Harmonic Oscillator

August 6, 2024

[3]:

```python
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
#sns.set() loads seaborn's default theme and color palatte
```

[4]:

```python
t = np.linspace(0,15,1000)
omega_sq = 1 #this means k/m = 1
c0 = 0.3
y = [0,2] #y[0]=x and y[1]=v
```

[5]:

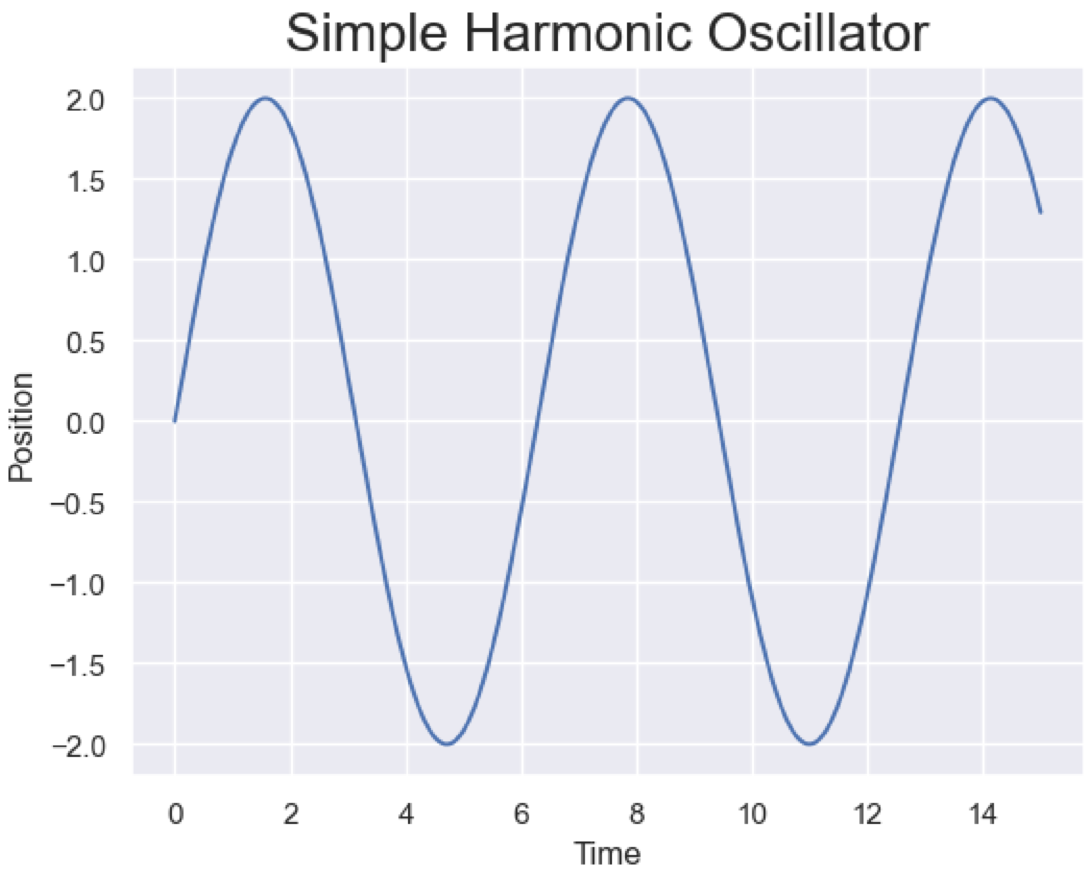
```python
def harmonic(t,y):
    solution = [y[1],-omega_sq*y[0]]
    return solution
# solution is [v, -kx/m]
sho = solve_ivp(harmonic, [0,1000], y0 = y, t_eval = t)
#solve_ivp is scipy function
```

[6]:

```python
plt.plot(t,sho.y[0])
plt.ylabel("Position")
plt.xlabel("Time")
plt.title('Simple Harmonic Oscillator', fontsize = 20)
```

[6]:

