Physics 40: Laboratory Seven

Tuesday, April 21, 2020

Today's Goals: Further Kepler Thoughts; Arrays.

[1] Further Kepler Thoughts.

For the harmonic oscillator we discussed the conserved energy:

$$E = \frac{1}{2}kx^2 + \frac{1}{2}mv^2$$

The proof that E is conserved was a simple exercise in the chain rule from calculus, combined with using Hooke's Law $F = -kx$:

$$\begin{aligned}\frac{dE}{dt} &= kx\frac{dx}{dt} + mv\frac{dv}{dt} \\ &= kxv + mva \\ &= kxv + mv\frac{-kx}{m} \\ &= kxv - kxv = 0.\end{aligned}$$

Since $dE/dt = 0$, the energy is constant in time.

The proof of energy conservation for the Kepler problem is identical in spirit, just a bit more complex mathematically. Again, it is just the chain rule combined with the use of Newton's universal law of gravity, which gives the forces F_x and F_y (derived in class last Thursday):

$$F_x = -\frac{GMmx}{r^3} = ma_x \qquad\qquad F_y = -\frac{GMmy}{r^3} = ma_y$$

It is useful to do a little initial exercise looking at the time derivative of $r = \sqrt{x^2 + y^2}$:

$$\begin{aligned}\frac{dr}{dt} &= \frac{d}{dt}(x^2 + y^2)^{1/2} \\ &= \frac{1}{2}(x^2 + y^2)^{-1/2}\ \frac{d}{dt}(x^2 + y^2) \\ &= \frac{1}{2r}\left(2x\,\frac{dx}{dt} + 2y\,\frac{dy}{dt}\right) \\ &= \frac{1}{r}\left(x\,v_x + y\,v_y\right)\end{aligned}$$

You should go through the algebra above and understand it! It will help a lot in Physics 105, in addition to being good calculus practice.

With that little exercise out of the way, we can go after the energy:

$$E = -\frac{GMm}{r} + \frac{1}{2}m(v_x^2 + v_y^2)$$

The first term in the equation is the gravitational potential energy. The second is the kinetic energy. Taking the derivative with respect to time:

$$\begin{aligned}\frac{dE}{dt} &= \frac{GMm}{r^2}\frac{dr}{dt} + m\big(\, v_x\frac{dv_x}{dt} + v_y\frac{dv_y}{dt}\,\big) \\ &= \frac{GMm}{r^2}\frac{1}{r}\big(\, x\,v_x + y\,v_y\,\big) + m\big(\, v_x a_x + v_y a_y\,\big)\end{aligned}$$

The last line uses dr/dt from the previous page. The final step is to use the equations for a_x and a_y for gravity (also on the preceding page):

$$\frac{dE}{dt} = \frac{GMm}{r^3}\big(\, x\,v_x + y\,v_y\,\big) + m\big(\, v_x\frac{-GMx}{r^3} + v_y\frac{-GMy}{r^3}\,\big) = 0$$

So the energy in the formula at the top of the page is conserved!

[HW4-1] Add two lines to your molecular dynamics code for the Kepler problem to compute E and print it. Note that when you have very large numbers to print, you may wish to do so in scientific notation. Both floats and doubles can be printed using %e rather than %f or %lf. This line prints the time t, the position x, y, the energy E and Q, all in scientific notation:

```
printf("\n%e   %e   %e   %e   %e",t,x,y,E,Q);
```

Make a plot of E versus t. Will google sheets accept scienttic notation? If not, you can look at your output and see what the order of magnitude of the numbers is. If you see the energy is 4.56×10^{32}, then divide E by 10^{32} before printing it, so you code just prints 4.56. Then, on your plot, say somewhere that 'energy is in units of 10^{32} Joules'. This approach is basically the same philosophically as using special names for large numbers. We don't want to write 2309000000, so instead we say 2.309 trillion. The word 'trillion' tells us the 2.309 is in units of 10^9.

You will find that E is not precisely constant. Discuss whether the variations in E are significant or not. Hint: when a quantity is changing, it is ofetn useful to ask what the percent variation is. For example, if your tuition was \$4500 last quarter, and it increases by \$50, the percent variation is $100(50/4500) = 1.11\%$, about one part in one hundred. By how many percent is E varying in your MD code?

[HW4-2] Using similar algebra, prove that

$$Q = m\big(\, xv_y - yv_x\,\big)$$

is conserved. This is a much easier problem than proving energy conservation! What is a better name for Q?

[HW4-3] Add a line to your molecular dynamics code for the Kepler problem to compute Q. Make a plot of Q versus t. Discuss the variation in Q.

[HW4-4] In some mechanics class (Physics 9A if you were a freshman at UC Davis), you learned that for uniform circular motion the acceleration points toward the center of the circle and has magnitude

$$a = \frac{v^2}{r} \qquad \text{so that} \qquad F = \frac{mv^2}{r}$$

If you combine this with Newton's law of gravity you find that for a circular orbit (which is what most planets have, roughly)

$$\frac{GMm}{r^2} = \frac{mv^2}{r}$$

This tells you a really simple relation between the kinetic energy $mv^2/2$ and the potential energy $-GMm/r$. What is that relation? What is the total energy?

