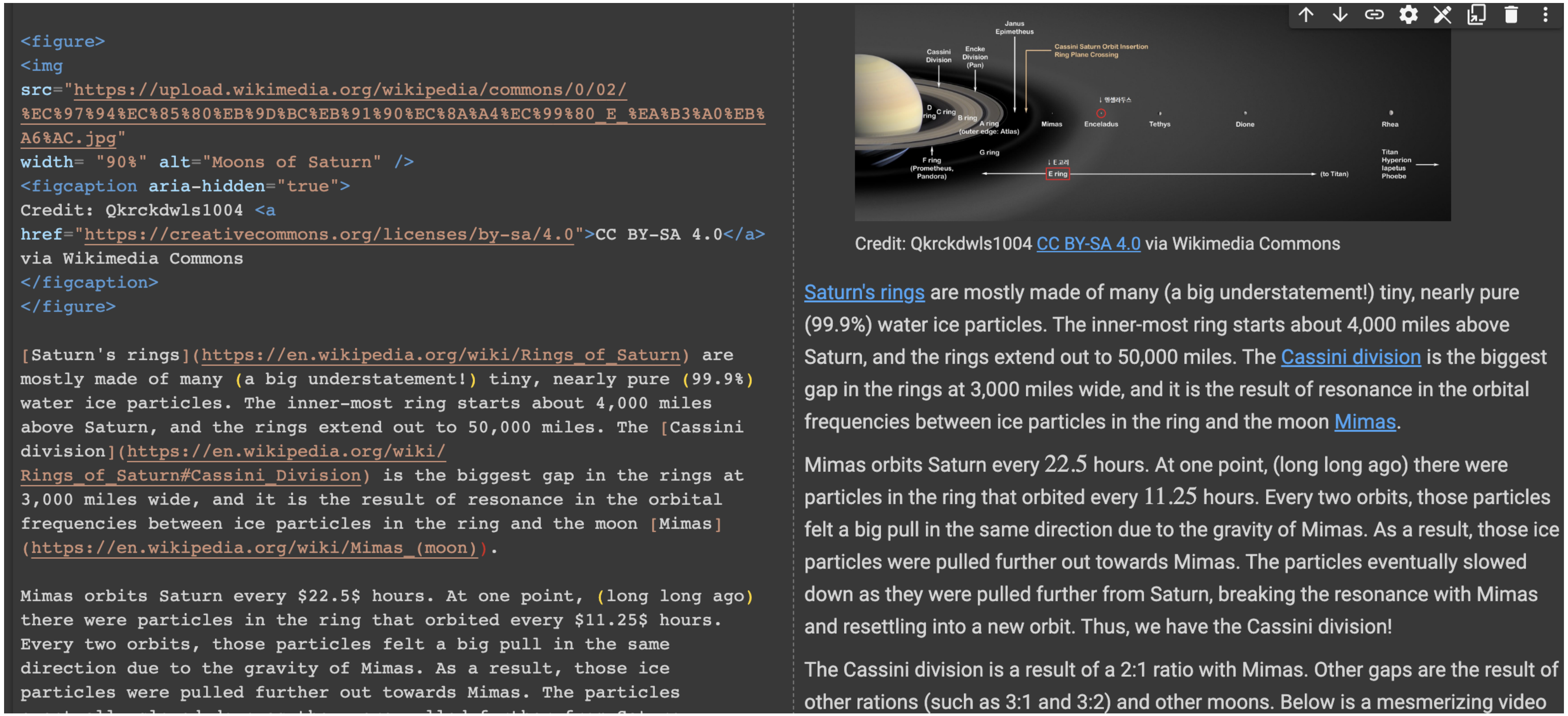
Often scientists use rate of change equations in their study of population growth for one or more species. In the question below, we study systems of rate of change equations designed to inform us about the future populations for two species that are either competitive (that is, both species are *harmed by* interaction) or cooperative (that is, both species *benefit from* interaction).

Question 1:

Which system of rate of change equations below describes a situation where the two species compete and which system describes cooperative species? Explain your reasoning.

https://githubtocolab.com/CU-Denver-MathStats-OER/ODEs/blob/main/Chp1/01-What-is-a-Differential-Equation.ipynb

Including Images in a Colab Text Cell



```
<figure>
<img
src="https://upload.wikimedia.org/wikipedia/commons/0/02/
%EC%97%94%EC%85%80%EB%9D%BC%EB%91%90%EC%8A%A4%EC%99%80_E_%EA%B3%A0%EB%
A6%AC.jpg"
width= "90%" alt="Moons of Saturn" />
<figcaption aria-hidden="true">
Credit: Qkrckdwls1004 <a
href="https://creativecommons.org/licenses/by-sa/4.0">CC BY-SA 4.0</a>
via Wikimedia Commons
</figcaption>
</figure>

[Saturn's rings](https://en.wikipedia.org/wiki/Rings_of_Saturn) are
mostly made of many (a big understatement!) tiny, nearly pure (99.9%)
water ice particles. The inner-most ring starts about 4,000 miles
above Saturn, and the rings extend out to 50,000 miles. The [Cassini
division](https://en.wikipedia.org/wiki/
Rings_of_Saturn#Cassini_Division) is the biggest gap in the rings at
3,000 miles wide, and it is the result of resonance in the orbital
frequencies between ice particles in the ring and the moon [Mimas]
(https://en.wikipedia.org/wiki/Mimas_(moon)).

Mimas orbits Saturn every $22.5$ hours. At one point, (long long ago)
there were particles in the ring that orbited every $11.25$ hours.
Every two orbits, those particles felt a big pull in the same
direction due to the gravity of Mimas. As a result, those ice
particles were pulled further out towards Mimas. The particles
```

Credit: Qkrckdwls1004 CC BY-SA 4.0 via Wikimedia Commons

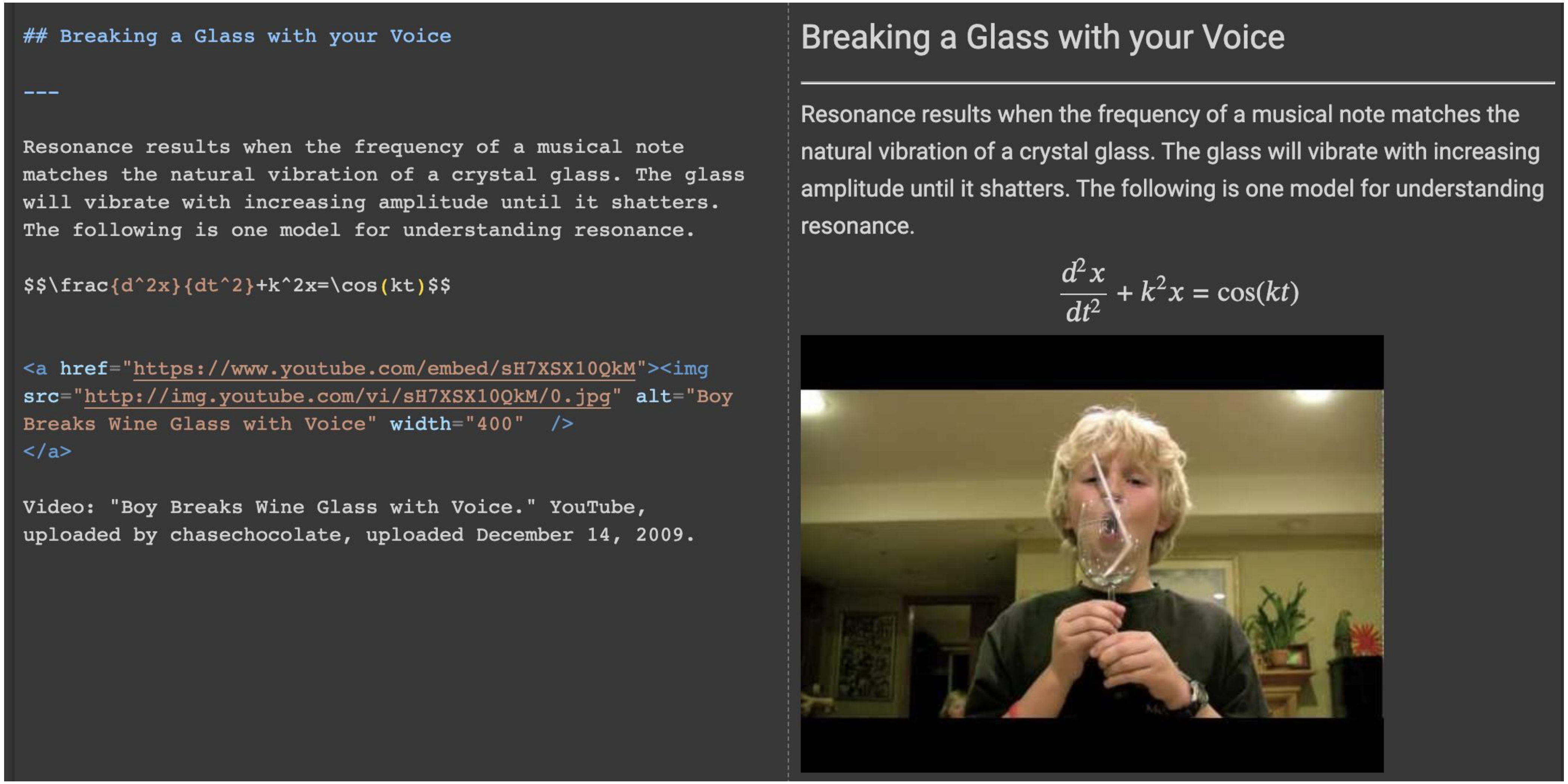
Saturn's rings are mostly made of many (a big understatement!) tiny, nearly pure (99.9%) water ice particles. The inner-most ring starts about 4,000 miles above Saturn, and the rings extend out to 50,000 miles. The Cassini division is the biggest gap in the rings at 3,000 miles wide, and it is the result of resonance in the orbital frequencies between ice particles in the ring and the moon Mimas.