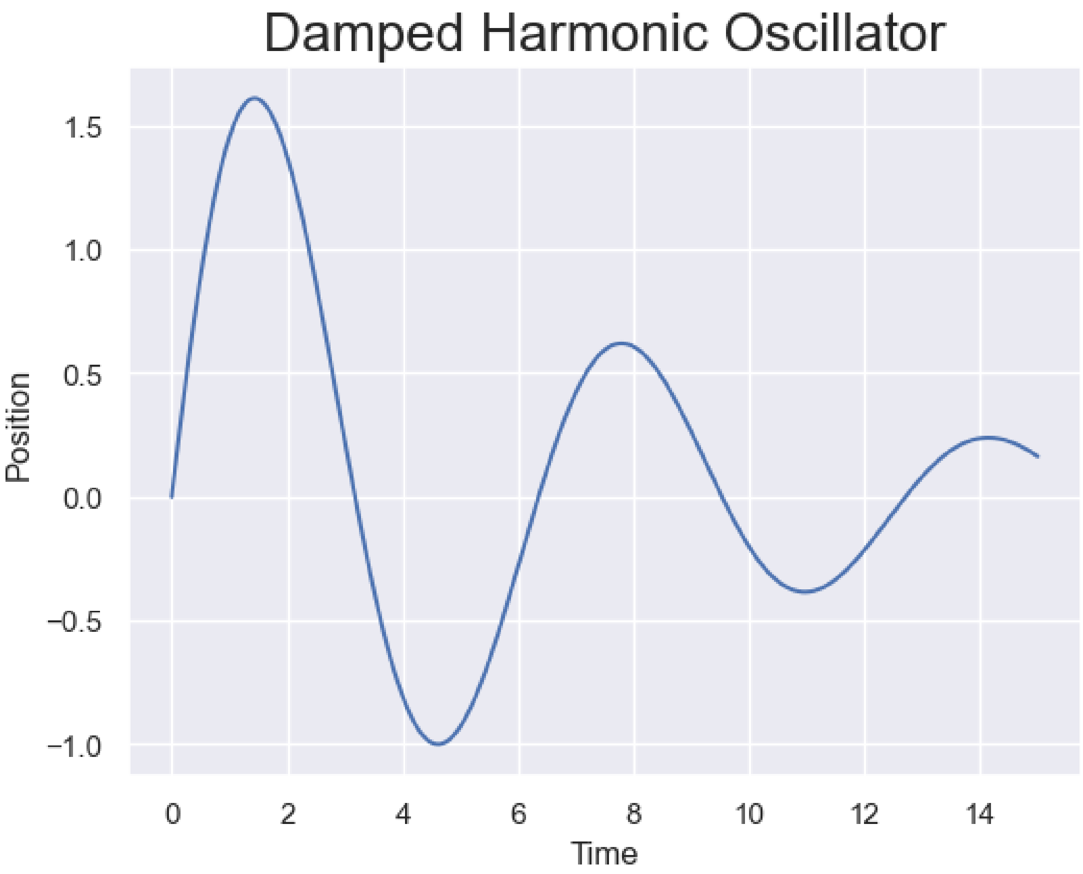
```
Text(0.5, 1.0, 'Simple Harmonic Oscillator')
```



[7]:
```
#now let's consider the damped harmonic oscillator
def damped(t,y):
    solution = [y[1],-omega_sq*y[0]-c0*y[1]]
    return solution
# solution is [v, -kx/m-cv]
dsho = solve_ivp(damped, [0,1000], y0 = y, t_eval = t)
#solve_ivp is scipy function
```

