PHYS280: HW8

Kavish Senthilkumar

Nonlinear oscillations: nondimensionalization

a

We have the below equations

$$\frac{d(\Delta m)}{dt} = -Q\theta/\sqrt{1+\theta^2}$$

$$\frac{d\theta}{dt} = (\Delta m)\gamma\cos\theta$$

We nondimensionalize with $\bar{t} = \frac{\sqrt{Q\gamma}}{2\pi}t$ and $\Delta\bar{m} = \alpha\Delta m$, where α is some quantity with units kg^{-1} we will use to rescale Δm.

We choose $\alpha = \frac{Q}{2\pi\sqrt{Q\gamma}} = \frac{1}{2\pi\sqrt{Q/\gamma}}$, seeing that since Q has units kg/s and γ has units $(kg\cdot s)^{-1}$, α now has units $(\sqrt{kg/s\cdot kg\cdot s})^{-1} = kg^{-1}$ as desired. We add the 2π to allow easy cancelling in the following step, noting 2π is unitless.

Substituting with the chain rule,

$$\frac{d(\Delta m)}{dt} = \frac{\sqrt{Q\gamma}}{2\pi\alpha}\frac{d(\Delta\bar{m})}{d\bar{t}} = \frac{\sqrt{Q\gamma}2\pi\sqrt{Q/\gamma}}{2\pi}\frac{d(\Delta\bar{m})}{d\bar{t}} = Q\frac{d(\Delta\bar{m})}{d\bar{t}}$$

$$Q\frac{d(\Delta\bar{m})}{d\bar{t}} = -Q\theta/\sqrt{1+\theta^2} \implies \boxed{\frac{d(\Delta\bar{m})}{d\bar{t}} = -\theta/\sqrt{1+\theta^2}}$$

We apply a similar process to $\frac{d\theta}{dt}$, replacing $\bar{m}$ with $\Delta\bar{m}$. Using the chain rule,

$$\frac{d\theta}{d\bar{t}}\frac{\sqrt{Q\gamma}}{2\pi} = 2\pi\sqrt{Q/\gamma}\Delta\bar{m}\gamma\cos\theta$$

$$\frac{d\theta}{d\bar{t}} = \frac{\sqrt{Q/\gamma}}{\sqrt{Q\gamma}}\Delta\bar{m}\gamma\cos\theta$$

