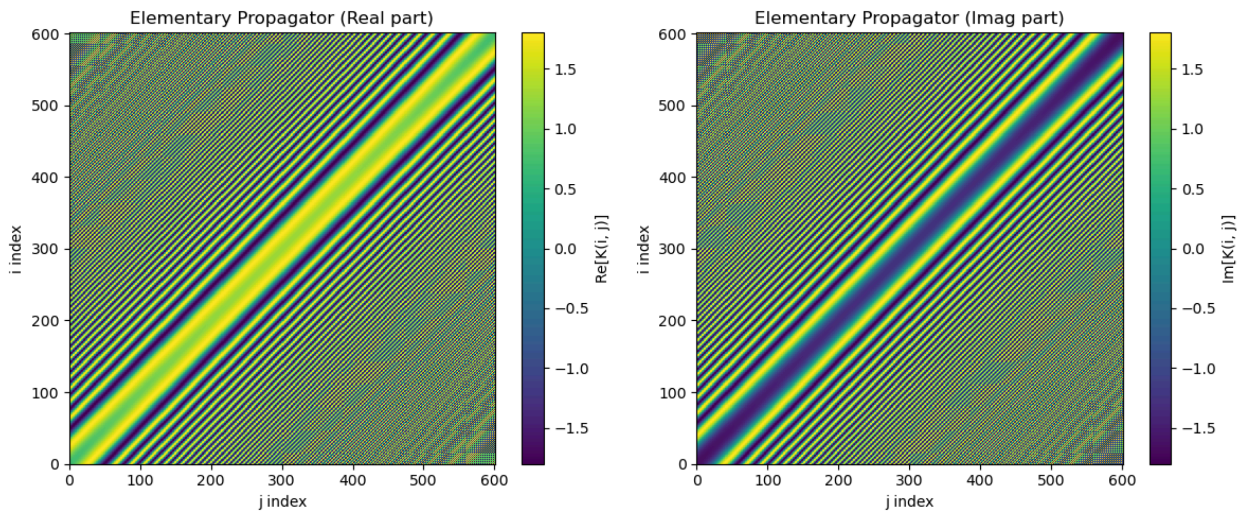
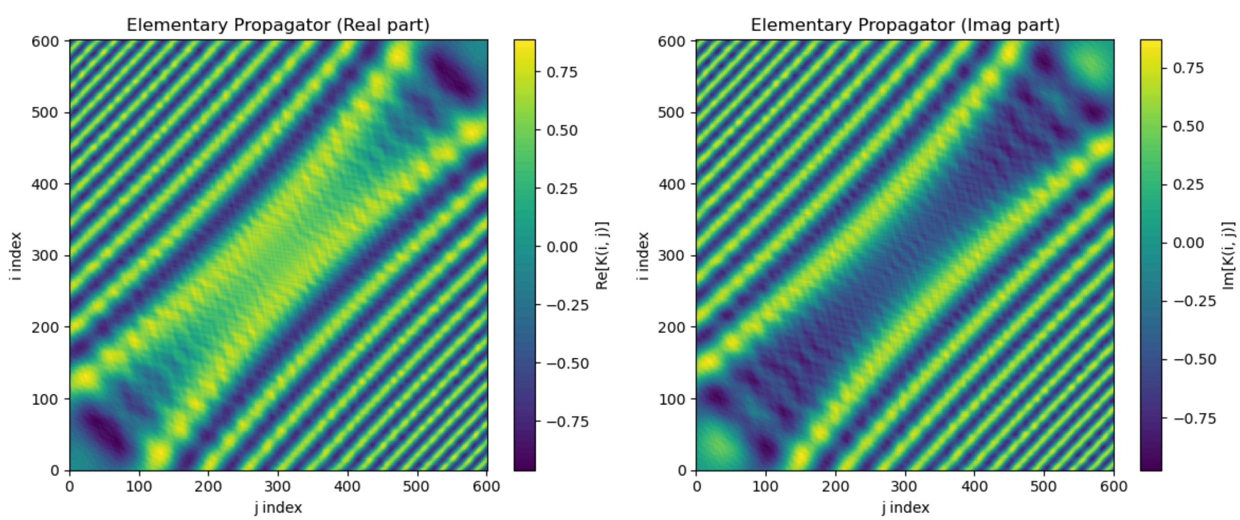
PROBLEM A





My propagator matrices align well with the graphs shown in class. The analog to the integral over the wave function multiplied by the propagator is the propagator matrix operating on the wave function vector. To advance the simulation one timestep you need to multiply the wave function by the elementary propagator

matrix and scalar multiply by delta x. The numerical integration is essentially the Riemann sum. In order to jump forward several N time steps you need to multiply the propagator matrix to a power of N. This approach will likely result in a memory overflow, so it's best to use a loop to get to the Nth timestep. This is the most difficult part of the assignment and rest can be understood intuitively once you know how to get the probability density by squaring the wave function and advancing the wave function by N timesteps.