Mimas orbits Saturn every 22.5 hours. At one point, (long long ago) there were particles in the ring that orbited every 11.25 hours. Every two orbits, those particles felt a big pull in the same direction due to the gravity of Mimas. As a result, those ice particles were pulled further out towards Mimas. The particles eventually slowed down as they were pulled further from Saturn, breaking the resonance with Mimas and resettling into a new orbit. Thus, we have the Cassini division!

The Cassini division is a result of a 2:1 ratio with Mimas. Other gaps are the result of other rations (such as 3:1 and 3:2) and other moons. Below is a mesmerizing video

https://githubtocolab.com/CU-Denver-MathStats-OER/ODEs/blob/main/Chp2/13-Mass-Spring.ipynb

Embed Videos in Colab



```
## Breaking a Glass with your Voice

---

Resonance results when the frequency of a musical note
matches the natural vibration of a crystal glass. The glass
will vibrate with increasing amplitude until it shatters.
The following is one model for understanding resonance.

$$\frac{d^2x}{dt^2}+k^2x=\cos(kt)$$


<a href="https://www.youtube.com/embed/sH7XSX10QkM"><img
src="http://img.youtube.com/vi/sH7XSX10QkM/0.jpg" alt="Boy
Breaks Wine Glass with Voice" width="400"  />
</a>

Video: "Boy Breaks Wine Glass with Voice." YouTube,
uploaded by chasechocolate, uploaded December 14, 2009.
```

Breaking a Glass with your Voice

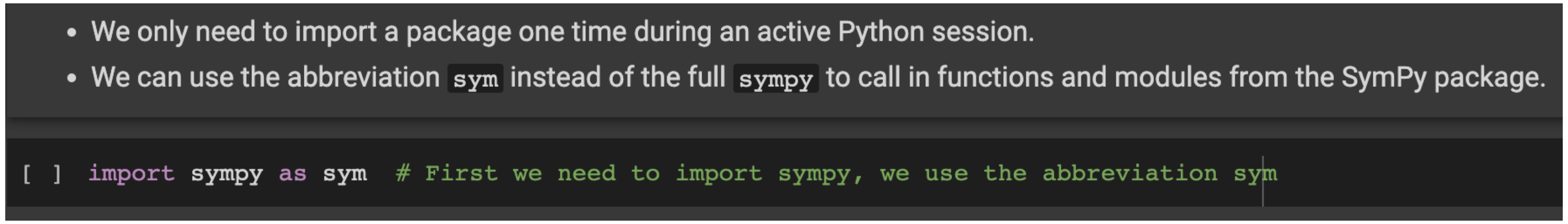
Resonance results when the frequency of a musical note matches the natural vibration of a crystal glass. The glass will vibrate with increasing amplitude until it shatters. The following is one model for understanding resonance.

$$\frac{d^2x}{dt^2}+k^2x=\cos(kt)$$

https://githubtocolab.com/CU-Denver-MathStats-OER/ODEs/blob/main/Chp2/13-Mass-Spring.ipynb

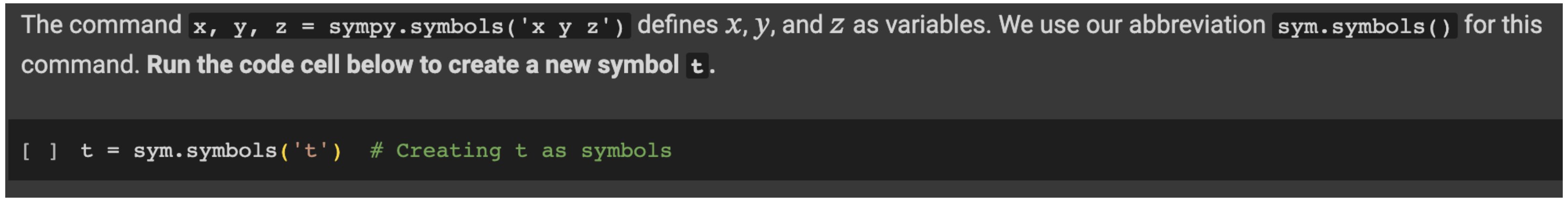
Using Code Cells in Colab

Lab 1 Student Introduction to Python



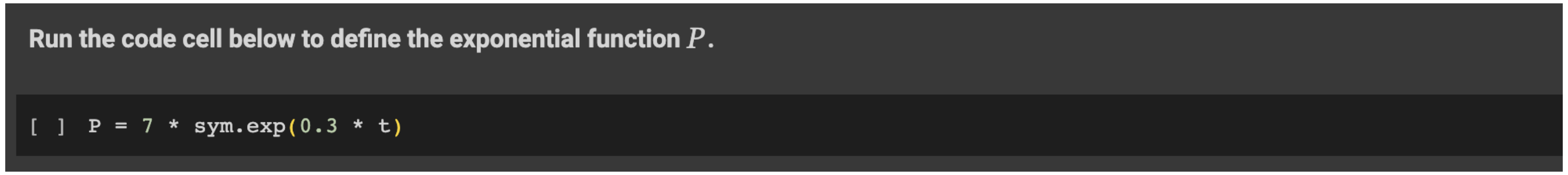
- We only need to import a package one time during an active Python session.
- We can use the abbreviation `sym` instead of the full `sympy` to call in functions and modules from the SymPy package.

```python
import sympy as sym  # First we need to import sympy, we use the abbreviation sym
```