[HW4-5] For a planet to be able to escape the sun, by definition it must be able to achieve $r \to \infty$. What is the potential energy if that happens? What is always true of the sign of the kinetic energy? Since energy is conserved, what must be true of the energy if a planet is to be able to escape the sun? In your homework last week, what were the energies when you set $v_y = V/2$ and $v_y = 2V$? Does this explain what you observed in the orbits?

[2] We have been defining variables one at a time using statements like:

```
int j,N;
double dt,t=0.,x,y,Msun,Mearth,r,vx,vy,G;
double E,Lz;
```

at the top of our codes. Suppose you wanted to do something like keep track of the homework scores of $j = 1, 2, 3, \cdots 200$ students in a class using a C code. You could define 200 variables:

```
double hw001,hw002,hw003,hw004,hw005,hw006,hw007,hw008,hw009,hw010;
double hw011,hw012,hw013,hw014,hw015,hw016,hw017,hw018,hw019,hw020;
double hw021,hw022,hw023,hw024,hw025,hw026,hw027,hw028,hw029,hw030;
[...]
double hw191,hw192,hw193,hw194,hw195,hw196,hw197,hw198,hw199,hw200;
```

That would be pretty tedious!

Even worse, suppose your grading rubric was that the final grade was determined by the average of the homework score and the quiz score. You would need to define 200 more quiz variables and 200 more average variables, and then compute for each student:

```
ave001=(hw001+quiz001)/2.0;
ave002=(hw002+quiz002)/2.0;
ave003=(hw003+quiz003)/2.0;
[...]
ave200=(hw200+quiz200)/2.0;
```

That's 200 lines of code!

There has to be a better way!

The answer is to define an "array". You can think of an array as a trick to define 200 variables all at once. The line

```
double hw[200],quiz[200],ave[200];
```

tells the computer that there are 200 variables with names

```
hw[0],hw[1],hw[2], ... hw[199];
```

and similarly for quiz and ave.

Important: Note that C begins counting from zero, so the names begin hw[0] and end hw[199] rather than hw[1] and hw[200].

[3] Type in the following program:

```
#include <stdio.h>
#include <math.h>
int main(void)
{
    int i,x[20];
    for (i=0; i<20; i=i+1)
    {
        x[i]=i*i;
    }
    for (i=1; i<20; i=i+1)
    {
        printf("\n%6i %10i",i,x[i]);
    }
    fprintf(stdout,"\n");
    return 0;
}
```

This code defines the array x, and tells the computer it has twenty elements. Then, in the first loop, the code fills the element x[i] with the square of i. The second loop prints out the values of i and x[i]. Compile and run your code and see that it works.

[4] The program below puts all the values of factorial into an array

```
#include <stdio.h>
#include <math.h>
int main(void)
{
    int j;
    long int temp,fact[20];
    temp=1;
    for (j=1; j<20; j=j+1)
    {
        temp=temp*j;
        fact[j]=temp;
    }
    for (j=1; j<20; j=j+1)
    {
         printf("\n %10d %20ld",j,fact[j]);
    }
    printf("\n");
    return 0;
}
```

Type it in, compile and run it, and see that it works. The code illustrates another useful aspect of arrays. They allow us to save a bunch of values for future use instead of recomputing them all the time.

Important: Up to now when we used 'printf' for an integer we used %i. It is also okay to use %d, and I have done that above. To print a long int, one uses %ld. Remember too that the '10' in %10d tells the computer how much space to allocate for the number to be printed. This allows for prettier formatting. (Compare the output of the code above to one without the numbers to specify the space for printing.)

[5] As we have seen, to do molecular dynamics, you only need to keep track of the present value of the position and velocity and update them many times in a loop. But suppose for some reason you wanted to save the trajectory for all the time steps. Arrays let you do that. Type in this code, compile, and run it:

```
#include <stdio.h>
#include <math.h>
int main()
{
     double x[20000],v[20000],k,m,dt,a,t=0.;
     int i,N;
     printf("\nEnter k,m,dt,N   ");
     scanf("%lf %lf %lf %i",&k,&m,&dt,&N);
     x[0]=5.0;
     v[0]=0.0;
     for (i=0; i<N+1; i=i+1)
     {
          x[i+1]=x[i]+v[i]*dt;
          a=-k*x[i+1]/m;
          v[i+1]=v[i]+a*dt;
          t=t+dt;
      }
      printf("\nEnter a time step at which you want to know");
      printf("\nthe position and velocity:   ");
      scanf("%d",&i);
      printf("\n The position at step %5i is %8.4lf",i,x[i]);
      printf("\n The velocity at step %5i is %8.4lf\n ",i,v[i]);
      return 0;
}
```

As you can see, because the trajectory is stored in the array x[] you can go back and examine the position at any time step you want.

[**HW4-6**] Explain **carefully** what the lines

```
x[i+1]=x[i]+v[i]*dt;
a=-k*x[i+1]/m;
v[i+1]=v[i]+a*dt;
```

are doing.

[**HW4-7**] Run the code with these values:

```
Enter k,m,dt,N   2 .5 .001 10000
Enter a time step at which you want to know
the position and velocity:   785
```

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

```
Enter k,m,dt,N   2 .5 .001 40000
```

What happens? Why?