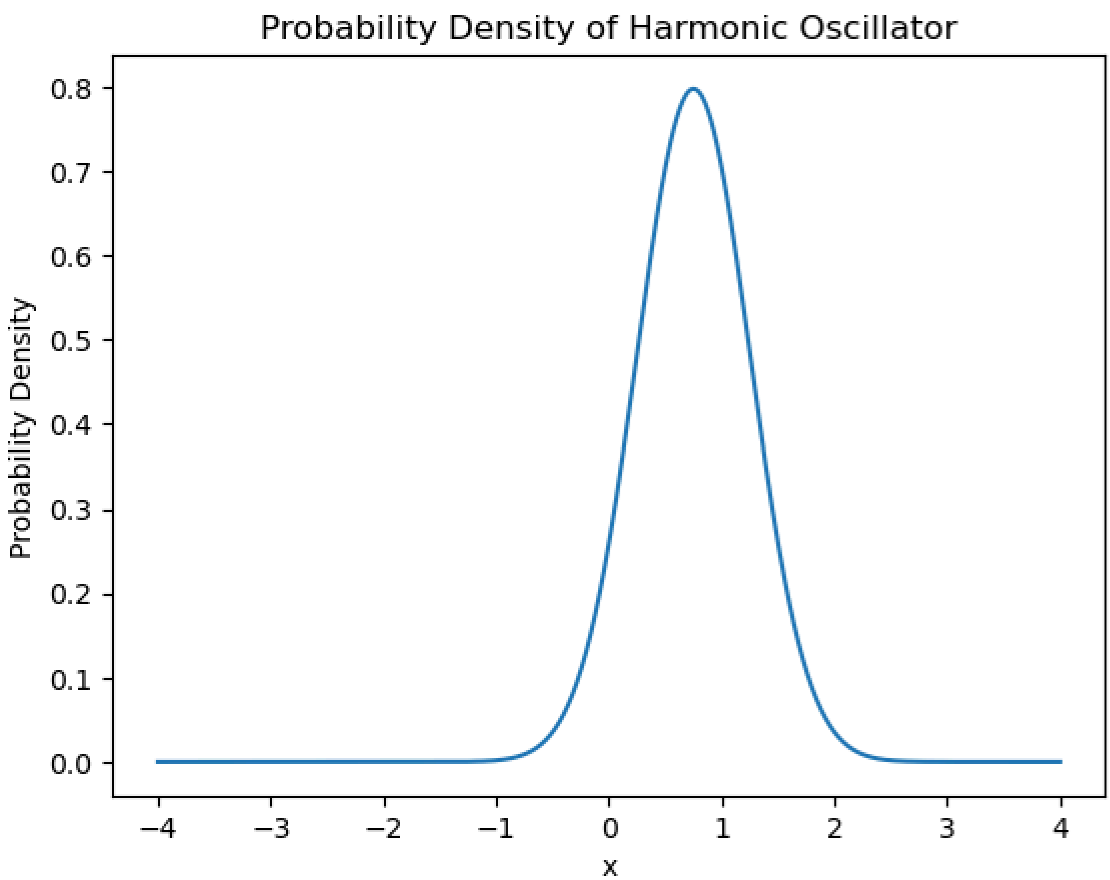
```python
print(k_8eps)
```

[24] ✓ 0.0s Open 'k_8eps' in Data Wrangler

```
[[-0.05955857+0.01786716j -0.07576122+0.02170276j -0.07967379+0.05414069j
  ... -0.46505934-0.30932662j -0.35439298-0.42735606j
  -0.2168743 -0.50917196j]
 [-0.07576125+0.02170283j -0.0896081 +0.03330258j -0.08308481+0.06724797j
  ... -0.5340409 -0.16837671j -0.4614194 -0.31131655j
  -0.3543931 -0.42735562j]
 [-0.07967392+0.05414072j -0.08308499+0.06724811j -0.06226969+0.0930306j
  ... -0.5640737 -0.01040979j -0.53404117-0.16837703j
  -0.46506009-0.3093264j ]
 ...
 [-0.46505934-0.3093265j  -0.5340409 -0.1683766j  -0.5640738 -0.01040973j
  ... -0.06227011+0.09303051j -0.08308521+0.06724772j
  -0.0796742 +0.05414026j]
 [-0.3543929 -0.42735612j -0.46141946-0.3113167j  -0.53404105-0.16837695j
  ... -0.08308517+0.06724776j -0.08960827+0.03330211j
  -0.0757613 +0.02170216j]
 [-0.2168743 -0.5091721j  -0.3543931 -0.42735577j -0.4650602 -0.30932644j
  ... -0.07967409+0.05414031j -0.07576124+0.02170226j
  -0.05955863+0.01786645j]]
```