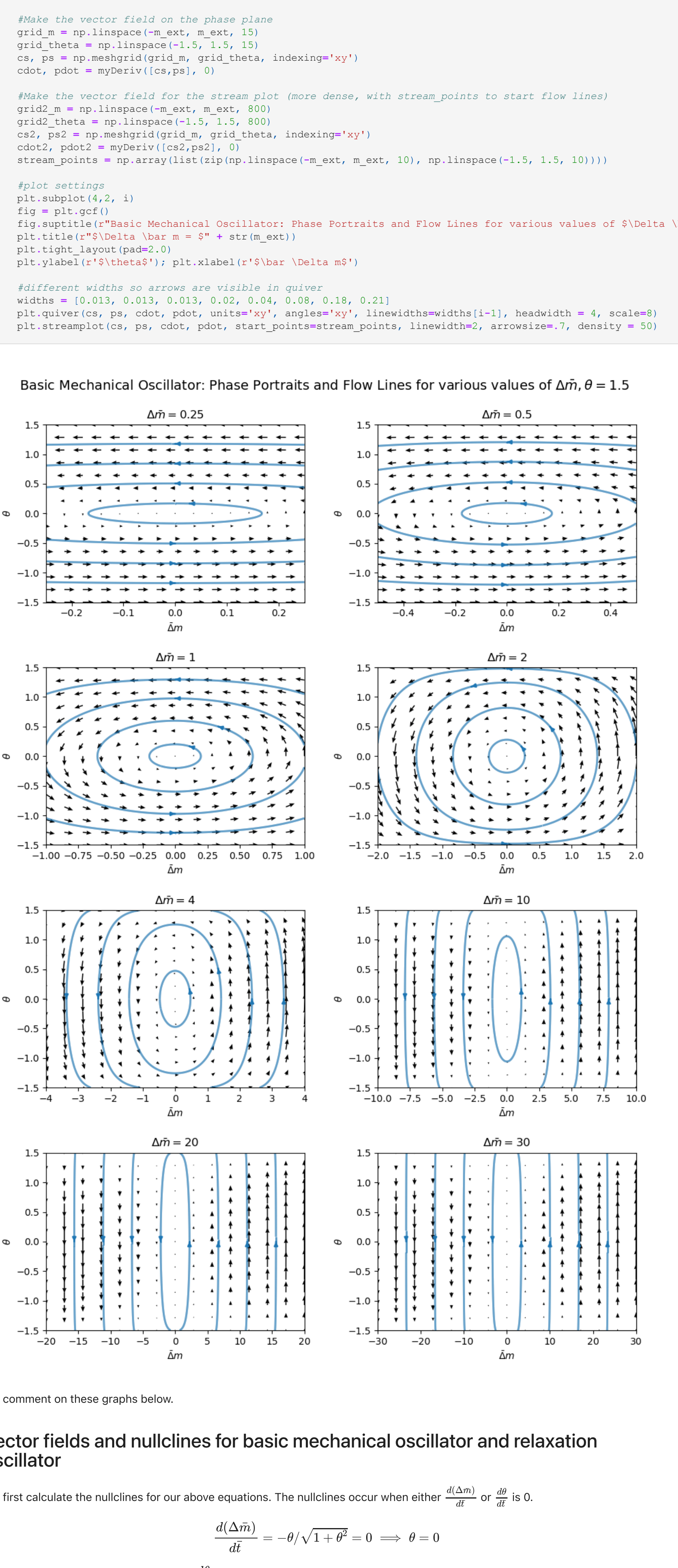
$$\frac{d\theta}{d\bar{t}} = (\frac{\sqrt{1/\gamma}}{\sqrt{\gamma}}\gamma)\Delta\bar{m}\cos\theta$$

$$\boxed{\frac{d\theta}{d\bar{t}} = \Delta\bar{m}\cos\theta}$$

Below, we graph the phase portrait and some typical streamlines

In [150…

```
%matplotlib notebook

import numpy as np; import matplotlib.pyplot as plt


#calculates derivative
def myDeriv(input, t):
    m=input[0]
    theta=input[1]
    dmdt=-theta/(np.sqrt(1+theta**2))
    dthetadt=m*np.cos(theta)
    return [dmdt, dthetadt]

#plot for various values of amplitudes
amplitudes = [0.25, 0.5, 1, 2, 4, 10, 20, 30]
plt.rcParams['figure.figsize'] = [10, 15]
for i in range(1,9):
    #Get amplitide
    m_ext = amplitudes[i-1]

    #Make the vector field on the phase plane
    grid_m = np.linspace(-m_ext, m_ext, 15)
    grid_theta = np.linspace(-1.5, 1.5, 15)
    cs, ps = np.meshgrid(grid_m, grid_theta, indexing='xy')
    cdot, pdot = myDeriv([cs,ps], 0)

    #Make the vector field for the stream plot (more dense, with stream_points to start flow lines)
    grid2_m = np.linspace(-m_ext, m_ext, 800)
    grid2_theta = np.linspace(-1.5, 1.5, 800)
    cs2, ps2 = np.meshgrid(grid_m, grid_theta, indexing='xy')
    cdot2, pdot2 = myDeriv([cs2,ps2], 0)
    stream_points = np.array(list(zip(np.linspace(-m_ext, m_ext, 10), np.linspace(-1.5, 1.5, 10))))

    #plot settings
    plt.subplot(4,2, i)
    fig = plt.gcf()
    fig.suptitle(r"Basic Mechanical Oscillator: Phase Portraits and Flow Lines for various values of $\Delta \b
    plt.title(r"$\Delta \bar m = $" + str(m_ext))
    plt.tight_layout(pad=2.0)
    plt.ylabel(r'$\theta$'); plt.xlabel(r'$\bar \Delta m$')

    #different widths so arrows are visible in quiver
    widths = [0.013, 0.013, 0.013, 0.02, 0.04, 0.08, 0.18, 0.21]
    plt.quiver(cs, ps, cdot, pdot, units='xy', angles='xy', linewidths=widths[i-1], headwidth = 4, scale=8)
    plt.streamplot(cs, ps, cdot, pdot, start_points=stream_points, linewidth=2, arrowsize=.7, density = 50)
```



We comment on these graphs below.