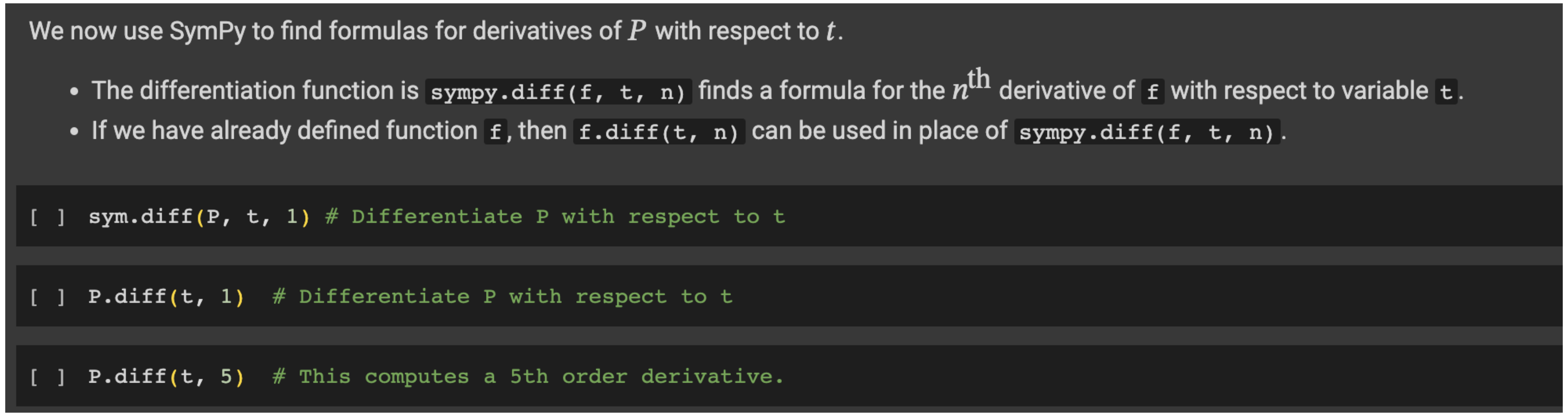
The command `x, y, z = sympy.symbols('x y z')` defines x, y, and z as variables. We use our abbreviation `sym.symbols()` for this command. **Run the code cell below to create a new symbol `t`.**

```python
t = sym.symbols('t')  # Creating t as symbols
```



Run the code cell below to define the exponential function P.

```python
P = 7 * sym.exp(0.3 * t)
```



We now use SymPy to find formulas for derivatives of P with respect to t.

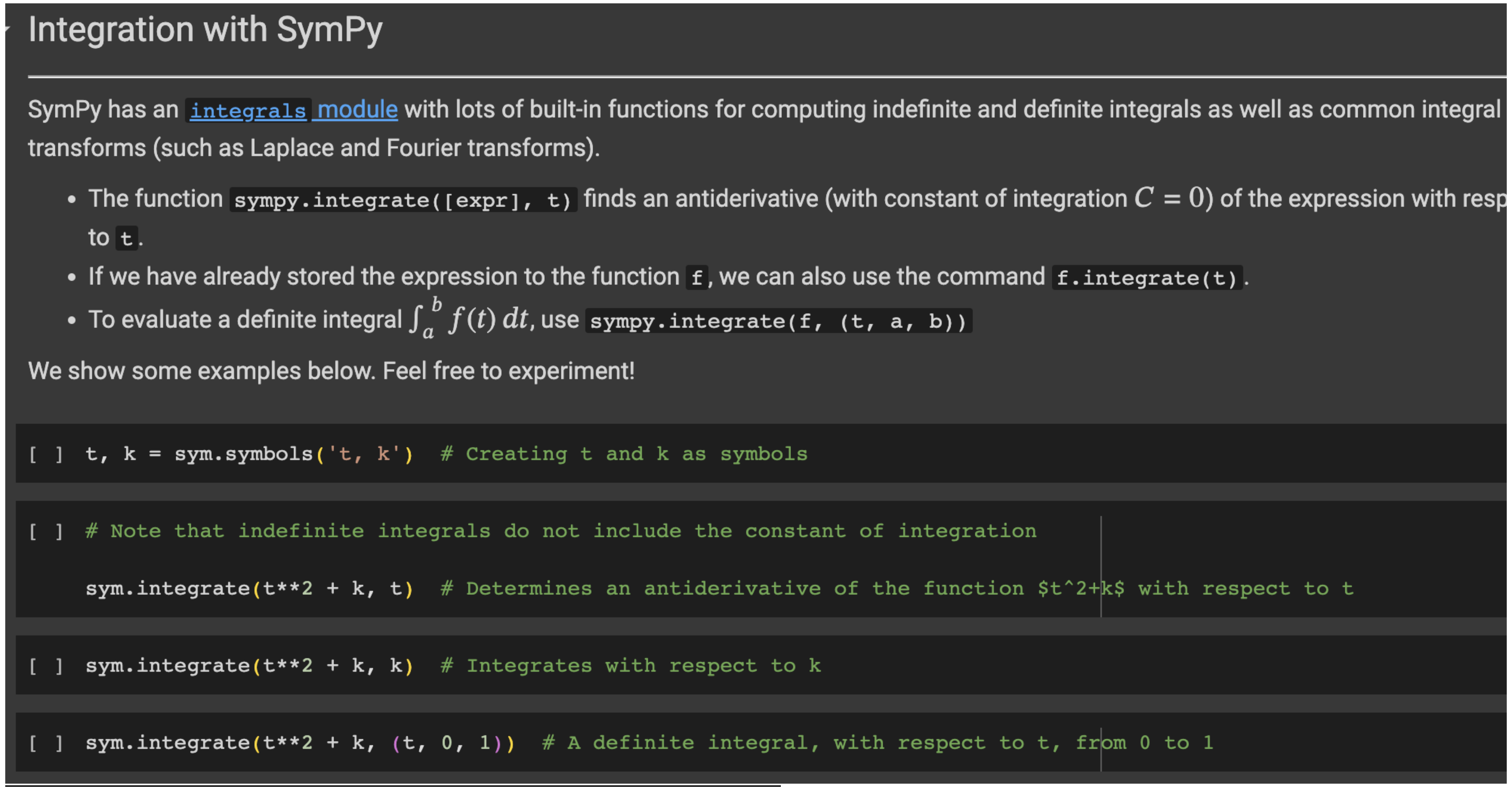
- The differentiation function is `sympy.diff(f, t, n)` finds a formula for the n^{th} derivative of `f` with respect to variable `t`.
- If we have already defined function `f`, then `f.diff(t, n)` can be used in place of `sympy.diff(f, t, n)`.

```python
sym.diff(P, t, 1) # Differentiate P with respect to t
```

```python
P.diff(t, 1)  # Differentiate P with respect to t
```

```python
P.diff(t, 5)  # This computes a 5th order derivative.
```

https://githubtocolab.com/CU-Denver-MathStats-OER/ODEs/blob/main/Chp1/01-What-is-a-Differential-Equation.ipynb



Integration with SymPy

SymPy has an `integrals` module with lots of built-in functions for computing indefinite and definite integrals as well as common integral transforms (such as Laplace and Fourier transforms).

- The function `sympy.integrate([expr], t)` finds an antiderivative (with constant of integration $C = 0$) of the expression with resp to `t`.
- If we have already stored the expression to the function `f`, we can also use the command `f.integrate(t)`.
- To evaluate a definite integral $\int_a^b f(t)\,dt$, use `sympy.integrate(f, (t, a, b))`

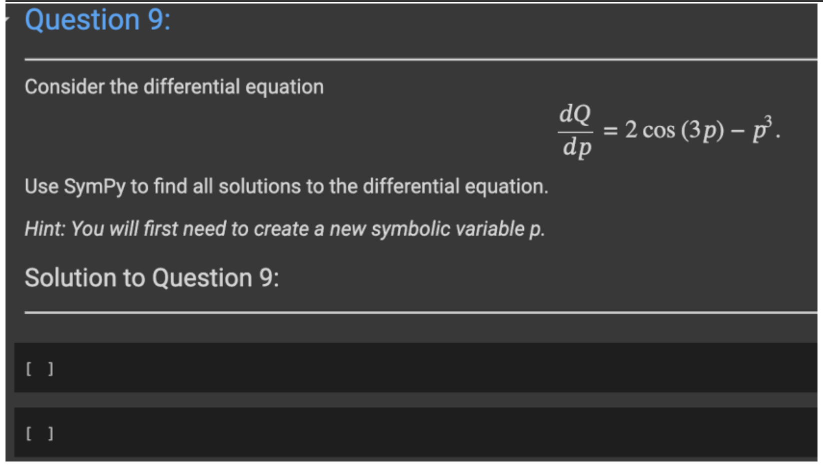
We show some examples below. Feel free to experiment!

```
[ ] t, k = sym.symbols('t, k')  # Creating t and k as symbols
```

```
[ ] # Note that indefinite integrals do not include the constant of integration

    sym.integrate(t**2 + k, t)  # Determines an antiderivative of the function $t^2+k$ with respect to t
```

```
[ ] sym.integrate(t**2 + k, k)  # Integrates with respect to k
```

```
[ ] sym.integrate(t**2 + k, (t, 0, 1))  # A definite integral, with respect to t, from 0 to 1
```



Question 9:

Consider the differential equation

$$\frac{dQ}{dp} = 2\cos(3p) - p^3.$$

Use SymPy to find all solutions to the differential equation.