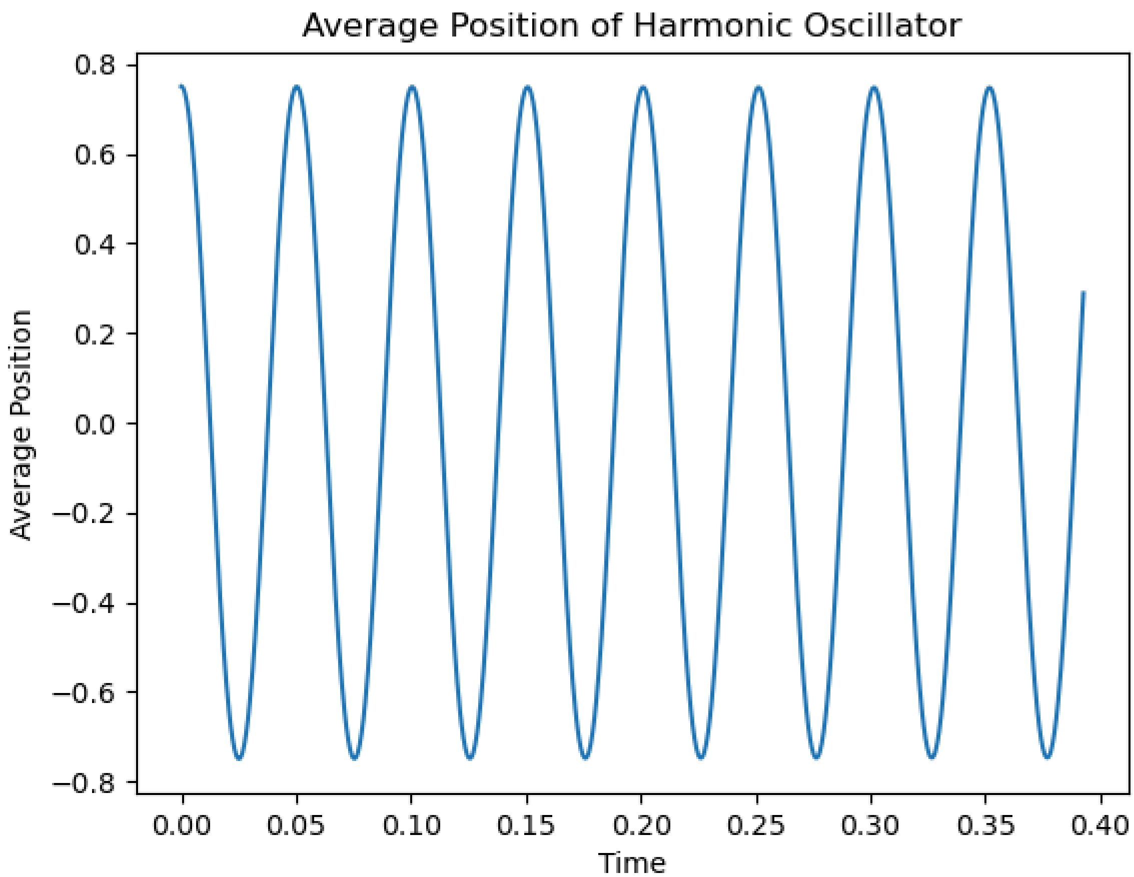
This data aligns exactly with the answers given in the solution indicating that the propagator matrix is correct and problem A is as well.

Problem B



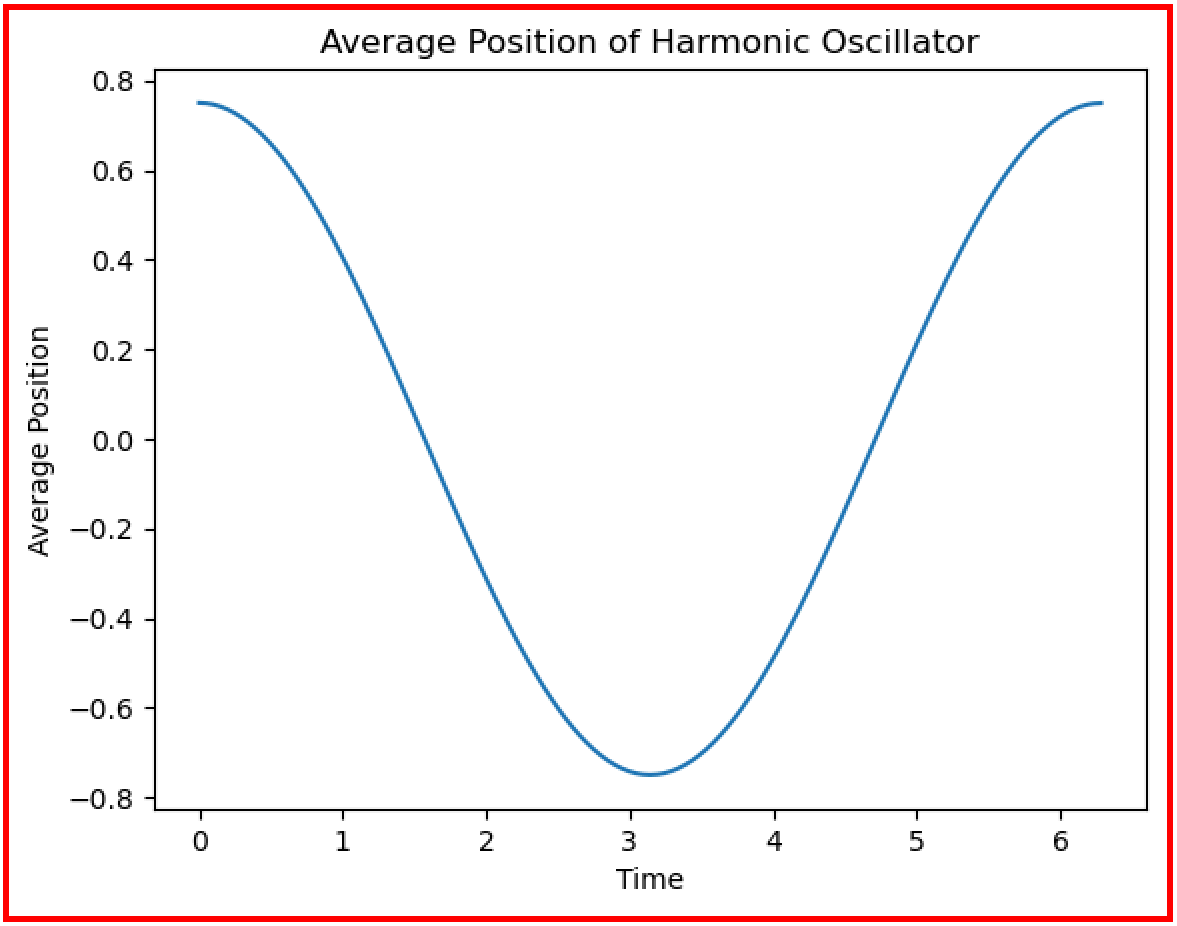
This is the probability density function plotted with respect to X. This was done to verify results and make sure the answer makes sense somewhat.