Vector fields and nullclines for basic mechanical oscillator and relaxation oscillator

We first calculate the nullclines for our above equations. The nullclines occur when either $\frac{d(\Delta\bar{m})}{d\bar{t}}$ or $\frac{d\theta}{d\bar{t}}$ is 0.

$$\frac{d(\Delta\bar{m})}{d\bar{t}} = -\theta/\sqrt{1+\theta^2} = 0 \implies \theta = 0$$

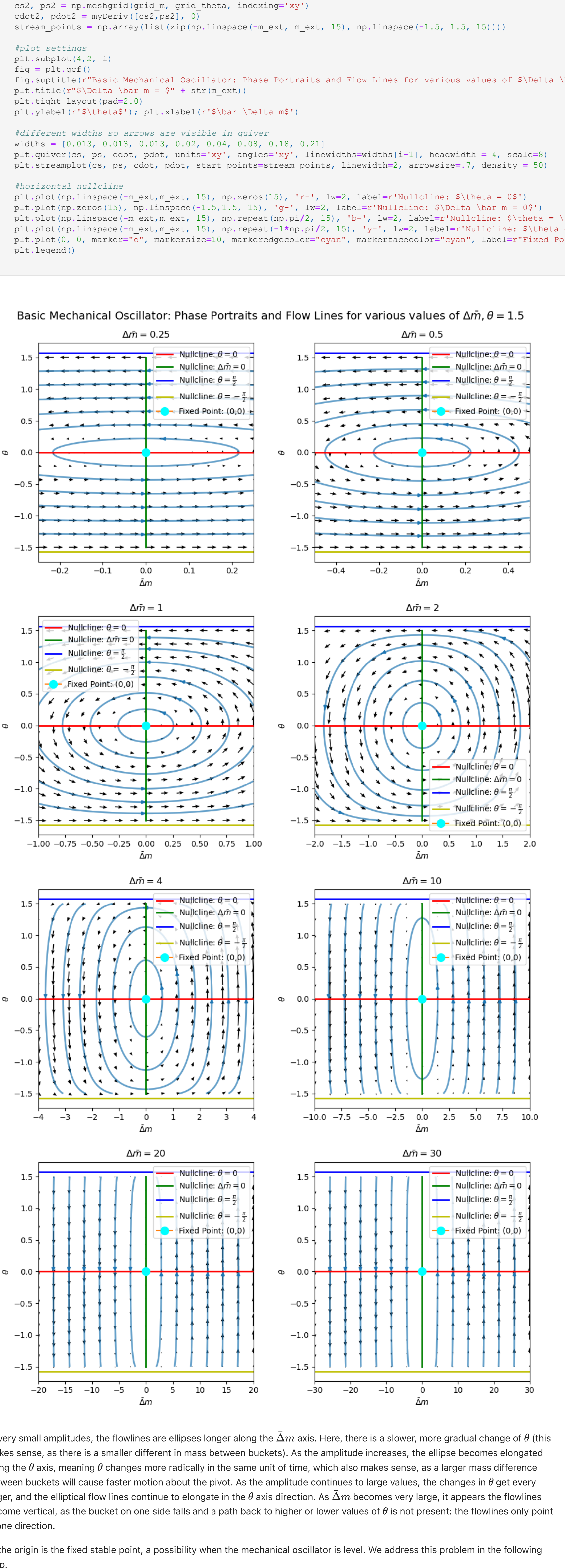
$$\frac{d\theta}{d\bar{t}} = \Delta\bar{m}\cos\theta = 0 \implies \Delta\bar{m} = 0 \; or \; \theta = \pm\frac{\pi}{2}$$

Our nullclines are $\theta = 0 and \pm\frac{\pi}{2}, \Delta\bar{m} = 0$. We draw them below.

