Physics 40: Laboratory Four

Thursday, April 9, 2019

Today's Goals: A few more integrals; Printing to a File; Molecular Dynamics.

- Overview of molecular dynamics.

The heart of molecular dynamics is to rewrite the definitions of velocity and acceleration

$$\begin{aligned} & v = \frac{dx}{dt} & \qquad a = \frac{dv}{dt} \\ \longrightarrow \quad & dx = v\,dt & \qquad dv = a\,dt \end{aligned}$$

If you combine these with Newton's second law, $F = ma$, you get a procedure to "update" the position and velocity for a brief time interval dt.

$$\begin{aligned} x &= x + v * dt; \\ a &= F/m; \\ v &= v + a * dt; \end{aligned}$$

Finally, if you insert that into a loop over N iterations, you can generate the trajectory a 'long' time $t = N\,dt$ into the future.

One nice thing is that all molecular dynamics codes look largely the same. What changes is the particular force F that is being studied.

As we will see in future labs, we can very easily generalize this procedure to do motion in two or three dimensions and even to consider more than one moving object. But for today we will stick to $d = 1$ and one object only.

Choosing the 'time step' dt is an important issue. You already have some insight into this from doing integrals numerically: In doing an integral, you needed to choose dx small enough to get accurate answers. How small depended on the integration scheme (rectangle or trapezoidal rule).

One can easily spend a week discussing the 'numerical analysis' behind the answer. Looking at different algorithms for molecular dynamics ('leapfrog' versus 'Euler') will be one possible final project for you. For now, let's just use these two, complementary, principles:

[**i**] Choose $dt/T \ll 1$, where T is the physical time scale in your problem. For a mass on a spring, the period $T = 2\pi\sqrt{m/k}$. A very safe choice is $dt = 10^{-3}\,T = 10^{-3}\,2\pi\sqrt{m/k}$. Notice the value of dt depends on k and m. If you have a really huge mass $m = 10000$ on a very weak spring, $k = 0.0001$, the period $T = 62800$ seconds, and $dt = 62.8$ seconds would be small! On the other hand, for a light mass $m = 0.0001$ on a very stiff spring, $k = 10000$, the period $T = 0.000628$ seconds, and $dt = 0.001$ seconds (which sounds 'small') would be far too large for accurate results!

[ii] It's very wise to try to figure out dt from a knowledge of the time scales T, but in hard problems you might not know what they are. So, often you simply figure out dt 'empirically': you run your molecular dynamics code for smaller and smaller dt until the resulting trajectories no longer change, just as you decreased dx (by increasing N) until your integrals stopped changing.

Back to **[i]**: A really important principle in physics (or in many aspects of science and life in general) is that the statement "x is small" should never be made. You should always say "x is small compared to X, where X is another object with the same units as X. Put another way, **never** say x is small. Always say x/X is small, where x/X is **dimensionless**.

Example: Is $x = \$10000$ a lot of money? It depends! $x = \$10000$ is a lot of money to a college student, because we can compare it to her financial aid check $X = \$5000$ and $x/X = 2$. But $x = \$10000$ is not a lot of money to Jim Walmart, whose net worth $X = \$48,400,000,000$ so that $x/X = 0.000000207$.

This dependence on the situation is analogous to our molecular dynamics example. Is $dt =$ 62.8 seconds small? Yes, if $m = 10000$ and $k = 0.0001$. But emphatically no if $m = 0.0001$ and $k = 10000$.

[PS-2-7] Modify your trapezoidal rule integration code from Lab Three to compute

$$\mathcal{I} = \int_0^3 \sqrt{9 - x^2}\, dx$$

Hint: All you need to do is modify one line of code!!!! Note: If you did not get the trapezoid code working, just use the rectangle rule code. What do you get for $N = 10$; $N = 100$; and $N = 1000$? Give your answers to five decimal places. Use one of your calculus procedures to get the exact result and compare. Sketch the integrand $\sqrt{9 - x^2}$ and explain why the exact answer is right from an elementary geometry fact you know (i.e without any calculus).

[PS-2-8] Modify your trapezoidal rule integration code from Lab Three to compute

$$\mathcal{I} = \int_1^4 \frac{x^{0.34}\, e^{0.25\,x} \sin x}{2.2 + 0.9\sqrt{x}}\, dx$$

Of course there is no way you can do this integral by any calculus procedure you learned. Is there any way you could possibly get a rough idea what the answer might be without a computer?

[1] So far, we have just sent the output of our codes directly to the screen. For more complex situations, with a lot of output, it is often more convenient to print to a file. Type in the following code:

```
#include <stdio.h>
#include <math.h>
int main(void)
{
     int j;
     FILE * fileout;
     fileout=fopen("Hermione","w");
     for (j=0; j<20; j=j+2)
     {
          fprintf(fileout,"\n%i",j);
     }
     fclose(fileout);
     return 0;
}
```

Run your code. It should create the file 'Hermione' which has the first ten even integers in it. Look for that file in your directory. Make sure it is there and contains the right output.

Important: For xcode on a mac, one needs to specify the full path for the directory where one would like the file. For example,

```
fopen("/Users/XYZ/Desktop/Hermione","w");
```

[2] Here is a molecular dynamics code for a mass on a spring. Notice it writes the trajectory (the value of position x for every time t) to the file 'Dumbledore.txt'.

```
#include <stdio.h>
#include <math.h>

int main()
{
     double x,v,k,m,dt,a,t=0.;
     int i,N;
     FILE * fileout;
     fileout=fopen("Dumbledore.txt","w");
     printf("\nEnter initial x,v   \n");
     scanf("%lf %lf",&x,&v);
     printf("\nEnter k,m,dt,N   ");
     scanf("%lf %lf %lf %i",&k,&m,&dt,&N);
     for (i=1; i<N+1; i=i+1)
     {
          x=x+v*dt;
          a=-k*x/m;
          v=v+a*dt;
          t=t+dt;
          fprintf(fileout,"\n%12.6lf %12.6lf",t,x);
      }
fclose(fileout);
return 0;
}
```

[PS 2-9] Run the code for initial $x = 5$, initial $v = 0$, spring constant $k = 8$, mass $m = 2$, time step $dt = 0.01$ and $N = 1000$ total steps. Open the file 'Dumbledore.txt'. At the top you should see:

```
0.010000     5.000000
0.020000     4.998000
0.030000     4.994001
0.040000     4.988004
0.050000     4.980012
0.060000     4.970028
0.070000     4.958056
```

(a) Why does this make sense?

Farther down the file you should encounter:

```
0.760000     0.303684
0.770000     0.203808
0.780000     0.103850
0.790000     0.003851
0.800000    -0.096150
0.810000    -0.196112
```

(b) Why does this make sense?

[3] Modify your code so that it prints out three columns of data instead: t, x, v so that you can also look at how the velocity v evolves.

[PS 2-10] Run your modified code for the same parameters as in PS2-9. Now at the top of the code you should see:

```
0.010000     5.000000    -0.200000
0.020000     4.998000    -0.399920
0.030000     4.994001    -0.599680
0.040000     4.988004    -0.799200
0.050000     4.980012    -0.998401
0.060000     4.970028    -1.197202
0.070000     4.958056    -1.395524
```

(a) Explain why the behavior of the velocity (the third column) makes sense.
Again, look farther down the file:

```
0.760000     0.303684    -9.987614
0.770000     0.203808    -9.995766
0.780000     0.103850    -9.999920
0.790000     0.003851   -10.000074
0.800000    -0.096150    -9.996228
0.810000    -0.196112    -9.988384
```

Why is the velocity $v = -10$ at these times?