This was correct.



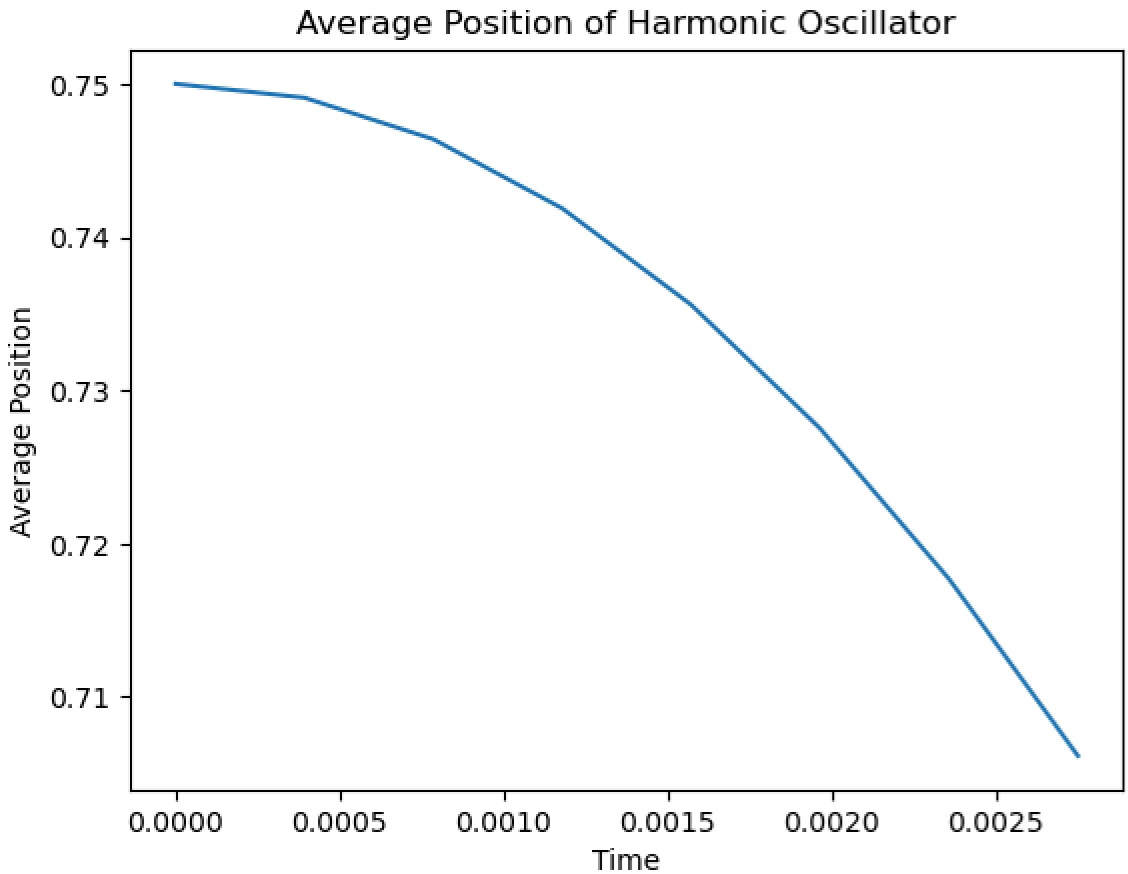
The problem asks to plot 8 timesteps but it's difficult to verify what the function is so I plotted 1000 to make sure I get periodic behaviour.

This graph is incorrect. Most of my mistakes in assignment 1 were due to timestep errors. This graph shows that the period of the average position is around 0.05 seconds, when it should be around 2Pi



