Chaos

PHYS 246 class 5

https://lkwagner.github.io/IntroductionToComputationalPhysics/intro.html

```
from google.colab import drive
drive.mount('/content/drive')
!cp /content/drive/MyDrive/Colab\ Notebooks/Dynamics.ipynb ./
!jupyter nbconvert --to HTML "Dynamics.ipynb"
```

Announcements/notes

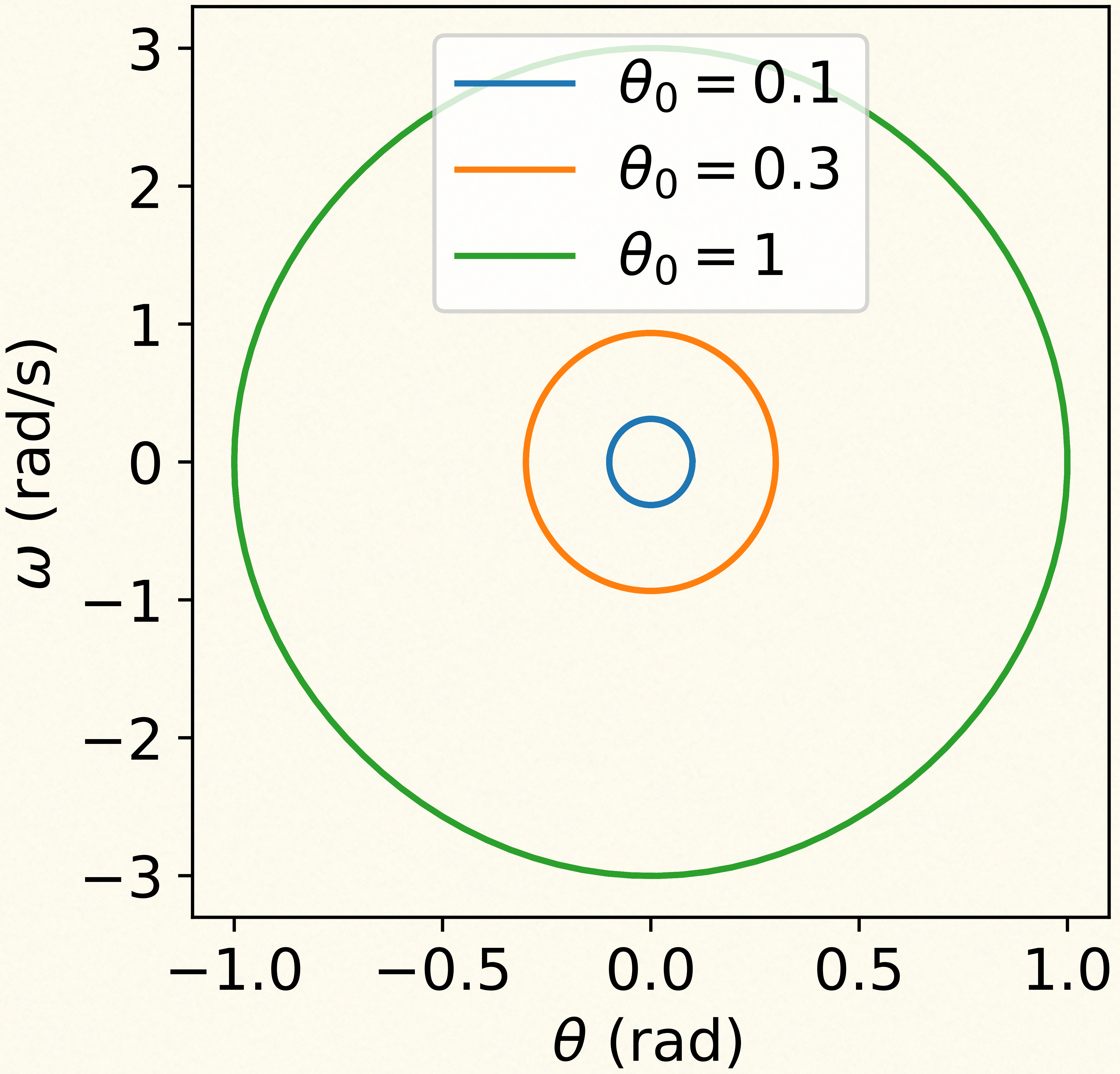
- 'Exoplanets' is due tonight.
- Please do make sure you have set permissions so we can see the ipynb file. You can also just upload it if that's easier.
- Labels and units are required for graphs (graphs are not complete unless they have this!) You will start to lose points for this!
- Some of you are still having trouble with PDFs (you will have gotten notes about this on your homework). We are now starting to remove points if this is not correct.