a

In [151…

```
%matplotlib notebook

import numpy as np; import matplotlib.pyplot as plt


#calculates derivative
def myDeriv(input, t):
    m=input[0]
    theta=input[1]
    dmdt=-theta/(np.sqrt(1+theta**2))
    dthetadt=m*np.cos(theta)
    return [dmdt, dthetadt]

#plot for various values of amplitudes
amplitudes = [0.25, 0.5, 1, 2, 4, 10, 20, 30]
plt.rcParams['figure.figsize'] = [10, 20]
for i in range(1,9):
    #Get amplitide
    m_ext = amplitudes[i-1]

    #Make the vector field on the phase plane
    grid_m = np.linspace(-m_ext, m_ext, 15)
    grid_theta = np.linspace(-1.5, 1.5, 15)
    cs, ps = np.meshgrid(grid_m, grid_theta, indexing='xy')
    cdot, pdot = myDeriv([cs,ps], 0)

    #Make the vector field for the stream plot (more dense, with stream_points to start flow lines)
    grid2_m = np.linspace(-m_ext, m_ext, 800)
    grid2_theta = np.linspace(-1.5, 1.5, 800)
    cs2, ps2 = np.meshgrid(grid_m, grid_theta, indexing='xy')
    cdot2, pdot2 = myDeriv([cs2,ps2], 0)
    stream_points = np.array(list(zip(np.linspace(-m_ext, m_ext, 15), np.linspace(-1.5, 1.5, 15))))

    #plot settings
    plt.subplot(4,2, i)
    fig = plt.gcf()
    fig.suptitle(r"Basic Mechanical Oscillator: Phase Portraits and Flow Lines for various values of $\Delta \b
    plt.title(r"$\Delta \bar m = $" + str(m_ext))
    plt.tight_layout(pad=2.0)
    plt.ylabel(r'$\theta$'); plt.xlabel(r'$\bar \Delta m$')

    #different widths so arrows are visible in quiver
    widths = [0.013, 0.013, 0.013, 0.02, 0.04, 0.08, 0.18, 0.21]
    plt.quiver(cs, ps, cdot, pdot, units='xy', angles='xy', linewidths=widths[i-1], headwidth = 4, scale=8)
    plt.streamplot(cs, ps, cdot, pdot, start_points=stream_points, linewidth=2, arrowsize=.7, density = 50)

    #horizontal nullcline
    plt.plot(np.linspace(-m_ext,m_ext, 15), np.zeros(15), 'r-', lw=2, label=r'Nullcline: $\theta = 0$')
    plt.plot(np.zeros(15), np.linspace(-1.5,1.5, 15), 'g-', lw=2, label=r'Nullcline: $\Delta \bar m = 0$')
    plt.plot(np.linspace(-m_ext,m_ext, 15), np.repeat(np.pi/2, 15), 'b-', lw=2, label=r'Nullcline: $\theta = \f
    plt.plot(np.linspace(-m_ext,m_ext, 15), np.repeat(-1*np.pi/2, 15), 'y-', lw=2, label=r'Nullcline: $\theta =
    plt.plot(0, 0, marker="o", markersize=10, markeredgecolor="cyan", markerfacecolor="cyan", label=r"Fixed Poi
    plt.legend()
```



[illegible]

b: Repeated process for relaxation oscillator

We repeat the above process for the relaxation oscillator.

Our equations are

$$\frac{d(\Delta\bar{m})}{d\bar{t}} = -\theta/\sqrt{1+\theta^2}$$

$$\frac{d\theta}{d\bar{t}} = \Delta\bar{m}\cos\theta + (\theta - \theta^3)$$

Note the original equation: $\frac{d\theta}{dt} = \Delta m\cos\theta + \alpha(\theta - \theta^3)$. When we use our substitutions above, we get the same simplification for $\Delta m\cos\theta$, and some contant for α. From the problem directions, we simply assume $\alpha = 1$.

We also add nullclines to the below diagram.

The nullclines occur when either $\frac{d(\Delta\bar{m})}{d\bar{t}}$ or $\frac{d\theta}{d\bar{t}}$ is 0.

