

PHYSICS 115/242

Homework 4

Due in class, Monday, May 5.

Note: I'm giving you two weeks for this assignment because I expect you to be working on the molecular dynamics project at the same time.

You must explain your work.

1. *Physics 242 students only*

The Gaussian curve

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$$

figures extensively in statistics. You are *given* that the total area under the curve is unity, i.e.

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp(-x^2/2) dx = 1.$$

Find the value of a to 5 decimal places such that interval $-a$ to a contains *half* the total area, i.e.

$$\frac{1}{\sqrt{2\pi}} \int_{-a}^a \exp(-x^2/2) dx = \frac{1}{2}.$$

This involves two steps: (i) a code to do the integral, and (ii) a code to solve for a . You should write the code for the integral as a function $F[a]$ which can then be substituted into a root finding algorithm.

Hint: Keep things simple. I suggest trapezium rule or Simpson's rule for the integration and bisection for root finding.

2. Consider the *anharmonic* oscillator discussed in Qu. 6 of HW 3, which has energy

$$E = \frac{p^2}{2} + \frac{x^4}{4}.$$

Consider initial conditions $x(0) = 1, v(0) = 0$. for which you found the period T in Qu. 6 of HW3.

(a) *Physics 115 students only*

Consider both the second order Runge-Kutta (RK2) algorithm and the second order symplectic leapfrog (i.e. position or velocity Verlet) algorithm described in the hand-out. Determine the *maximum* deviation of the energy from its exact (constant) value for each of these methods for timestep $h = 0.02 \times T$ (where T is the period) for the following lengths of time:

- i. 1 period
- ii. 10 periods
- iii. 100 periods

Comment on your results.

(b) *Physics 242 students only*

Consider both the 4-th order Runge-Kutta algorithm and one of the 4-th order symplectic algorithms described in the handout. Determine the *maximum* deviation of the energy from its exact (constant) value for each of these methods for timestep $h = 0.02 \times T$ (where T is the period) for the following lengths of time:

- i. 1 period
- ii. 1000 periods

Comment on your results.

3. Consider a planet of unit mass moving in a plane under a gravitational potential

$$V(r) = -\frac{1}{r}.$$

(We have set the gravitational constant, G , and the mass of the sun at the origin to be unity.)

(a) Show *analytically* that the period of a circular orbit of radius unity is 2π .

Hint: First equate the centripetal force, v^2/r to the gravitational force, $1/r^2$, to show that the speed is given by $v = 1$.

(b) Starting with appropriate initial conditions for a circular orbit, show that the leapfrog (velocity or position Verlet) method reproduces this orbit provided the step size is not too large. Also show that the energy is conserved to a good approximation.

Note: In this question to show whether the numerically determined orbit agrees or disagrees with the analytical result you can either show the results graphically (if you have a plotting routine) or put the results in neat columns so the differences in the numbers can be seen quite easily. (Save paper by editing out many of the lines and only printing out some of them.) Later on, when we do the Mathematica part of the course we will see how to easily plot results of calculations.

Please present your results in polar coordinates (r, θ) . To get θ from (x, y) , use `atan2(y, x)`, rather than `atan(y/x)`. The former gives the result correctly in the interval $-\pi < \theta \leq \pi$, whereas the latter gives a result between $-\pi/2$ and $+\pi/2$.

(c) Now start with $x = 1, y = 0$ and $v_x = 0, v_y = 0.7$. Note that this is similar to the previous part but with a smaller initial speed. Determine the angular momentum, L , and energy E .

You are now *given* that the orbit is an ellipse with eccentricity, ϵ , related to the energy and angular momentum by

$$\epsilon = \sqrt{1 + 2EL^2}.$$

You are also given that the equation of the ellipse is

$$r = \frac{a(1 - \epsilon^2)}{1 - \epsilon \cos \theta},$$

where a is the semi-major axis, as in the handout discussed in class.

Note: you start the particle off at the aphelion Q , where $Q = a(1 + \epsilon)$ ($= 1$ here), and the eccentricity is given by $\epsilon = 1 - (v_i/v_{\text{circ}})^2$ where v_{circ} , the speed needed for a circular orbit, is equal to 1 as shown in 3a.

- (d) Show that the leapfrog (velocity or position verlet) method reproduces the expected elliptical curve provided that the step size is small enough, but that a smaller step size is needed than for the circular orbit in part (b) to get a comparable accuracy.

Note: For a highly elliptical orbit one should ideally use a method with adaptive step-size control, which will automatically reduce the stepsize when the planet approaches the sun where its momentum changes rapidly (not required for the course.)

