Lab 7 Problem Set Questions

[HW4-1]

Discuss whether the variations in E are significant or not. By how many percent is E varying in your MD code?

i. For the time step I used dt = 31356 which is approximately 1/3 of the amount of seconds in a day. I used N = 1000 for the number of steps. After calculating my values of E, I took the difference of the first two values of it, took the result of that and divided by the initial value in energy and then multiped by 100. This told me that the variation in energy was approximately .004%. This percent is so small it's not considered significant. Going back to the analogy of tuition, if I were paying 4500 per year, a raise of .004% would mean 18cents extra per year. Below is an image of my txt file and my graph for the values of E vs. t

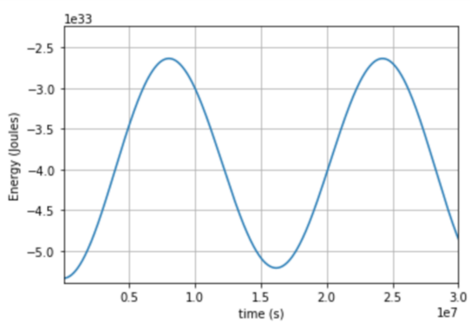
In [1]:
```python
import numpy as np
import matplotlib.pyplot as plt
```

In [5]:
```python
file_path1 = '/Users/Austin/Desktop/PHY 40/Homework/4/PS4_1.txt'

x, y = np.loadtxt(file_path1, unpack = True) #Method to extract/import information from txt file

# print(min(x), max(x))
# print(min(y), max(y))


plt.plot(x,y)
plt.xlabel('time (s)')
plt.ylabel('Energy (Joules)')
#plt.legend()
ax=plt.gca() #Allows for axis limits
ax.set(xlim=(min(x), max(x)), ylim=(1.01*min(y), .85*max(y)))
#ax.get_yaxis().get_major_formatter().set_scientific(False) #turning off scientif notation
#ax.get_yaxis().get_major_formatter().set_scientific(False) #turning off scientific notation
plt.grid()
plt.show()
```



PS4_1.txt

```
31536.000000 -5335893862067229487687475618906112.000000
63072.000000 -5335680356336968100587734504046592.000000
94608.000000 -5335254658959023655706597865291776.000000
126144.000000 -5334616863530300932679839869042688.000000
157680.000000 -5333767097865913447019948950945792.000000
189216.000000 -5332705523966098061698425128222720.000000
220752.000000 -5331432337977192052979352050073600.000000
252288.000000 -5329947770146625360722554905427968.000000
283824.000000 -5328252084771994892603611385167872.000000
315360.000000 -5326345580144194365056109741342720.000000
346896.000000 -5324228588484645797796338217254912.000000
378432.000000 -5321901475876626897322882814181376.000000
409968.000000 -5319364642190720846585739352211456.000000
441504.000000 -5316618521004406947569013534752768.000000
```

[HW4-2]

Using similar algebra to prove Q = m(xvy – yvx). What is a better name for Q.

i. Below is a picture of my proof. A better name for Q would be angular momentum. It's rate of change would be the sum of the external torques acting on a body.

[PS 4.2]

$$\begin{array}{ccc} i & j & k \\ r_x & r_y & 0 \\ v_x & v_y & 0 \end{array}$$

$$L = (\vec{r} \times m\vec{v})$$

$$= m(xv_y - yv_x)$$

$$\frac{dL}{dt} = m\left(\frac{dx}{dt}v_y + x\frac{dv_y}{dt} - \frac{dy}{dt}v_x - y\frac{dv_x}{dt}\right)$$

$$\cancel{v_x v_y} + x a_y - \cancel{v_y v_x} - y a_x$$

$$a_y = -GM\frac{y}{r^3}$$

$$= -x\frac{GMy}{r^3} + \frac{yGMx}{r^3}$$

$$a_x = -\frac{GMx}{r^3}$$

$$= 0$$

[HW4-3]

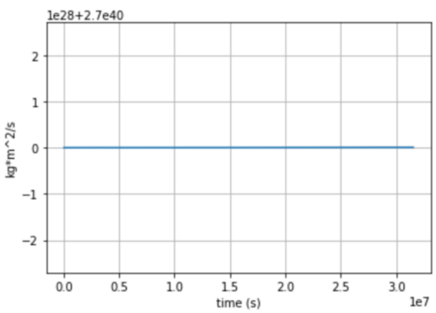
Discuss the variation in Q**.**

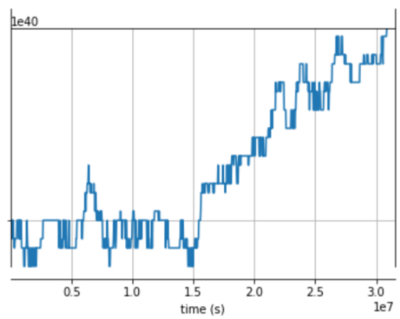
i. There is barely any variation in the numbers. It's a matter of 0.0000000000001791% which basically shows us that the torque is conserved. The image on the left hand side has the same scale as the previous graph for energy. The graph on the right is when I uncomment out the code below to set the axis limits thus making the scale VERY VERY small. The real depiction is on the left.