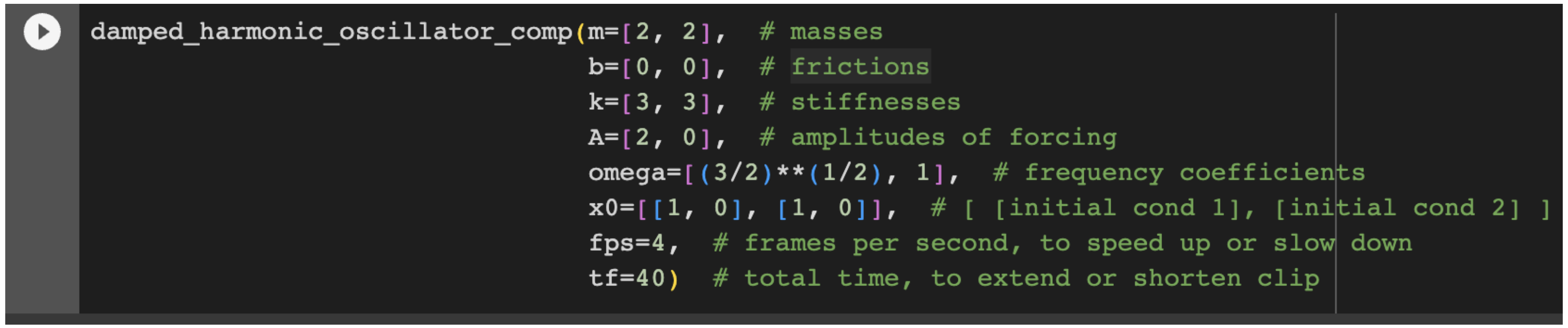
Hint: You will first need to create a new symbolic variable p.

Solution to Question 9:

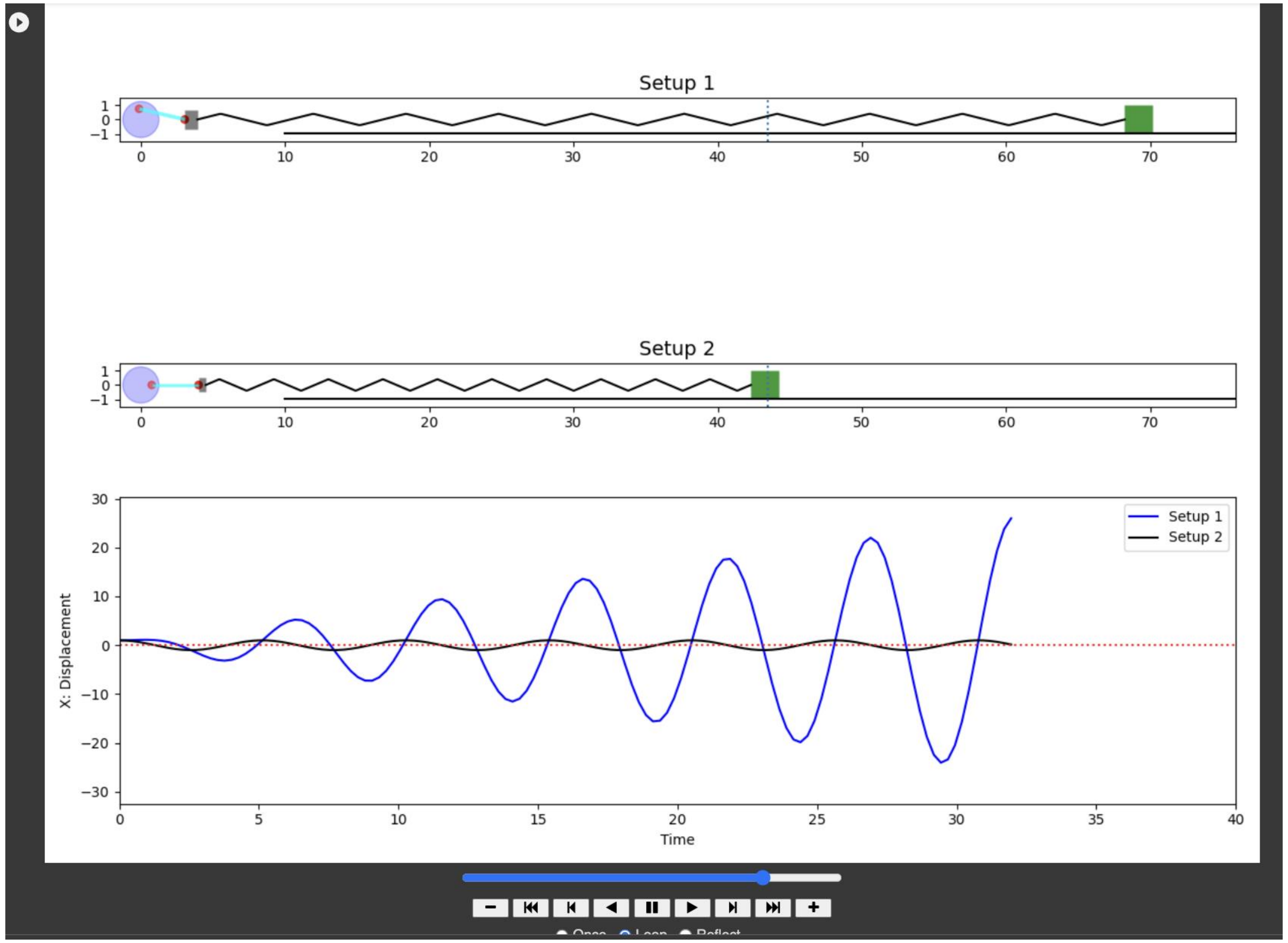
```
[ ]
```

```
[ ]
```

Using Code Cells to Solve a First Differential Equation

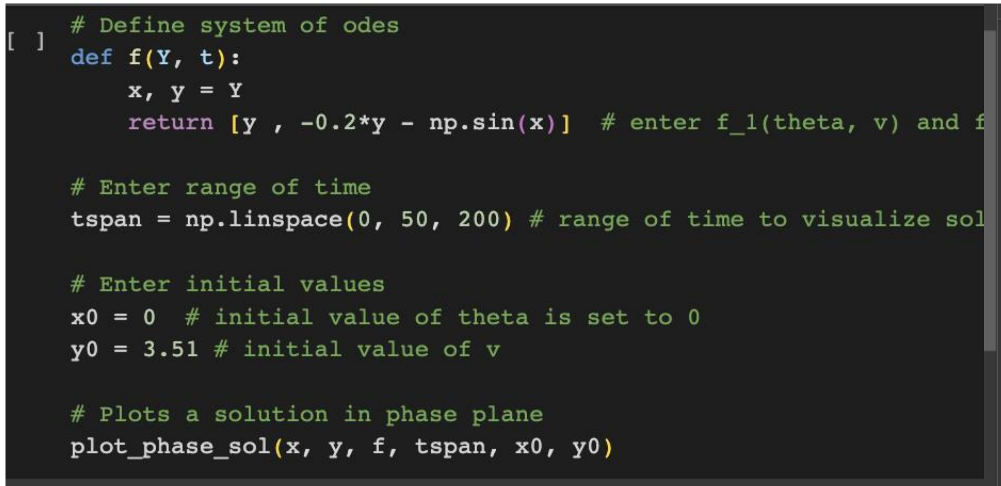


```
damped_harmonic_oscillator_comp(m=[2, 2],  # masses
                                b=[0, 0],  # frictions
                                k=[3, 3],  # stiffnesses
                                A=[2, 0],  # amplitudes of forcing
                                omega=[(3/2)**(1/2), 1],  # frequency coefficients
                                x0=[[1, 0], [1, 0]],  # [ [initial cond 1], [initial cond 2] ]
                                fps=4,  # frames per second, to speed up or slow down
                                tf=40)  # total time, to extend or shorten clip
```