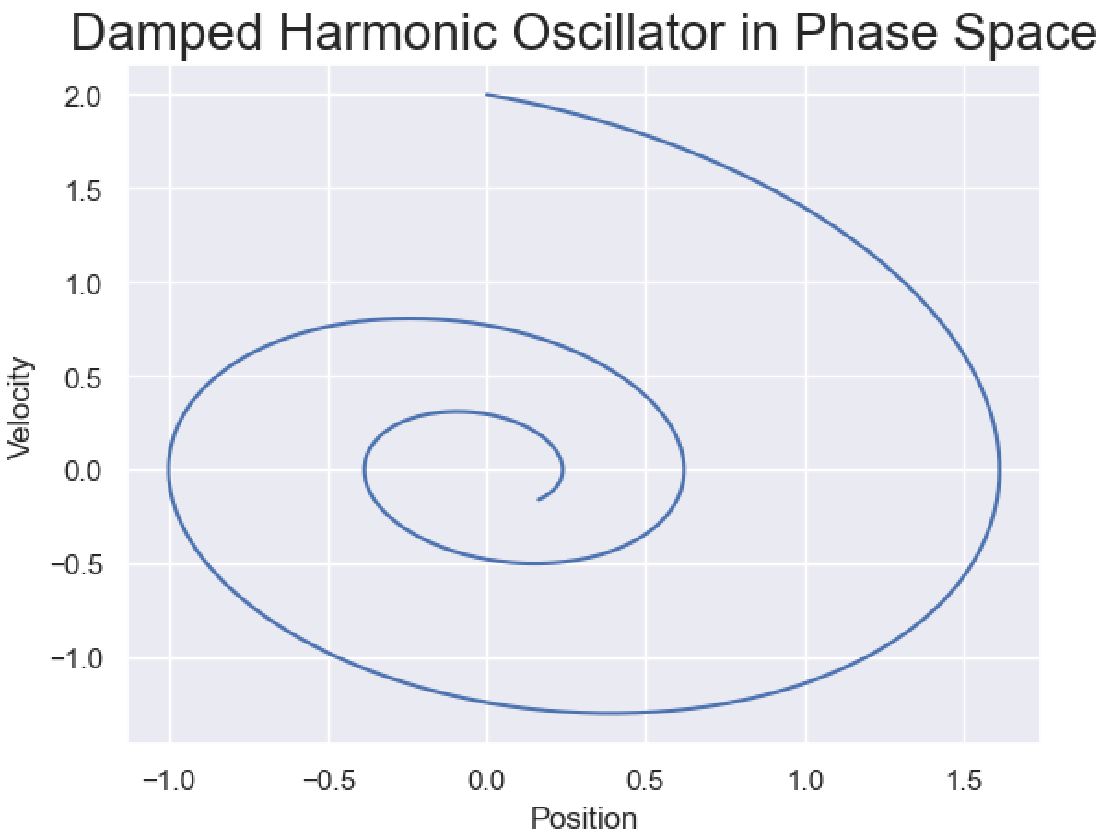
[37]:
```
plt.plot(t,dsho.y[0])
plt.ylabel("Position")
plt.xlabel("Time")
plt.title('Damped Harmonic Oscillator', fontsize = 20)
```

[37]:
```
Text(0.5, 1.0, 'Damped Harmonic Oscillator')
```



```
[12]: #what if we throw this in phase space?
      plt.plot(dsho.y[0],dsho.y[1])
      plt.ylabel("Velocity")
      plt.xlabel("Position")
      plt.title('Damped Harmonic Oscillator in Phase Space', fontsize = 20)
```

```
[12]: Text(0.5, 1.0, 'Damped Harmonic Oscillator in Phase Space')
```



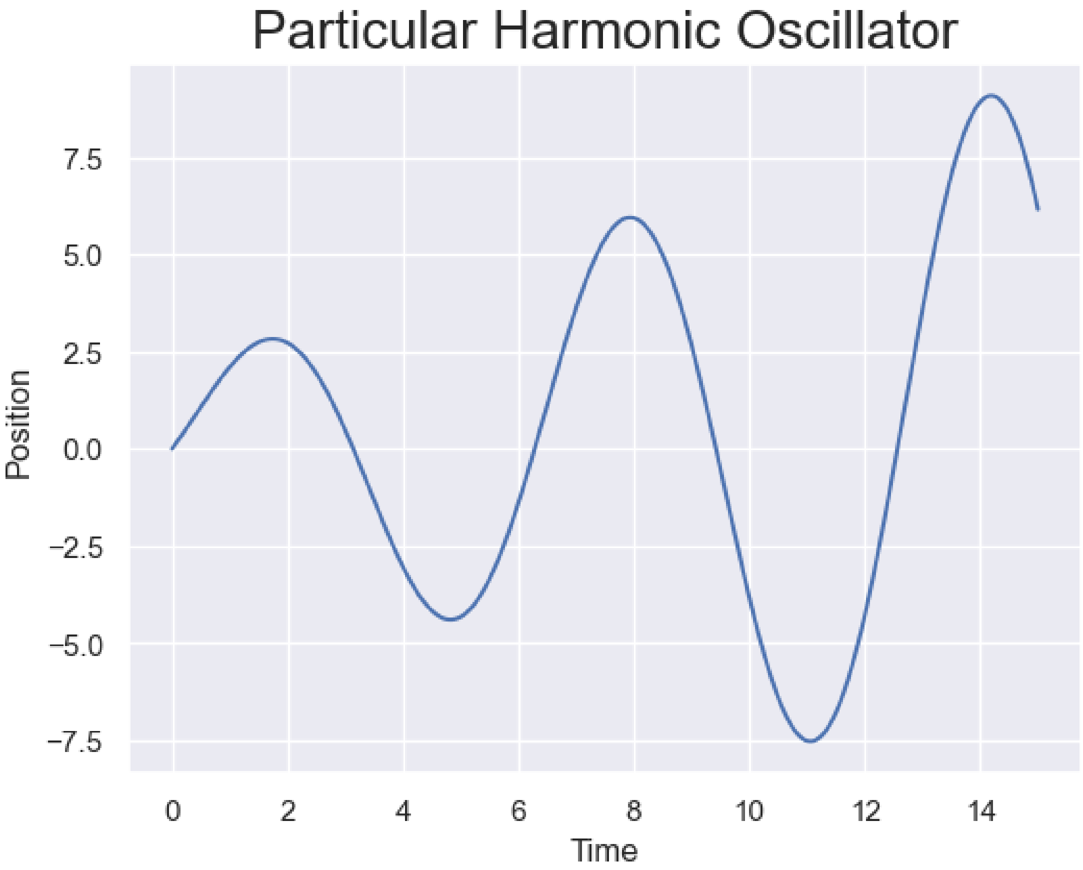
[40]:
```
#what if we hit it with an external force?
omega = 1
amp = 1

def extforce(t,omega,amp):
    solution = amp*np.cos(omega*t)
    return solution

def particular(t,y):
    solution = [y[1],-omega_sq*y[0]+extforce(t,omega,amp)]
    return solution
# solution is [v, -kx/m]
psho = solve_ivp(particular, [0,1000], y0 = y, t_eval = t)
#solve_ivp is scipy function
```