Summary

Going to consider 3 situations:

- Just a pendulum
- A driven pendulum with damping
- A double pendulum

Pendulum



x->theta (rad),
v -> omega (rad/s)

update rules:

$$\theta(t+\Delta t) \simeq \theta(t) + \omega \Delta t$$

$$\omega(t+\Delta t) = \omega(t) + \frac{d\omega}{dt}\Delta t$$

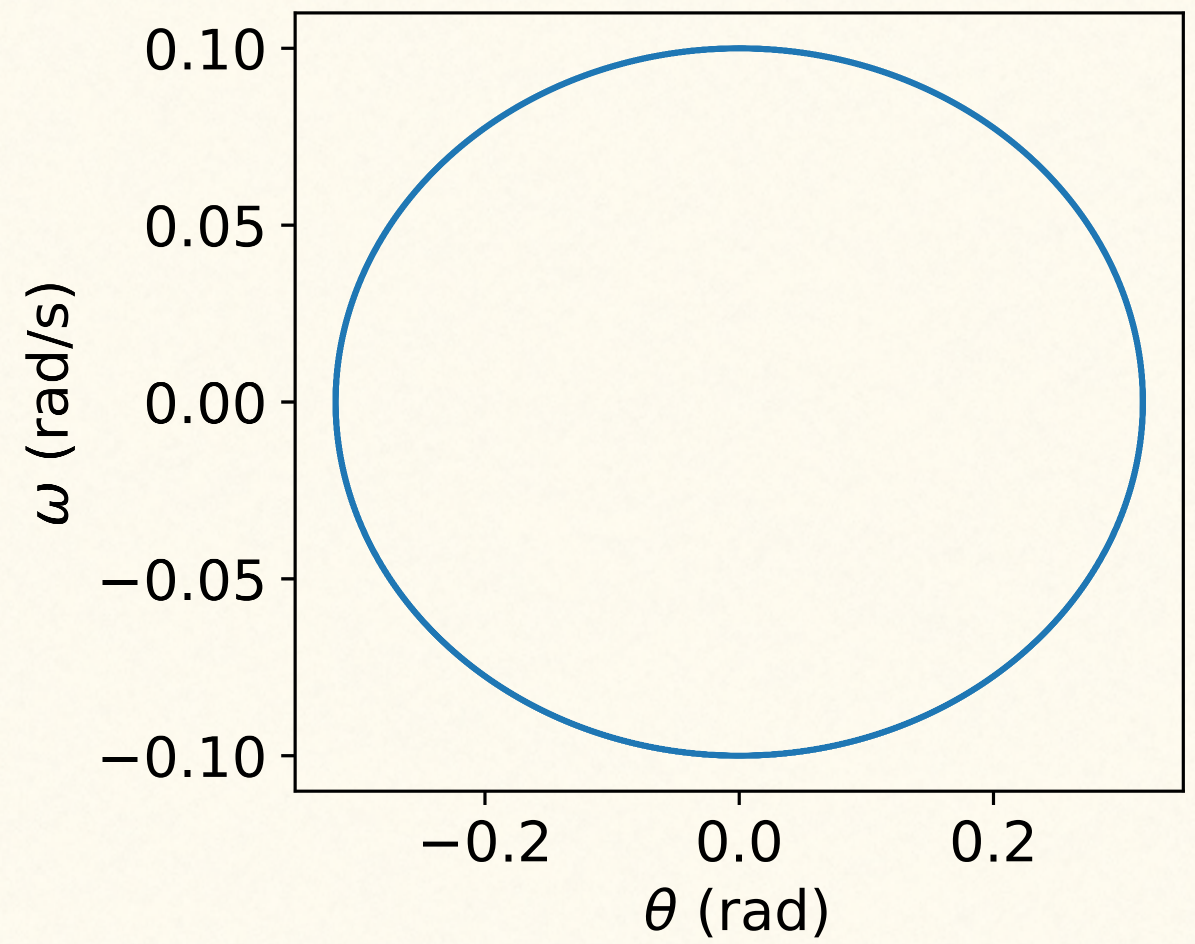
$$\frac{d\omega}{dt} = \frac{-g\sin\theta}{L}$$

note: the analytic results we have are ONLY for small angles! In this case we trust the computation more than the analytic parts.