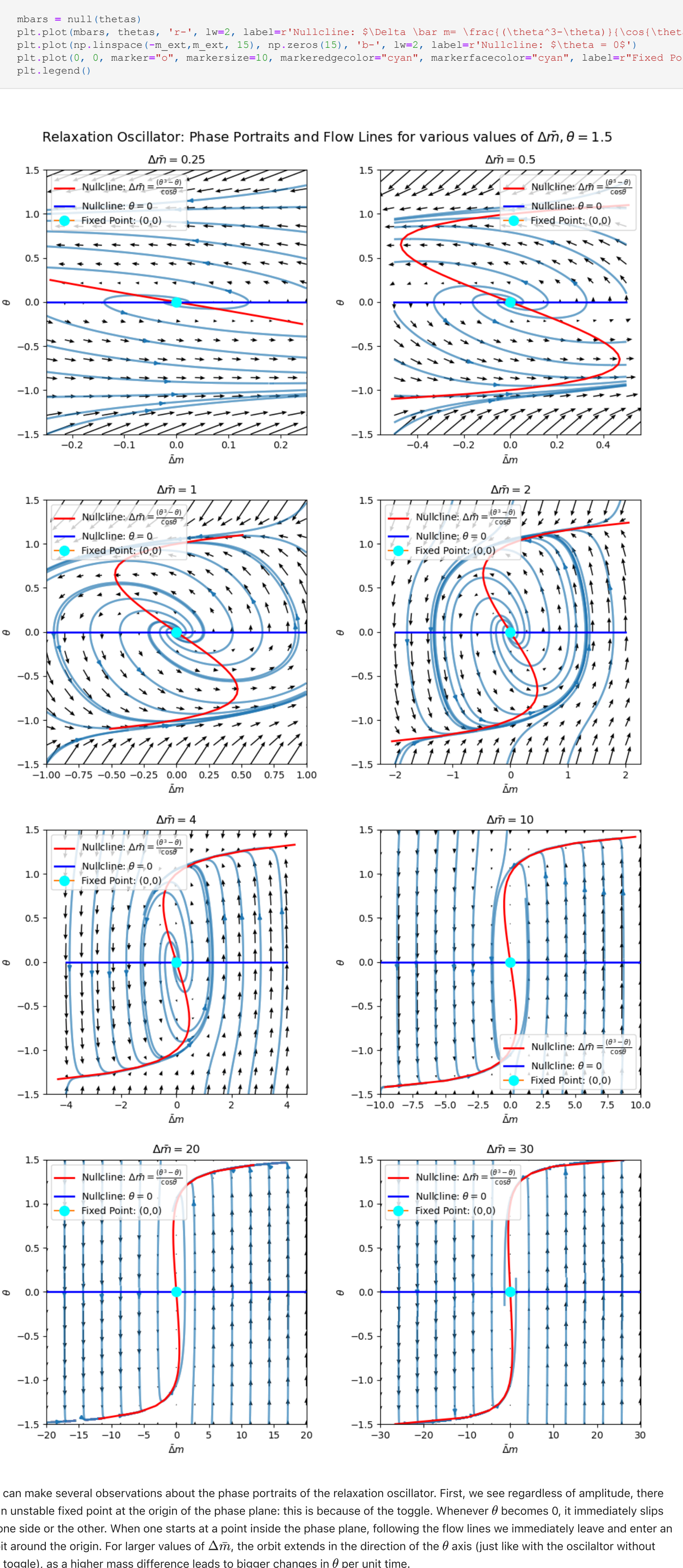
$$\frac{d(\Delta\bar{m})}{d\bar{t}} = -\theta/\sqrt{1+\theta^2} = 0 \implies \theta = 0$$

$$\frac{d\theta}{d\bar{t}} = \Delta\bar{m}\cos\theta + (\theta - \theta^3) = 0 \implies \Delta\bar{m} = \frac{(\theta^3 - \theta)}{\cos\theta}$$

Our nullclines are $\theta = 0 and \pm\frac{\pi}{2}, \Delta\bar{m} = 0$. We draw them below.

In [152…

```
%matplotlib notebook

import numpy as np; import matplotlib.pyplot as plt


#calculates derivative
def myDeriv(input, t):
    m=input[0]
    theta=input[1]
    dmdt=-theta/(np.sqrt(1+theta**2))
    dthetadt=m*np.cos(theta)+(theta-theta**3)
    return [dmdt, dthetadt]

#plot for various values of amplitudes
amplitudes = [0.25, 0.5, 1, 2, 4, 10, 20, 30]
plt.rcParams['figure.figsize'] = [10, 20]
for i in range(1,9):
    #Get amplitide
    m_ext = amplitudes[i-1]

    #Make the vector field on the phase plane
    grid_m = np.linspace(-m_ext, m_ext, 15)
    grid_theta = np.linspace(-1.5, 1.5, 15)
    cs, ps = np.meshgrid(grid_m, grid_theta, indexing='xy')
    cdot, pdot = myDeriv([cs,ps], 0)

    #Make the vector field for the stream plot (more dense, with stream_points to start flow lines)
    grid2_m = np.linspace(-m_ext, m_ext, 800)
    grid2_theta = np.linspace(-1.5, 1.5, 800)
    cs2, ps2 = np.meshgrid(grid_m, grid_theta, indexing='xy')
    cdot2, pdot2 = myDeriv([cs2,ps2], 0)
    stream_points = np.array(list(zip(np.linspace(-m_ext, m_ext, 15), np.linspace(-1.5, 1.5, 15))))

    #plot settings
    plt.subplot(4,2, i)
    fig = plt.gcf()
    fig.suptitle(r"Relaxation Oscillator: Phase Portraits and Flow Lines for various values of $\Delta \bar m,
    plt.title(r"$\Delta \bar m = $" + str(m_ext))
    plt.tight_layout(pad=2.0)
    plt.ylabel(r'$\theta$'); plt.xlabel(r'$\bar \Delta m$')

    #different widths so arrows are visible in quiver
    widths = [0.013, 0.013, 0.013, 0.02, 0.04, 0.08, 0.18, 0.21]
    plt.quiver(cs, ps, cdot, pdot, units='xy', angles='xy', linewidths=widths[i-1], headwidth = 4, scale=8)
    plt.streamplot(cs, ps, cdot, pdot, start_points=stream_points, linewidth=2, arrowsize=.7, density = 80)

    #Graph nullcline
    nullc_ints = [0.25, 1.1, 1.1, 1.24, 1.33, 1.42, 1.44, 1.5]
    thetas = np.linspace(-nullc_ints[i-1], nullc_ints[i-1], 40)
    def null(c):
        return (c**3-c)/(np.cos(c))

    mbars = null(thetas)
    plt.plot(mbars, thetas, 'r-', lw=2, label=r'Nullcline: $\Delta \bar m= \frac{(\theta^3-\theta)}{\cos{\theta
    plt.plot(np.linspace(-m_ext,m_ext, 15), np.zeros(15), 'b-', lw=2, label=r'Nullcline: $\theta = 0$')
    plt.plot(0, 0, marker="o", markersize=10, markeredgecolor="cyan", markerfacecolor="cyan", label=r"Fixed Poi
    plt.legend()
```



We can make several observations about the phase portraits of the relaxation oscillator. First, we see regardless of amplitude, there is an unstable fixed point at the origin of the phase plane: this is because of the toggle. Whenever θ becomes 0, it immediately slips to one side or the other. When one starts at a point inside the phase plane, following the flow lines we immediately leave and enter an orbit around the origin. For larger values of $\Delta\bar{m}$, the orbit extends in the direction of the θ axis (just like with the oscilaltor without the toggle), as a higher mass difference leads to bigger changes in θ per unit time.

Specifically, when we note the tendency for the system to enter the orbit, we are observing a limit cycle. We see in the phase portaits when we start with a small θ and small $\Delta\bar{m}$, we move away form the origin into a cycle. We observe two more steady states (two points on the cycle where the surrounding vectors point in), and see the relaxation oscilaltor has bistable behavior.

Credit

Marija for nondimensionalization help, Andres for graph insight