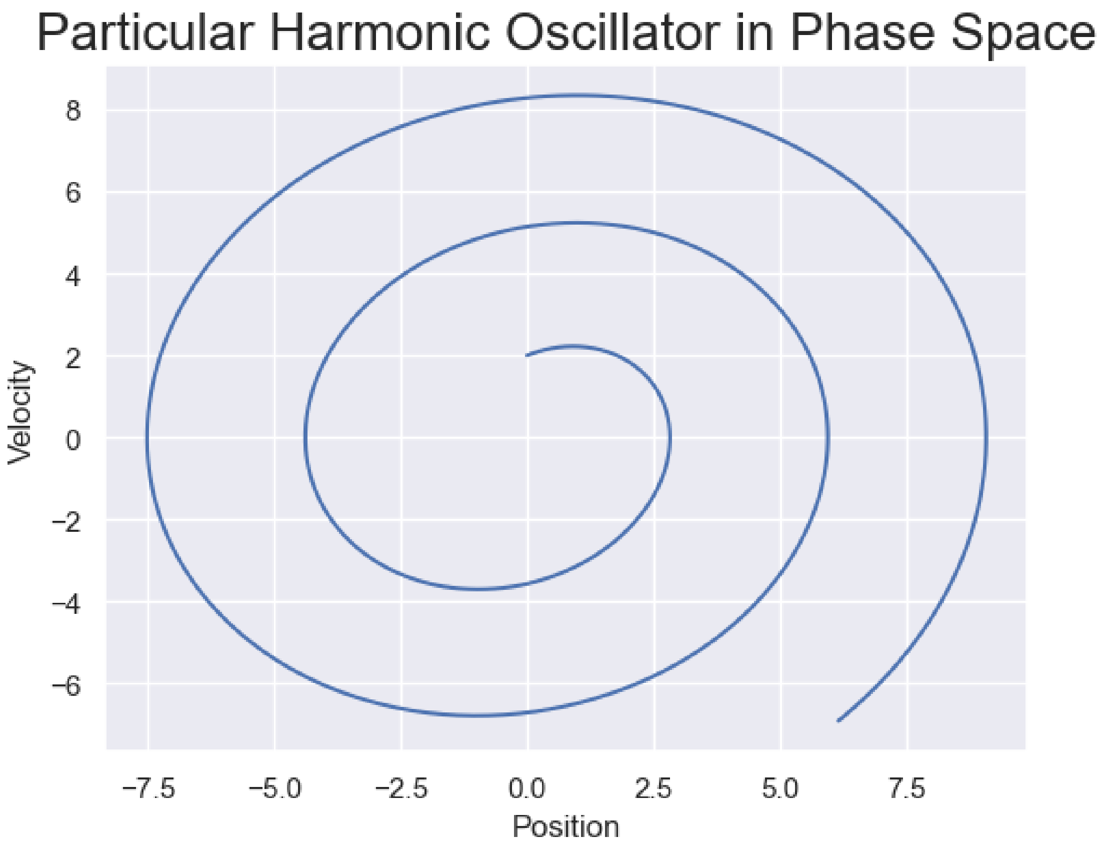
[41]:
```
plt.plot(t,psho.y[0])
plt.ylabel("Position")
plt.xlabel("Time")
plt.title('Particular Harmonic Oscillator', fontsize = 20)
```

```
[41]: Text(0.5, 1.0, 'Particular Harmonic Oscillator')
```



```
[15]: #let's throw that in phase space
      plt.plot(psho.y[0],psho.y[1])
      plt.ylabel("Velocity")
      plt.xlabel("Position")
      plt.title('Particular Harmonic Oscillator in Phase Space', fontsize = 20)
```