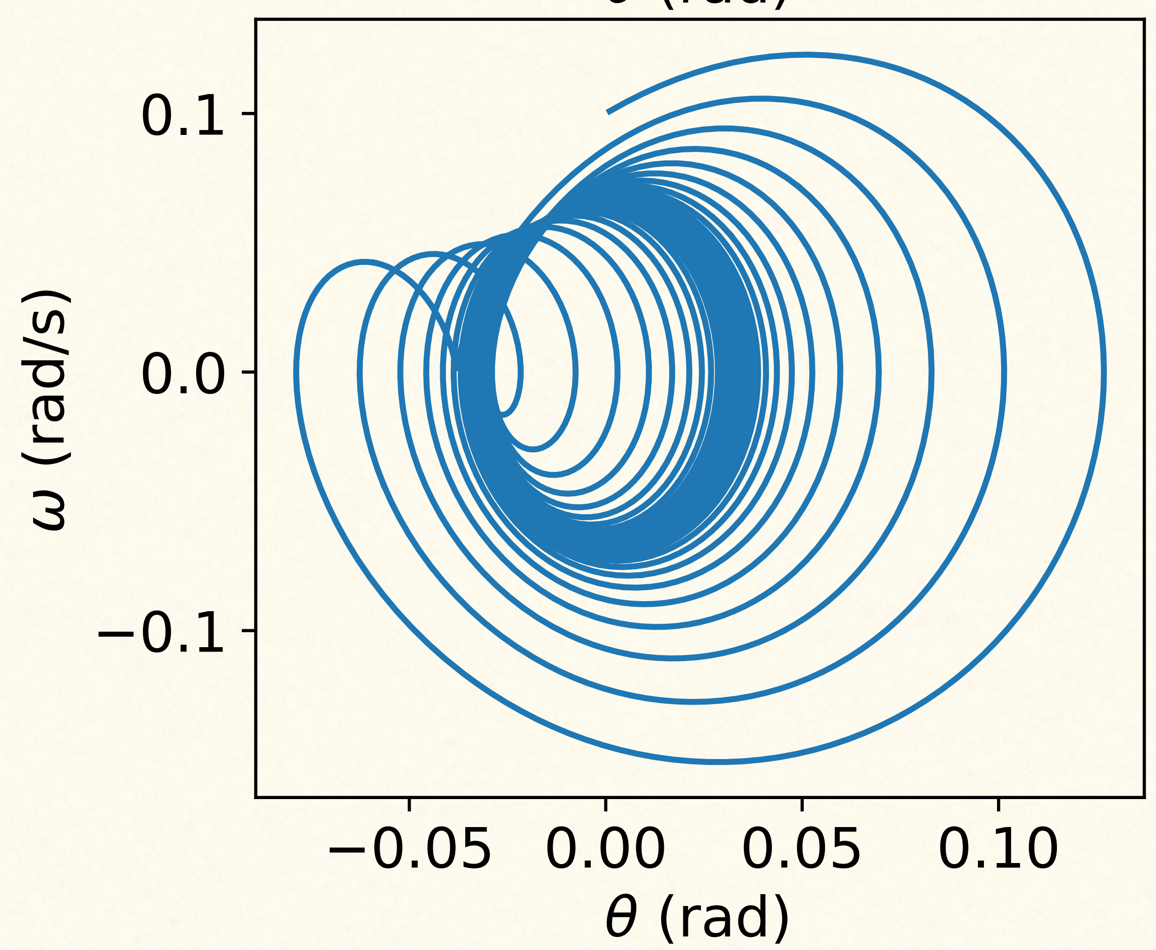
Driven pendulum

$$\frac{d\omega}{dt} = -A\omega - B\sin\theta + C\sin\Omega_{ext}t$$

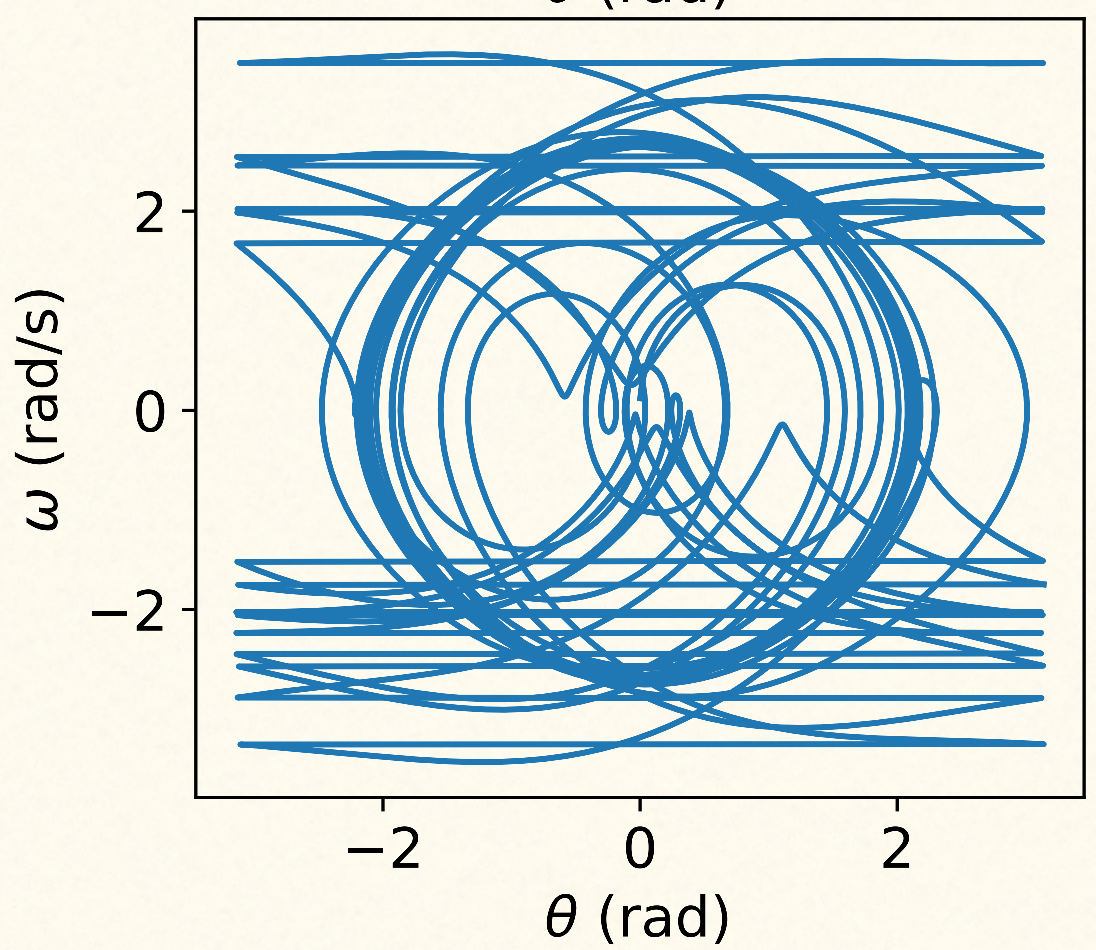
No drive,
no damping



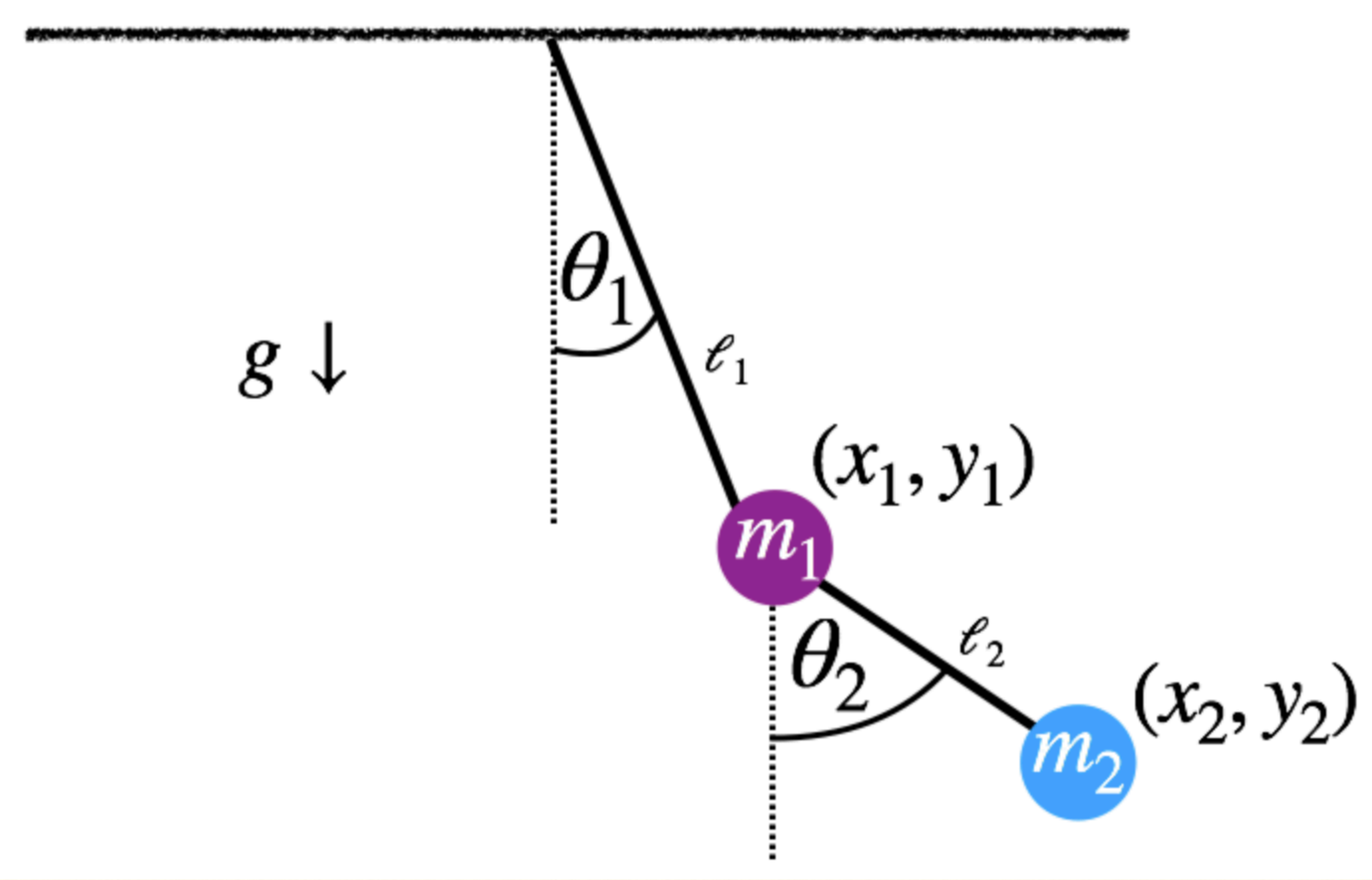
Overdamped



Underdamped



Double pendulum



A bit difficult to work out forces!
Easier to use Euler-Lagrange method (PHYS 325) (principle of least action)

$$\alpha_1(\theta_1,\theta_2) := \frac{l_2}{l_1}\left(\frac{m_2}{m_1+m_2}\right)\cos(\theta_1-\theta_2)$$

$$\alpha_2(\theta_1,\theta_2) := \frac{l_1}{l_2}\cos(\theta_1-\theta_2)$$

$$f_1(\theta_1,\theta_2,\dot\theta_1,\dot\theta_2) := -\frac{l_2}{l_1}\left(\frac{m_2}{m_1+m_2}\right)\dot\theta_2^2\sin(\theta_1-\theta_2) - \frac{g}{l_1}\sin\theta_1$$

$$f_2(\theta_1,\theta_2,\dot\theta_1,\dot\theta_2) := \frac{l_1}{l_2}\dot\theta_1^2\sin(\theta_1-\theta_2) - \frac{g}{l_2}\sin\theta_2$$