- (e) The angular momentum (about the z -axis) is given by

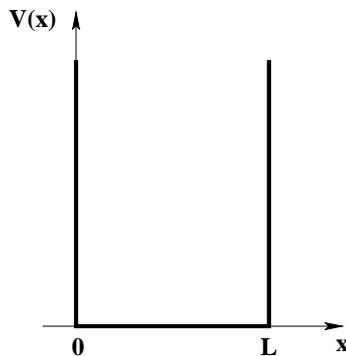
$$L_z = xp_y - yp_x,$$

where $\vec{p} = m\vec{v}$. Show *analytically* that the position (or velocity) Verlet method conserves angular momentum *exactly* for *any* central force. It is convenient to use vector notation, i.e. $\vec{L} = \vec{r} \times \vec{p}$, rather than using components. Verify numerically that your algorithm does conserve L_z apart from roundoff errors.

4. Schrödinger Equation

Consider the Schrödinger equation for a particle in an *infinitely high* potential well of width L , shown in the figure, where

$$V(x) = \begin{cases} 0 & (0 < x < L) \\ \infty & (\text{otherwise}) \end{cases} . \quad (1)$$



The boundary conditions are $\psi(0) = \psi(L) = 0$

- (a) Solve equation *analytically* and find the energy levels.
- (b) Taking $L = m = \hbar = 1$, use the 4th order Runge-Kutta method to integrate the equation from $x = 0$ (putting $\psi(0) = 0, \psi'(0) = 1$), to $x = 1$, and verify that $\psi(1) \simeq 0$ (the exact value is, of course, zero) if you use the lowest eigenvalue for the energy. I suggest you try several values for stepsize h , e.g. $h = 0.2, 0.1$ and 0.05 , and check convergence of the values of $\psi(1)$.

5. Sorting algorithm

(a) (*115 students only*)

- i. Using a random number generator, generate a list of 1000 random numbers in the range 0 to 1. (Don't include a printout of this.)
- ii. Write your own routine to sort the numbers in ascending order. Print out the first 5 and last 5 entries of the sorted list.
- iii. Using a bisection algorithm find between which two elements lies 0.7.
- iv. Compare the times it takes your routine to sort random lists of length 10000 and 20000. (Since simple sorting algorithms take a time of order N^2 you should find the time increases by roughly a factor of 4.)

(b) (*242 students only*)

- i. Using a random number generator, generate an array of 10^6 random numbers in the range 0 to 1 (don't include a printout of this!) and use the heapsort routine discussed in class, see also numerical recipes <http://www.nr.com>, to sort the numbers in ascending order. Print out the first 5 and last 5 entries of the sorted list.
- ii. Using a bisection algorithm to find between which two elements (of the sorted list) lies the value 0.7. (i.e. Your answer should be something like "0.7 lies between elements 699107 and 699108.")
- iii. Compare the times it takes your routine to sort random lists of length 10^6 and 10^7 . (Since heapsort takes a time of order $n \log_2 N$ you should find the time increases by not much more than a factor of 10.)

Note: One way to generate random numbers is to use the built-in C random number generator `random()`. This generates a random integer between 0 and `RAND_MAX`, where `RAND_MAX` is usually the largest integer, $2^{31} - 1$. Two things need to be done to make this routine useful here:

- (a) The generator has to be randomized using `srandom(seed)` where `seed` is an integer. Different values for `seed` will generate different strings of random numbers. A popular, choice, which gives a different value for `seed` each time, is to generate it from the `time()` command as follows:

```
srandom(time(NULL));
```

You need to include `#include <time.h>` in the preamble to your code for this to work.

- (b) You need to convert the random `int` coming from `random()` to a random `double` between 0 and 1. Here is one way to do that (in which `ONE` has been declared `double` set to have value 1):

```
x = random() / (RAND_MAX + ONE) ;
```

This include the value exactly 0 but excludes the value exactly 1.

6. *Least Squares Fit*

- (a) Perform a least squares straight line fit to the data in <http://young.physics.ucsc.edu/115/homework/data>. The first column is x and the second is y and the third is the error in y . There are $N = 200$ points. You should determine the parameters a and b in the fit $y = a + bx$ as well as the error bars in a and b .
- (b) Also determine the χ^2 per degree of freedom. Remember that this expected to be around 1 if the fitting function is a good model, the error bars are correct, and the number of points is large enough that the central limit theorem works well. *Note:* What is the number of degrees of freedom?
- (c) (*242 students only*) Determine the goodness of fit parameter Q defined in the class handout on least squares fitting.