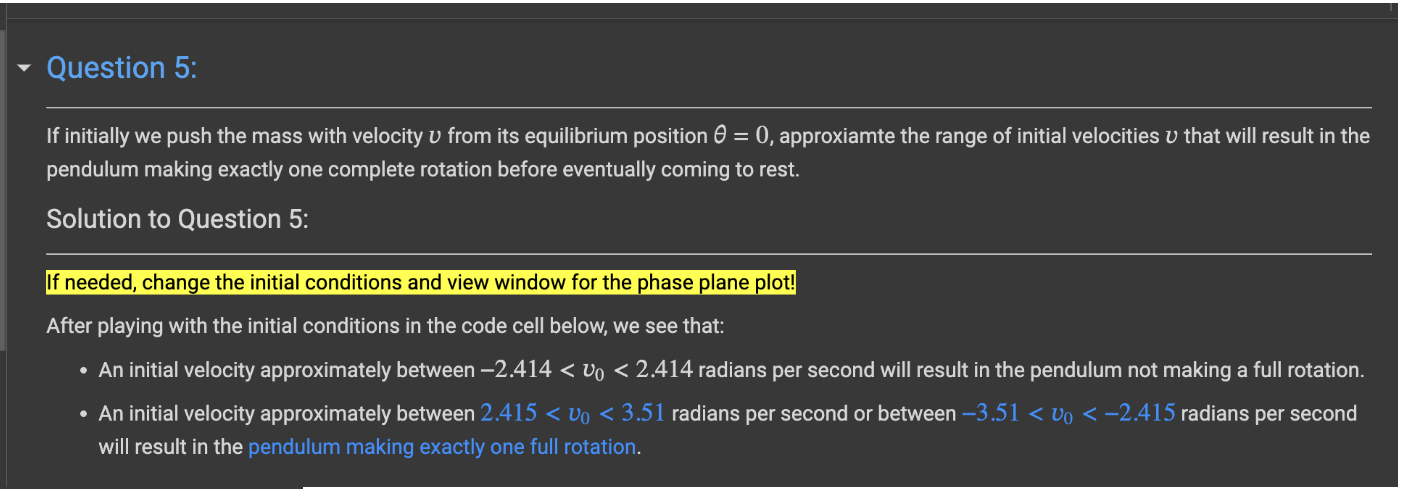


Python Modules for Experimenting with Differential Equations

Interactive Experiments with Harmonic Oscillators



```
[ ]  # Define system of odes
     def f(Y, t):
         x, y = Y
         return [y , -0.2*y - np.sin(x)]  # enter f_1(theta, v) and f

     # Enter range of time
     tspan = np.linspace(0, 50, 200) # range of time to visualize sol

     # Enter initial values
     x0 = 0  # initial value of theta is set to 0
     y0 = 3.51 # initial value of v

     # Plots a solution in phase plane
     plot_phase_sol(x, y, f, tspan, x0, y0)
```



Question 5:

If initially we push the mass with velocity v from its equilibrium position $\theta = 0$, approxiamte the range of initial velocities v that will result in the pendulum making exactly one complete rotation before eventually coming to rest.

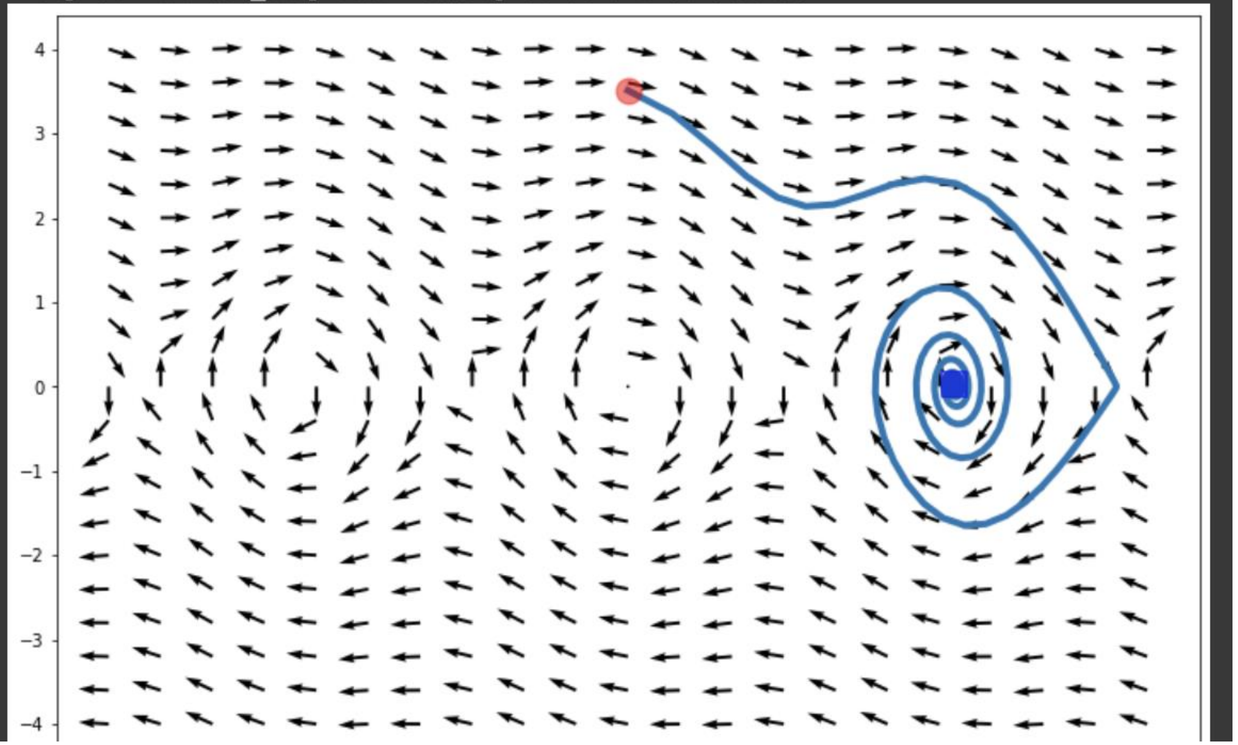
Solution to Question 5:

If needed, change the initial conditions and view window for the phase plane plot!

After playing with the initial conditions in the code cell below, we see that:

- An initial velocity approximately between $-2.414 < v_0 < 2.414$ radians per second will result in the pendulum not making a full rotation.
- An initial velocity approximately between $2.415 < v_0 < 3.51$ radians per second or between $-3.51 < v_0 < -2.415$ radians per second will result in the pendulum making exactly one full rotation.