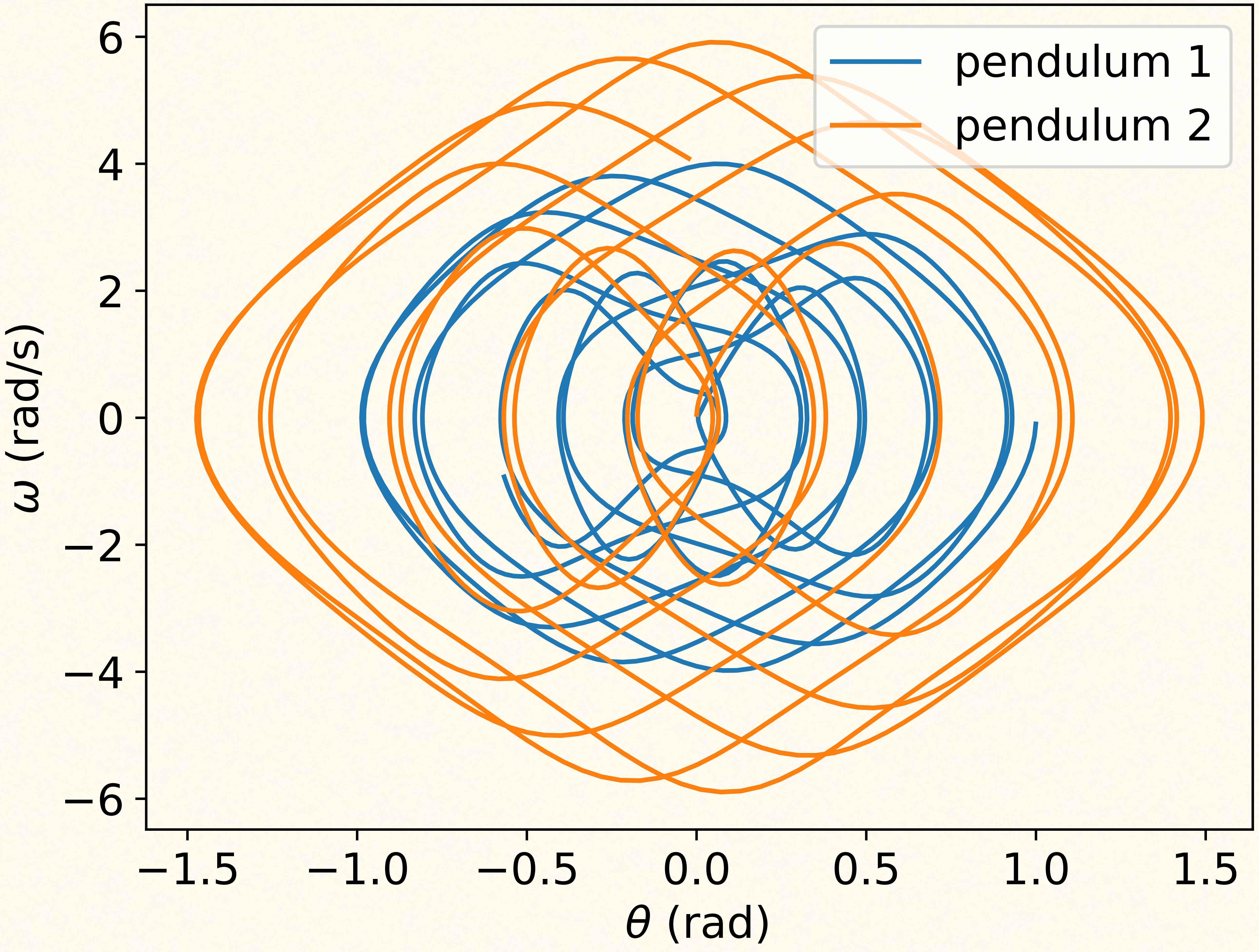
$$g_1 := \frac{f_1-\alpha_1 f_2}{1-\alpha_1\alpha_2}$$

$$g_2 := \frac{-\alpha_2 f_1+f_2}{1-\alpha_1\alpha_2}$$

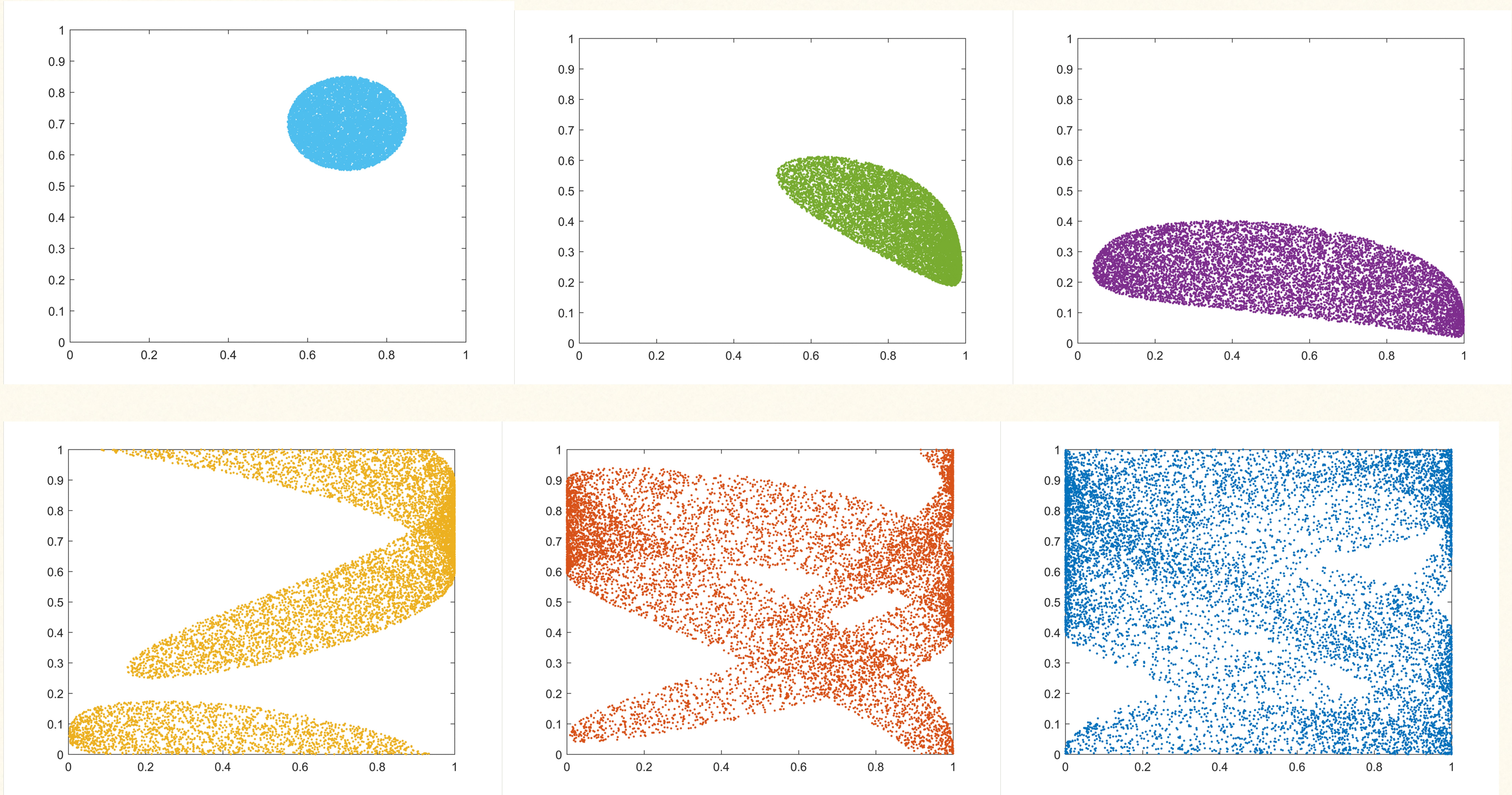
$$\frac{d}{dt}\begin{pmatrix}\theta_1\\ \theta_2\\ \omega_1\\ \omega_2\end{pmatrix} = \begin{pmatrix}\omega_1\\ \omega_2\\ g_1(\theta_1,\theta_2,\omega_1,\omega_2)\\ g_2(\theta_1,\theta_2,\omega_1,\omega_2)\end{pmatrix}$$