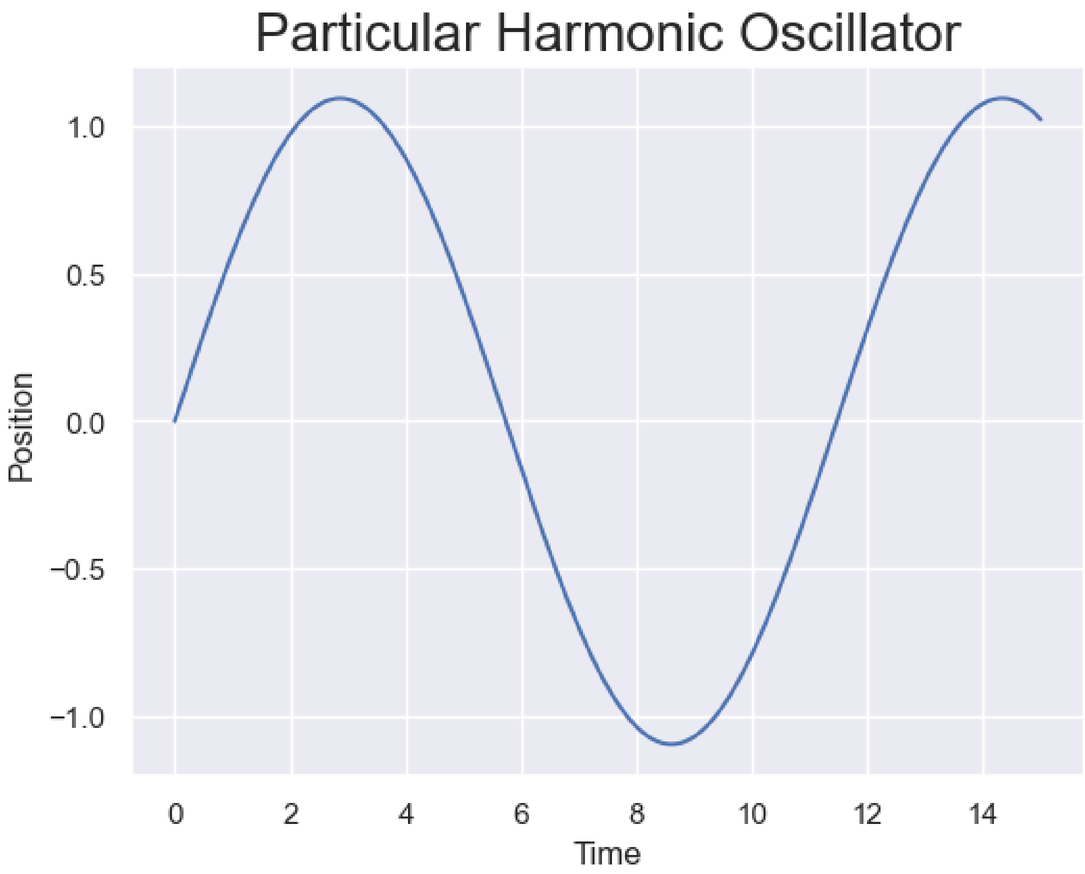
```
[15]: Text(0.5, 1.0, 'Particular Harmonic Oscillator in Phase Space')
```



```
[29]: #did I damp this right? let me try something
      def damped2(t,y):
          solution = [c0*y[1],-omega_sq*y[0]]
          return solution
      # solution is [v, -kx/m-cv]
      dsho2 = solve_ivp(damped2, [0,1000], y0 = y, t_eval = t)
      #solve_ivp is scipy function
```

```
[32]: #looks like my original was correct
      plt.plot(t,dsho2.y[0])
      plt.ylabel("Position")
      plt.xlabel("Time")
      plt.title('Particular Harmonic Oscillator', fontsize = 20)
```

```
[32]: Text(0.5, 1.0, 'Particular Harmonic Oscillator')
```


Particular Harmonic Oscillator
Position
1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8
10
12
14
Time

[ ]: