Plotting the harmonic oscillator

Below are the two example scripts used in today's class. If you save each one into separate files, e.g. **`ex_1.py`** and **`ex_2.py`** then you can run them by first running python by typing

`python3`

into a command prompt, followed by

`exec(open("./ex_1.py").read())`

which will run the script.

```
import numpy as np
import matplotlib.pyplot as plt
#
phi = 0
w = 1
#
t = np.linspace(0,2*np.pi,1e3)
x = np.cos(w*t+phi)
f1 = plt.figure()
plt.plot(t/(np.pi),np.cos(w*t+phi),label=r'$A=1, \omega=1,
\phi=0$',linewidth='2')
plt.plot(t/(np.pi),0.5*np.cos(2*w*t+phi),label=r'$A=0.5, \omega=2,
\phi=0$',linewidth='2')
plt.plot(t/(np.pi),np.cos(w*t+np.pi),'--',label=r'$A=1, \omega=1,
\phi=\pi$',linewidth='2')
plt.xlabel(r'$\omega t/\pi$', fontsize=14)
plt.ylabel(r'$x(t)$', fontsize=14)
plt.xlim([t[0],t[-1]/np.pi])
plt.ylim([x.min(),x.max()])
plt.grid()
plt.legend(loc='upper right')
f1.show()
#
```

Damped Harmonic Oscillator

```
import numpy as np
import matplotlib.pyplot as plt
#
x0 = 1
p0 = 0
dt = 1e-4
w = 1
g = 0.5
T = 4*2*np.pi
Nt = int(round(T/dt))
tsc = np.linspace(0,T,Nt)
#
x = np.zeros(Nt);
p = np.zeros(Nt);
x[0] = x0;
p[0] = p0;
#
for jj in range(0,Nt-1):
    #
    p[jj+1] = p[jj] - dt*(g*p[jj] + w**2*x[jj])
    x[jj+1] = x[jj] + dt*p[jj]
    #
#
f = plt.figure()
data =
plt.plot(tsc[::200]/np.pi,x[::200],tsc[::200]/np.pi,p[::200],linewidth='2')
plt.title('Harmonic oscillator solutions')
plt.autoscale(enable=True, axis='x', tight=True)
plt.xlabel(r'$t/\pi$', fontsize='14')
plt.ylabel('',fontsize='14')
plt.grid()
f.show()
#
f2 = plt.figure()
ax = plt.axes()
plt.plot(x[::200],p[::200])
ax.set_aspect('equal', adjustable='box')
ax.set_xlabel(r'$x$', fontsize=14)
ax.set_ylabel(r'$p$', fontsize=14)
plt.grid()
f2.show()
#
```