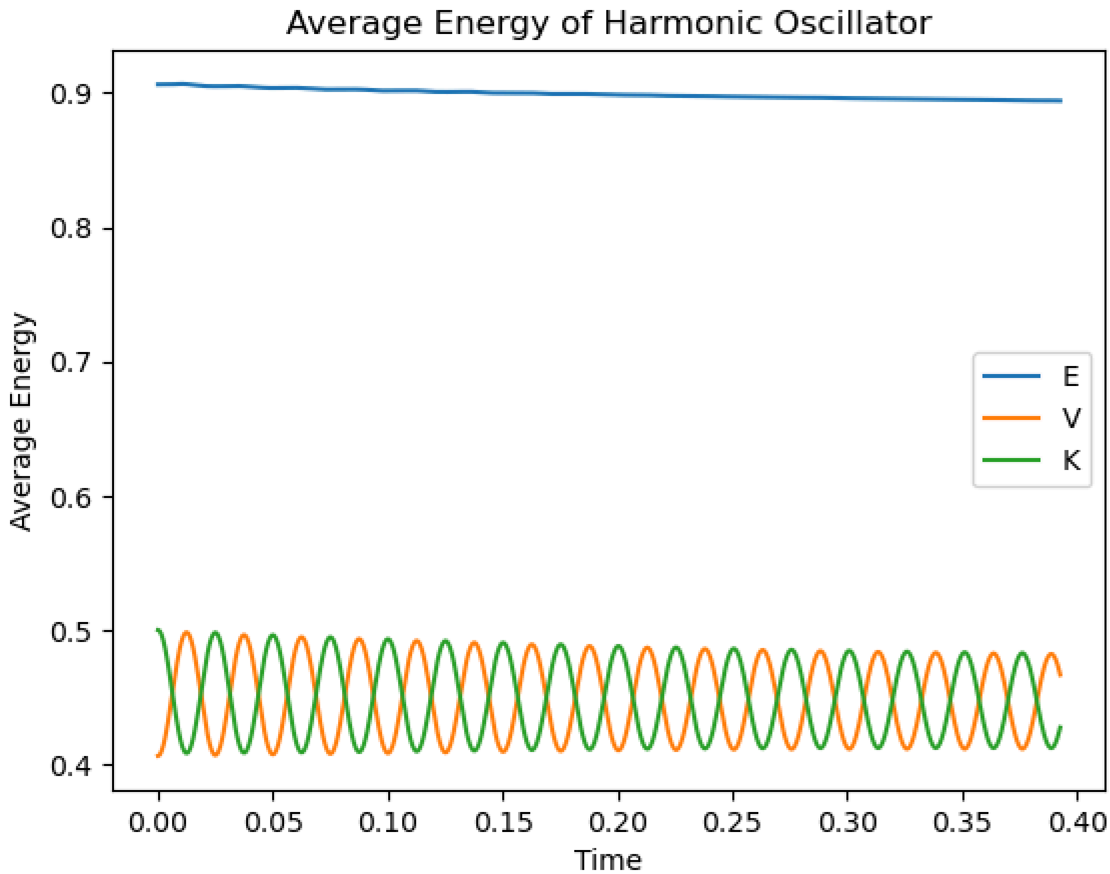
This is my corrected graph. It aligns much better with the example given in the solutions. The average position time evolves correctly here.

I also did not explain my work in the first attempt and did not show understanding so my explanation is as follows. This graph shows a periodic behaviour consistent with a harmonic oscillator because the particle starts from 0.75 as asked in the question and oscillates around the origin. This is essentially the solution to a classic harmonic oscillator.



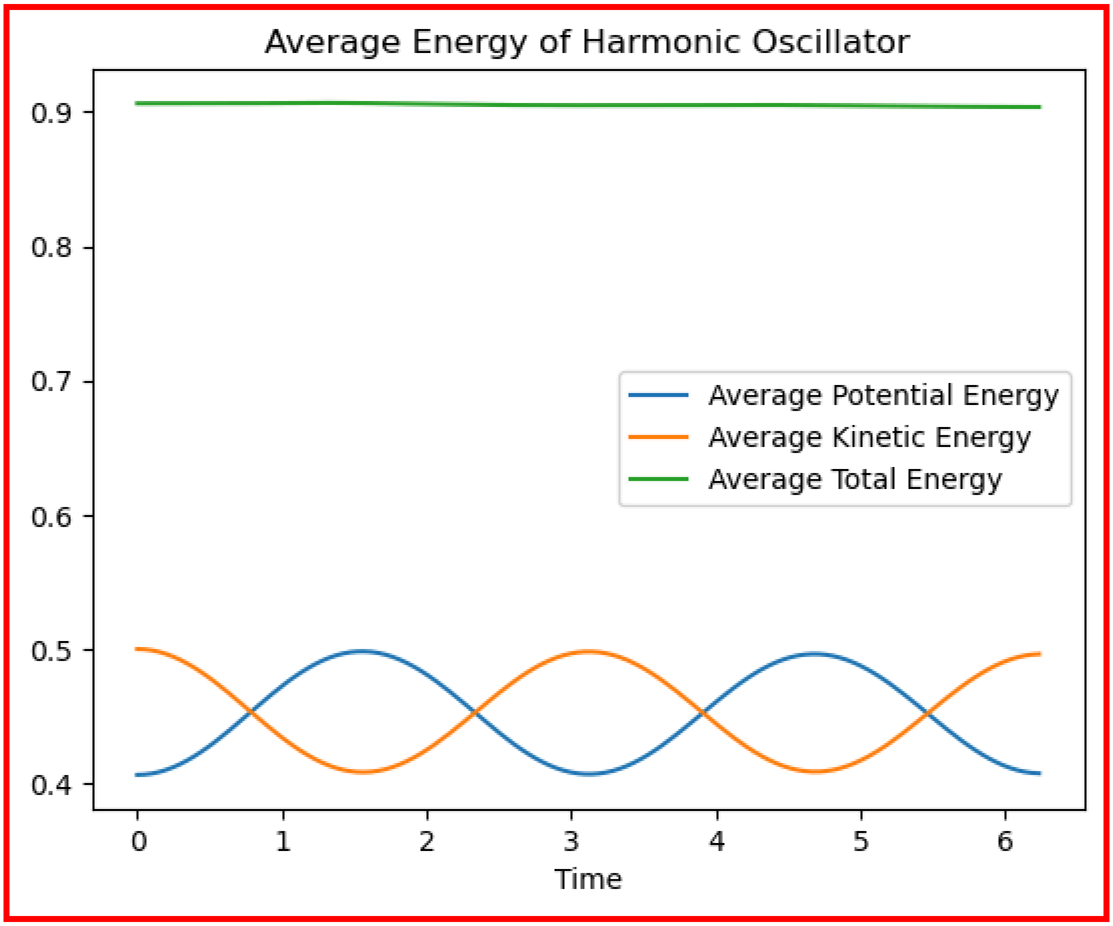
I believe my solution here was mostly correct except once again the timesteps were wrong. This is because I computed 1000 timesteps and likely I divided the timestep delta t by 1000. The first sharp edge on this graph occurs at around 0.0006 so it's likely 6.14/10000. I'm not sure where the extra factor of 10 comes from because I changed my code. This is likely just a minor issue with using np.linspace.

Problem C



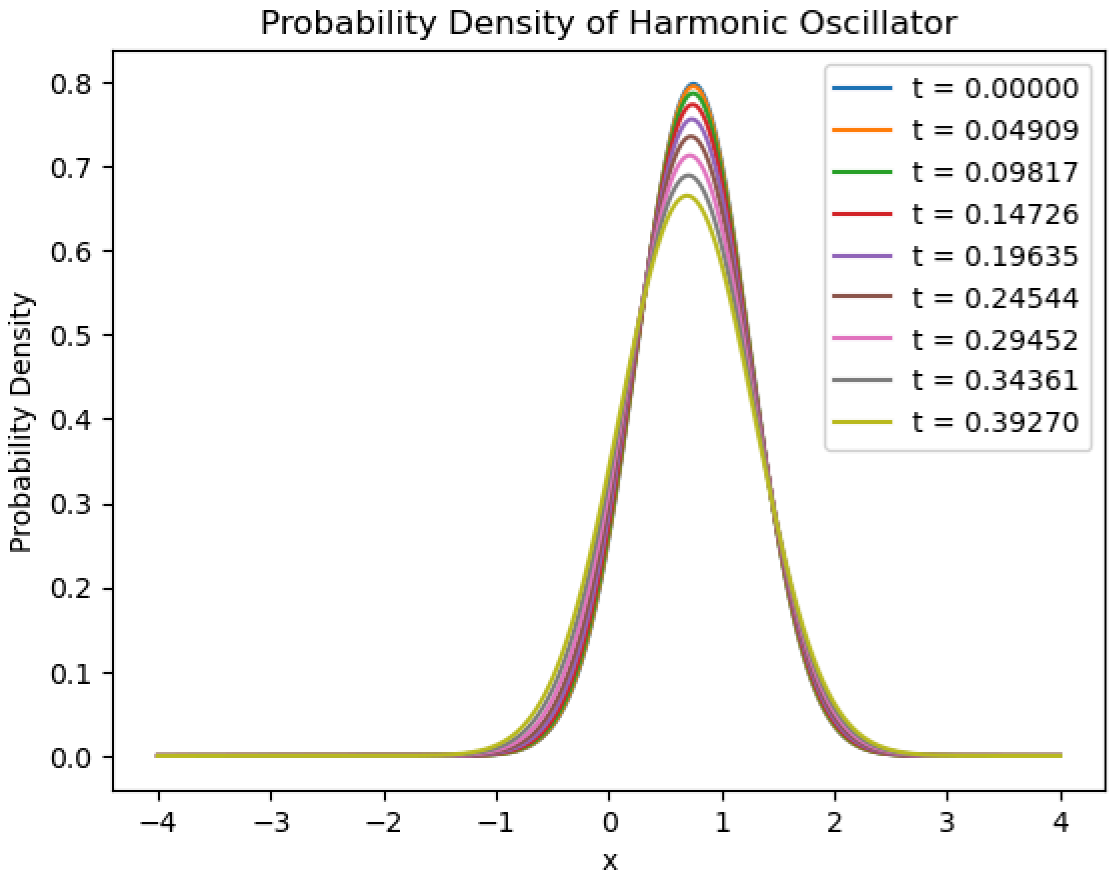
I once again plotted 1000 timesteps because I wanted to verify the results. The line for E is drifting due to numerical precision but should be a straight line.

The graph here looks correct, My timestep error just carried over. My new graph is below which aligns more closely with the solution set.