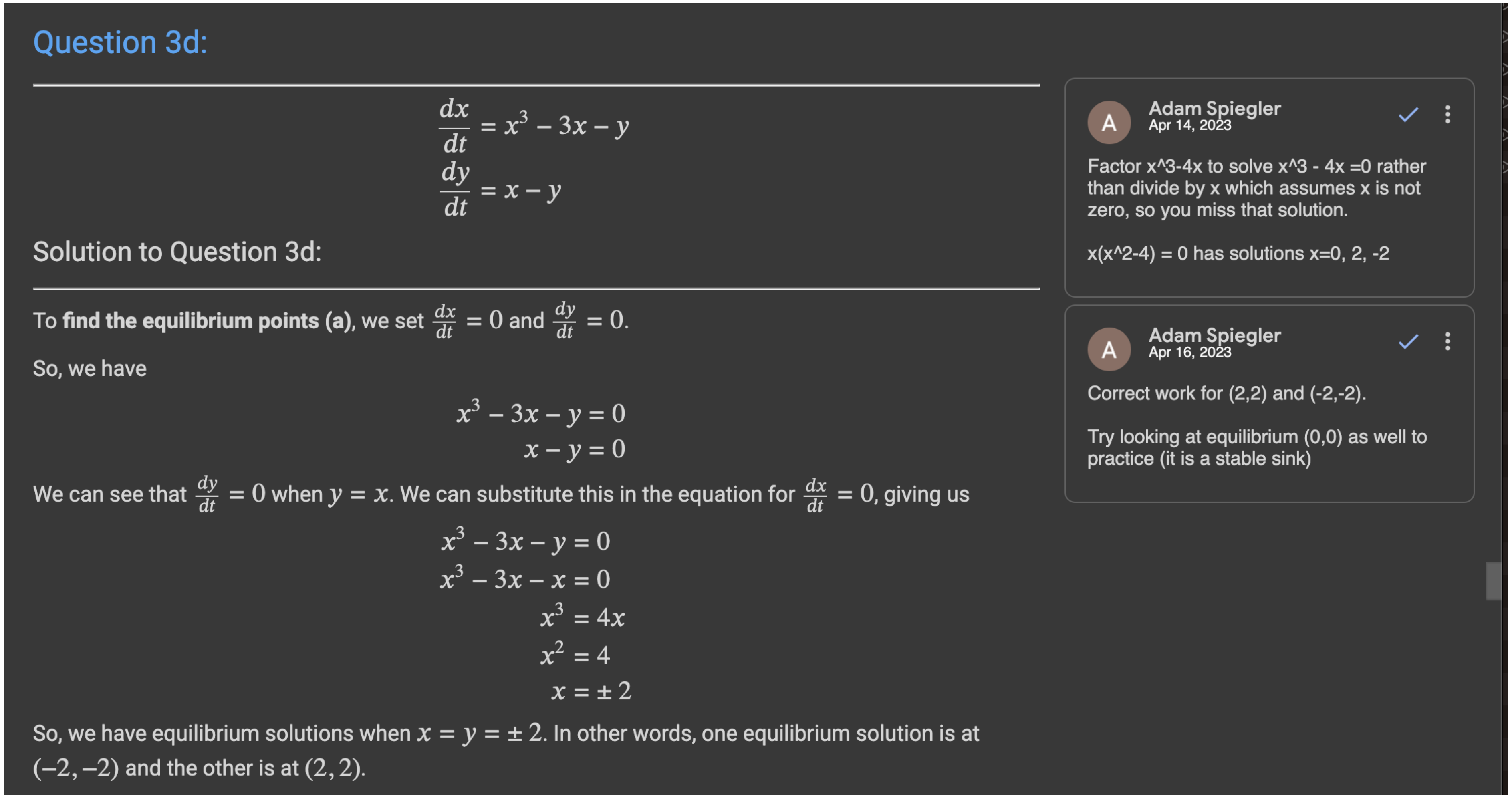
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8613b0eb50>
```



Python Modules for Experimenting with Differential Equations

Phase Plane Plots in Colab

Sample of Student Homework



Question 3d:

$$\frac{dx}{dt} = x^3 - 3x - y$$
$$\frac{dy}{dt} = x - y$$

Solution to Question 3d:

To **find the equilibrium points (a)**, we set $\frac{dx}{dt} = 0$ and $\frac{dy}{dt} = 0$.

So, we have

$$x^3 - 3x - y = 0$$
$$x - y = 0$$

We can see that $\frac{dy}{dt} = 0$ when $y = x$. We can substitute this in the equation for $\frac{dx}{dt} = 0$, giving us

$$x^3 - 3x - y = 0$$
$$x^3 - 3x - x = 0$$
$$x^3 = 4x$$
$$x^2 = 4$$
$$x = \pm 2$$

So, we have equilibrium solutions when $x = y = \pm 2$. In other words, one equilibrium solution is at $(-2, -2)$ and the other is at $(2, 2)$.

Adam Spiegler
Apr 14, 2023

Factor x^3-4x to solve x^3 - 4x =0 rather than divide by x which assumes x is not zero, so you miss that solution.

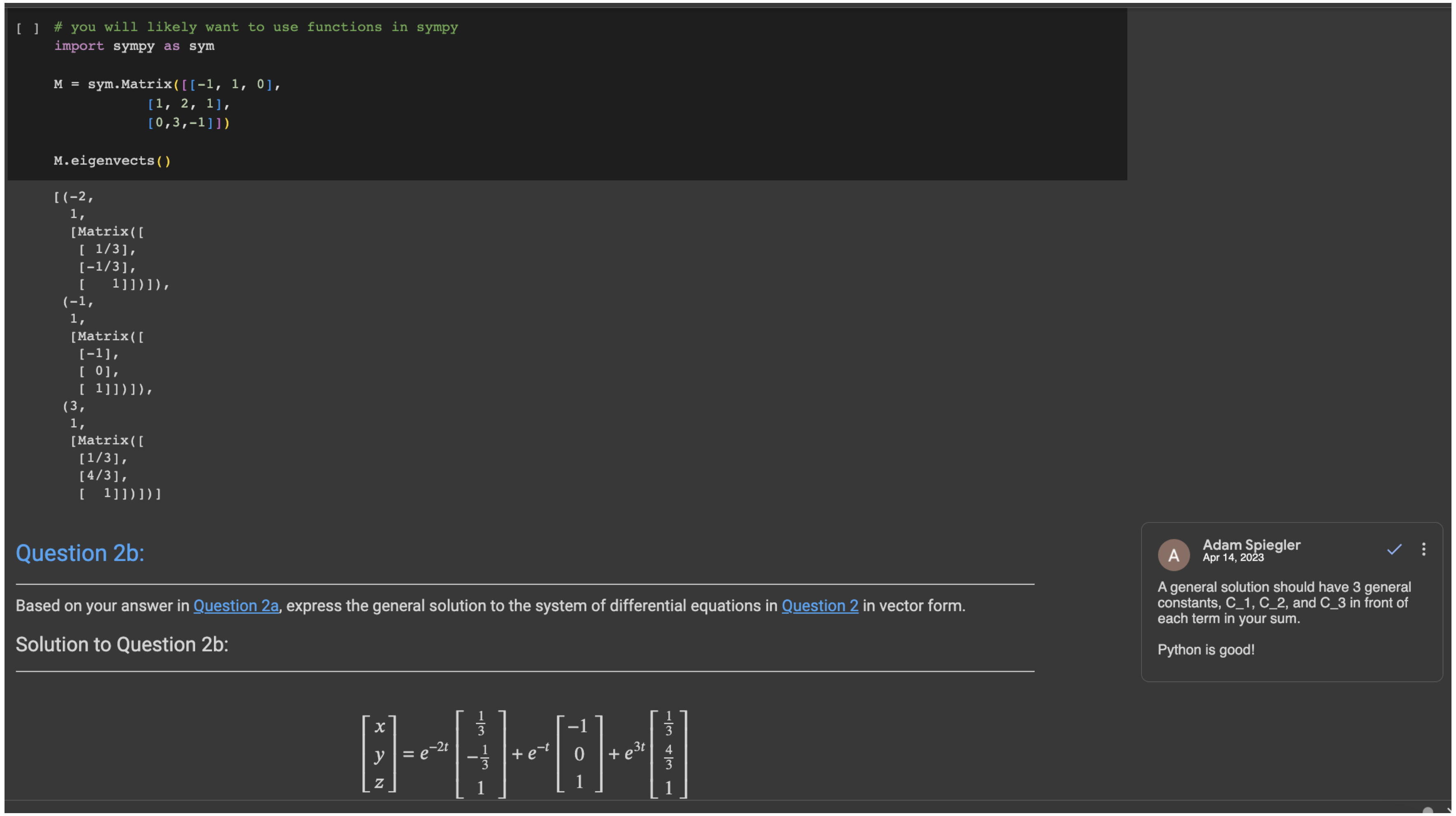
x(x^2-4) = 0 has solutions x=0, 2, -2

Adam Spiegler
Apr 16, 2023

Correct work for (2,2) and (-2,-2).

Try looking at equilibrium (0,0) as well to practice (it is a stable sink)

Sample of Student Homework



```
[ ] # you will likely want to use functions in sympy
    import sympy as sym

    M = sym.Matrix([[-1, 1, 0],
                    [1, 2, 1],
                    [0,3,-1]])

    M.eigenvects()
```

```
[(-2,
  1,
  [Matrix([
   [ 1/3],
   [-1/3],
   [   1]])]),
 (-1,
  1,
  [Matrix([
   [-1],
   [ 0],
   [ 1]])]),
 (3,
  1,
  [Matrix([
   [1/3],
   [4/3],
   [  1]])])]
```

Question 2b:

Based on your answer in Question 2a, express the general solution to the system of differential equations in Question 2 in vector form.

Solution to Question 2b:

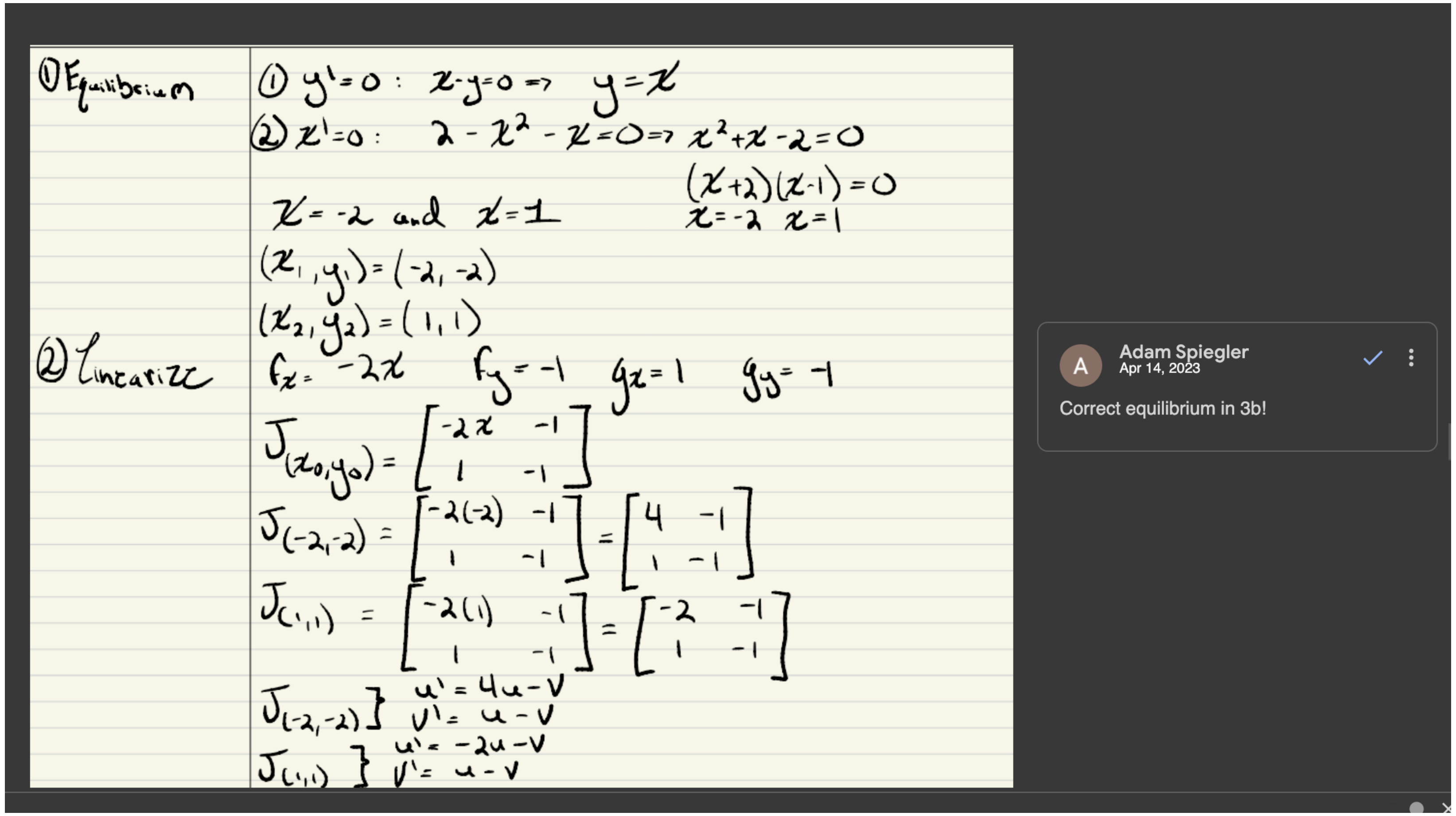
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = e^{-2t}\begin{bmatrix} \frac{1}{3} \\ -\frac{1}{3} \\ 1 \end{bmatrix} + e^{-t}\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} + e^{3t}\begin{bmatrix} \frac{1}{3} \\ \frac{4}{3} \\ 1 \end{bmatrix}$$

Adam Spiegler
Apr 14, 2023

A general solution should have 3 general constants, C_1, C_2, and C_3 in front of each term in your sum.

Python is good!

Sample of Student Homework



① Equilibrium

① $y'=0$: $x-y=0 \Rightarrow y=x$

② $x'=0$: $2-x^2-x=0 \Rightarrow x^2+x-2=0$

$(x+2)(x-1)=0$

$x=-2$ $x=1$

$x=-2$ and $x=1$

$(x_1,y_1)=(-2,-2)$

$(x_2,y_2)=(1,1)$

② Linearize

$f_x=-2x$ $f_y=-1$ $g_x=1$ $g_y=-1$

$$J_{(x_0,y_0)}=\begin{bmatrix}-2x & -1\\ 1 & -1\end{bmatrix}$$

$$J_{(-2,-2)}=\begin{bmatrix}-2(-2) & -1\\ 1 & -1\end{bmatrix}=\begin{bmatrix}4 & -1\\ 1 & -1\end{bmatrix}$$

$$J_{(1,1)}=\begin{bmatrix}-2(1) & -1\\ 1 & -1\end{bmatrix}=\begin{bmatrix}-2 & -1\\ 1 & -1\end{bmatrix}$$

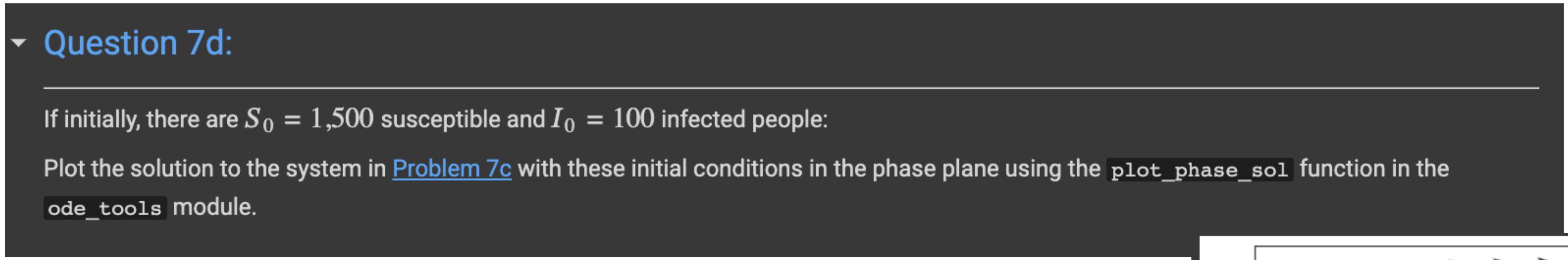
$$J_{(-2,-2)}\left.\right\}\begin{matrix}u'=4u-v\\ v'=u-v\end{matrix}$$

$$J_{(1,1)}\left.\right\}\begin{matrix}u'=-2u-v\\ v'=u-v\end{matrix}$$

Adam Spiegler
Apr 14, 2023

Correct equilibrium in 3b!

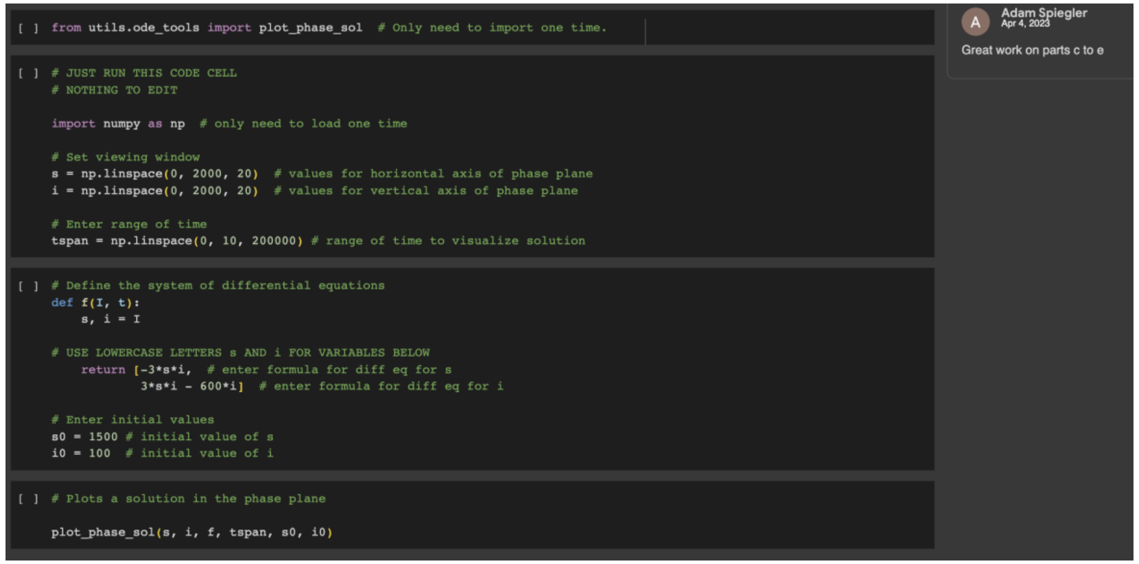
Sample of Student Homework



Question 7d:

If initially, there are $S_0 = 1{,}500$ susceptible and $I_0 = 100$ infected people:

Plot the solution to the system in Problem 7c with these initial conditions in the phase plane using the `plot_phase_sol` function in the `ode_tools` module.



```
[ ] from utils.ode_tools import plot_phase_sol  # Only need to import one time.
```

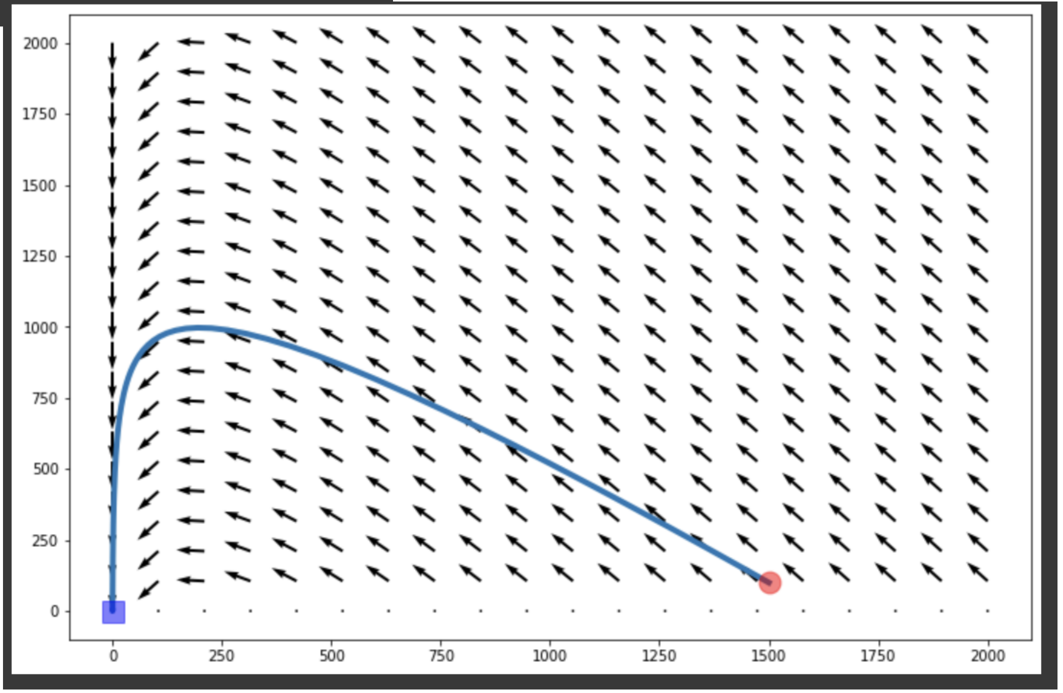
```
[ ] # JUST RUN THIS CODE CELL
    # NOTHING TO EDIT

    import numpy as np  # only need to load one time

    # Set viewing window
    s = np.linspace(0, 2000, 20)  # values for horizontal axis of phase plane
    i = np.linspace(0, 2000, 20)  # values for vertical axis of phase plane

    # Enter range of time
    tspan = np.linspace(0, 10, 200000) # range of time to visualize solution
```

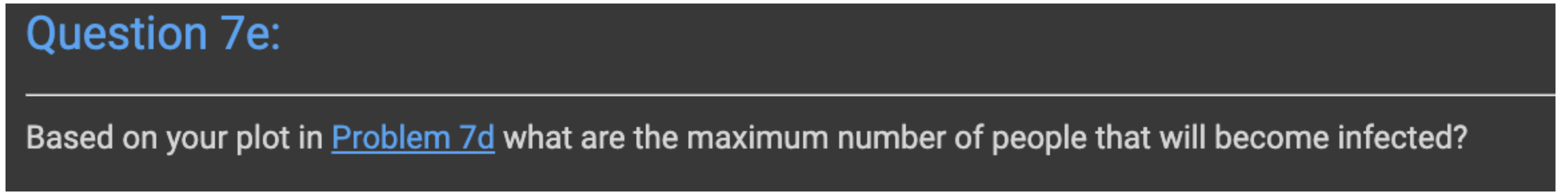
```
[ ] # Define the system of differential equations
    def f(I, t):
        s, i = I

    # USE LOWERCASE LETTERS s AND i FOR VARIABLES BELOW
        return [-3*s*i,  # enter formula for diff eq for s
                3*s*i - 600*i]  # enter formula for diff eq for i

    # Enter initial values
    s0 = 1500 # initial value of s
    i0 = 100  # initial value of i
```

```
[ ] # Plots a solution in the phase plane

    plot_phase_sol(s, i, f, tspan, s0, i0)
```

Adam Spiegler
Apr 4, 2023

Great work on parts c to e





Question 7e:

Based on your plot in Problem 7d what are the maximum number of people that will become infected?

Student Feedback

The integration of python was a really great learning experience. I also really appreciate the use of the OER, because there weren't any barriers to finding good information.

I think the CoLab notebooks were the most effective part. They were organized and they were easy to access.

I didn't think I could ever understand how to do a differential equation. The notes are very useful and helpful to understand the concepts and use real world applications that aren't cheesy or dumb.

I really appreciated the python notebooks, it helped give some real world applications to our problems and I also enjoyed learning more about coding in relation to diff eqs.

Coding is becoming more and more important and is much easier when every class includes it. Colab seems to be the way of the future. I prefer every course ask for some coding project,

Coding is cool. I think more coding would be beneficial. Definitely plan to build upon coding skills in the future.

"Which mathematics course(s) do you feel were the most useful with respect to preparing you for a career and/or graduate school, and why?"

9 out of 18 responses in Spring 2023 responded Math Clinic (more than any other course).

1 out of 18 responses in Spring 2023 mentioned Math Clinic as the least useful.

Reflections

- Introduce Python to students in the first week.
- Offer alternatives to Colab for work in class.
- Let the students come to the programming on their own terms rather than force it.
- Give students the opportunity to decide how to use Python.
  - More and more students opted to type up solutions in Colab.
- Students were motivated to put in the work when they understand “why”.

Thank You!

- Jonathan Hirschi and Troy Butler for help with Python modules.
- IODE (https://iode.sdsu.edu/) team: Chris Rasmussen, Karen Keene, Justin Dunmyre, and Nicholas Fortune
- Colorado Department of Higher Education and the Colorado OER Grant Program.

COLORADO
Department of Higher Education

Contact: Adam.spiegler@ucdenver.edu

HTML text: https://aspiegler.github.io/Exploring-Differential-Equations/

GitHub: https://github.com/CU-Denver-MathStats-OER/ODEs