In [24]:
```
file_path3 = '/Users/Austin/Desktop/PHY 40/Homework/4/PS4_3.txt'

x, y = np.loadtxt(file_path3, unpack = True) #Method to extract/import information from txt file


plt.plot(x,y)
plt.xlabel('time (s)')
plt.ylabel('Energy (Joules)')
# ax1=plt.gca() #Allows for axis limits
# ax1.set(xlim=(min(x), max(x)), ylim=(min(y), max(y)))
plt.grid()
plt.show()
```





PS4_3.txt

```
31536.000000 27000000000000000941114858714372818075648.000000
63072.000000 26999999999999996105411580255856119250944.000000
94608.000000 27000000000000000941114858714372818075648.000000
126144.000000 26999999999999996105411580255856119250944.000000
157680.000000 27000000000000000941114858714372818075648.000000
189216.000000 27000000000000000941114858714372818075648.000000
220752.000000 27000000000000000941114858714372818075648.000000
252288.000000 26999999999999996105411580255856119250944.000000
283824.000000 26999999999999996105411580255856119250944.000000
315360.000000 26999999999999991269708301797339420426240.000000
346896.000000 26999999999999991269708301797339420426240.000000
378432.000000 26999999999999996105411580255856119250944.000000
409968.000000 26999999999999996105411580255856119250944.000000
```

[HW4-4]

What is the relation? What is the total energy?

i. This relationship tells us that the energy of the system is dependent on the value r. The total energy is = 0.5*mv^2 + (-GMm/r). However, considering it's a circular orbit, we know that the velocity of an orbiting body v = (Gm/2)^.5 which if we plug into the equation give us a total energy of E = = -GMm/2r.

[HW4-5]

What is the potential energy if this happens? What is always true of the sign of kinetic enrgy? Since energy is conserved, what must be true of the energy is able to escape the sun? In your homework last week what were the energies when you set vy = V/2, 2V? Does this explain your orbits?

i. The potential energy if this happens is approaching 0. The sign of kinetic energy is always known to be positive. Since Energy is conserved, if the a planet is to be able to escape the sun this would mean that the energy is positive and it has gone to infinity out of orbit. From the homework last week we could see from the graph that for V/2 we had an elliptical orbit and the for v2 it was a bit like a part of a parabola then a straight line. This

[HW4-6]

Explain carefully what the lines are doing

1.) `x[i+1]=x[i]+v[i]*dt;`

i. Firstly, x[i] is initialized with some value (out of for loop)as the very first index from the arrays x and v. This line then takes the value in x[i] adds it with v[i]*dt and then assigns the it's value to the next element in the array [i+1]. The next loop essentially takes the values from the x and v array that were *just* input, does the same thing and then assigs it to the next, next element until the condition in the loop is satisfied.

j. `a=-k*x[i+1]/m;`

i. This line takes the k input from the user multiples it by the value that was just calculated(x[i+1]) and then divides by the constant m and assigns it to the variable a.

`v[i+1]=v[i]+a*dt;`

i. This line is take the current element value in the v array, and adds it to the value of a from the preceding line multiplied by dt from the preceding line of code and then assigns it to the next element in the v array.

[PS4-7]

What do you get for the position and velocity? Explain why those values make sense.

i. For the first part, the value I get for position is 0.00900 and the velocity I get is -10.0000. As previously stated, this does makes sense. Given that we start of at the position x = 5, and seeing that the line for the acceleration is negative, which means that the velocity will be negative. In effect, this tells us that the value of x will be lower as result.

ii. You get an exit code 11 which indicates failure. I believe this is due to the fact that N = 40000 is larger than the number of arrays = 20000.

Lab 8 Problem Set

[PS4-6]

The outputs that I get for the from the function are:

i. 3.020
j. 3.00020
k. 3.000002

We can see that the values are coming closer to the value 3.0. Just like the first derivative which tell us the slop of a line, this tells us the slope of the slope.

[PS4-7]

The outputs that I get from the function are

i. 6.70952
j. 6.718705
k. 6.718286

These outputs don't seem to be getting closer to an integer but we can still see it becoming a more defined point. I did notice, however, for the first input for example the value was slightly different than what I got on my calculator which was 6.6 where as in [PS4-6] my value was exactly 3.2 on my calculator.