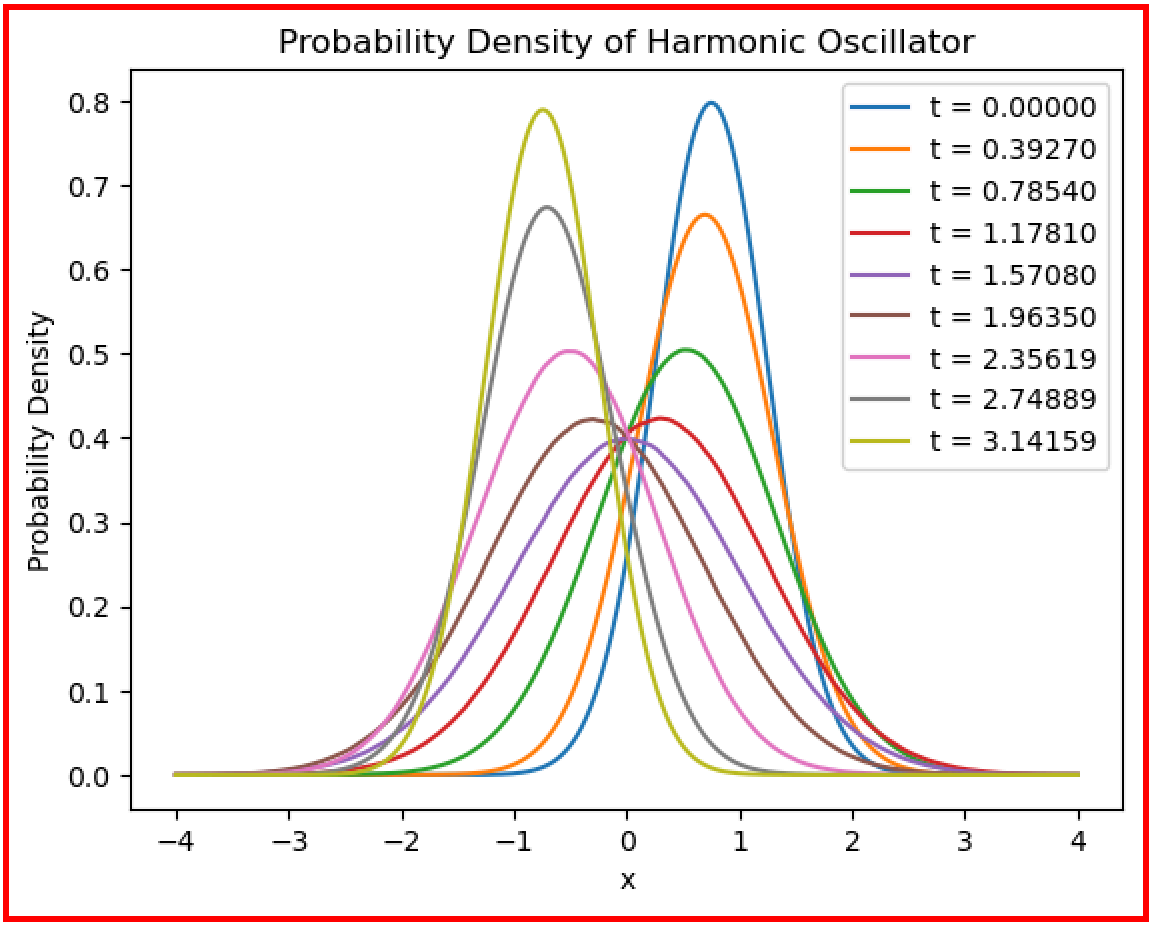


The average kinetic and potential energy also come out to the classical solution where they oscillate out of phase and the average energy remains constant.