https://dassencio.org/33

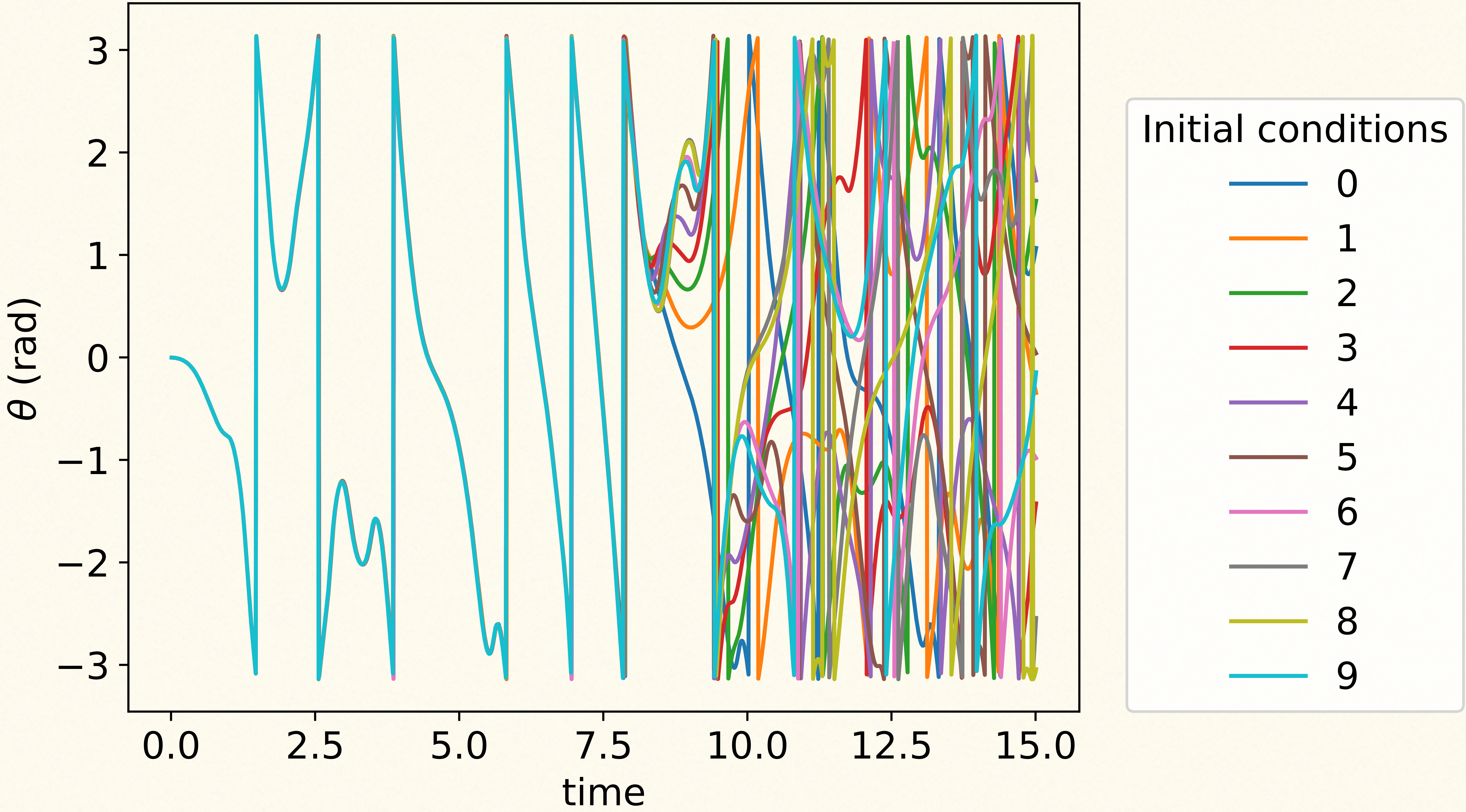
Double pendulum phase space



Chaos -- general concept



Chaos



θ (rad)
3
2
1
0
−1
−2
−3
0.0
2.5
5.0
7.5
10.0
12.5
15.0
time
Initial conditions
0
1
2
3
4
5
6
7
8
9

Checks for your solutions

Undamped, undriven pendulum:

•Small angles should match analytic result

•Energy should be conserved (phase space should be an ellipse)

Driven pendulum

Large driving force should make it unstable.

Small driving force plus damping --> should go to (0,0)

Double pendulum

Small initial angles/velocities should

Python stuff

animate is meant to visualize multiple runs

```python
def animateMe_single_pendulum(positions, params):
    """
    positions [run, time, theta]
    params {'l1': length of the pendulum}
    """
```

Magically wrap the positions to (-pi,pi)

```python
    pos = pos - 2*np.pi*np.ceil((pos-np.pi)/(2*np.pi))
```

Why I don't recommend global variables

```python
[7]
a=5
✓ 0.0s
```

More...

```python
[6]
def f(x):
    return x * a

print(f(2))
✓ 0.0s
... 14
```

```python
[5]
a=7
✓ 0.0s
```

```python
[8]
print(f(2))
✓ 0.0s
... 10
```