Problem D

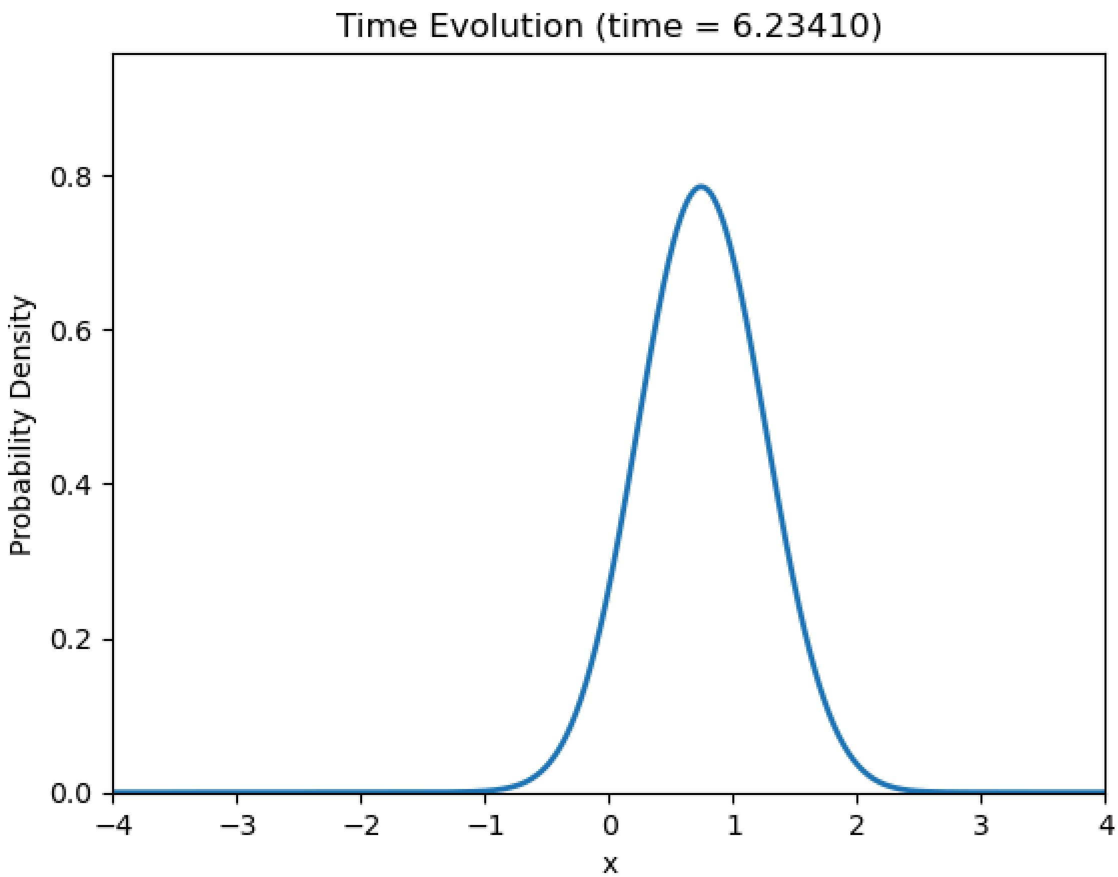


This was also correct except I misunderstood the problem and plotted 8 timesteps until T_0/16 rather than in timesteps of T_0/16 .



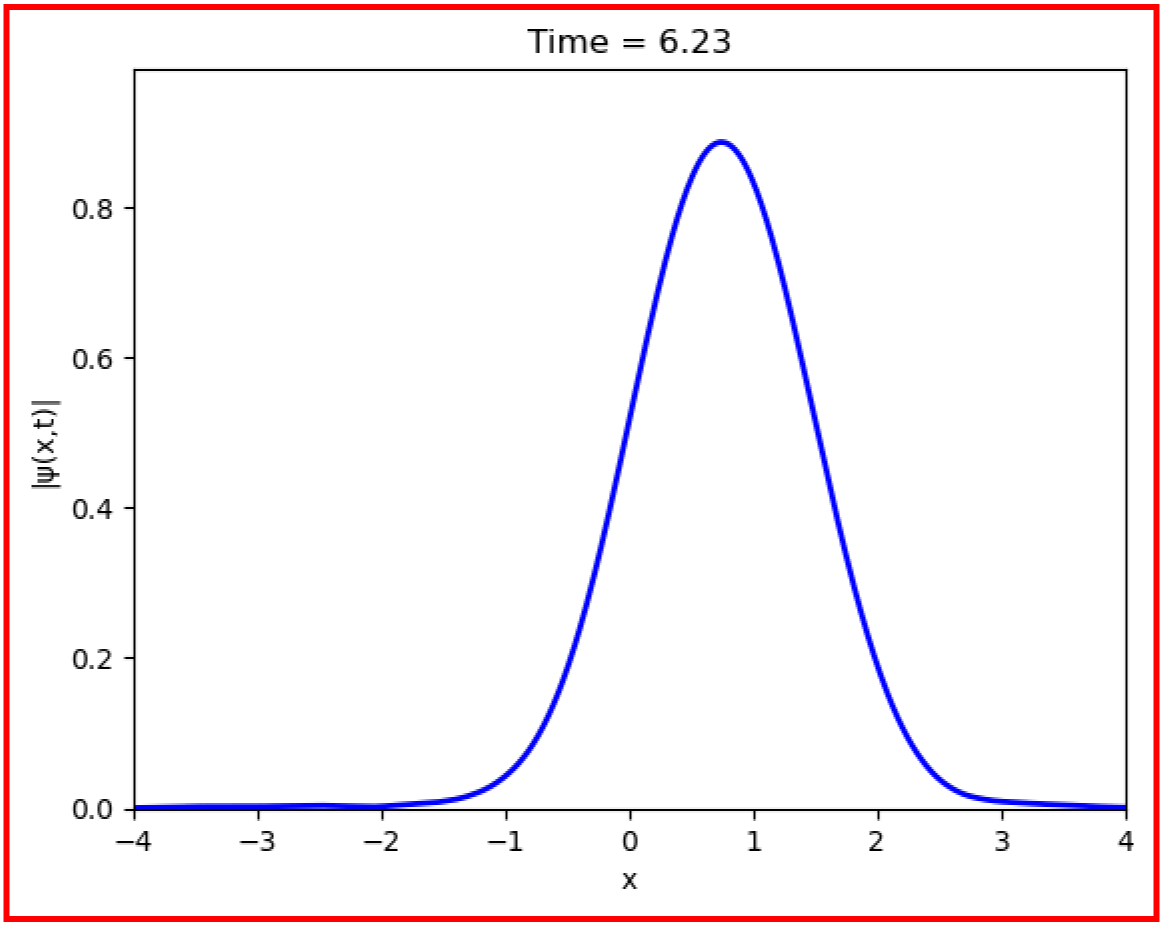
Probability Density of Harmonic Oscillator
Probability Density
x
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
−4
−3
−2
−1
0
1
2
3
4
t = 0.00000
t = 0.39270
t = 0.78540
t = 1.17810
t = 1.57080
t = 1.96350
t = 2.35619
t = 2.74889
t = 3.14159

Problem E



The GIF is bundled with the code in the zip file.

I had to redo this because I accidentally deleted all my code since the last commit but my new gif is the same except a darker blue.



Time = 6.23
|ψ(x,t)|
0.0
0.2
0.4
0.6
0.8
−4
−3
−2
−1
0
1
2
3
4
x