# Physics 411: Homework 3

Because of the cancellation of class on January 28, this homework is a double-length homework covering two week's material, and you have two weeks to do it. It is due in class on **Thursday, February 13**.

1. **Quadratic equations:**

   (a) Write a program that takes as input three numbers, $a$, $b$, and $c$, and prints out the two solutions to the quadratic equation $ax^2 + bx + c = 0$ using the standard formula

   $$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

   Use your program to compute the solutions of $0.001x^2 + 1000x + 0.001 = 0$.

   (b) There is another way to write the solutions to a quadratic equation. Multiplying top and bottom of the solution above by $-b \mp \sqrt{b^2 - 4ac}$, show that the solutions can also be written as

   $$x = \frac{2c}{-b \mp \sqrt{b^2 - 4ac}}.$$

   Add further lines to your program to print these values in addition to the earlier ones and again use the program to solve $0.001x^2 + 1000x + 0.001 = 0$. What do you see? How do you explain it?

   (c) Using what you have learned, modify your program so that it calculates both roots of a quadratic equation accurately in all cases.

   ☑ **For full credit** turn in your answers to part (b), a copy of your final program, and a printout of it in action, showing the solution of the equation $0.001x^2 + 1000x + 0.001 = 0$.

   This is a good example of how computers don't always work the way you expect them to. If you simply apply the standard formula for the quadratic equation, the computer will sometimes get the answer wrong. In practice the method you have worked out here is the correct way to solve a quadratic equation on a computer, even though it's more complicated than the standard formula. If you were writing a program that involved solving many quadratic equations this method might be a good candidate for a user-defined function.

2. **Calculating derivatives:** Suppose we have a function $f(x)$ and we want to calculate its derivative at a point $x$. We can do that with pen and paper if we know the mathematical form of the function, or we can do it on the computer by making use of the definition of the derivative:

   $$\frac{\mathrm{d}f}{\mathrm{d}x} = \lim_{\delta \to 0} \frac{f(x+\delta) - f(x)}{\delta}.$$

   On the computer we can't actually take the limit as $\delta$ goes to zero, but we can get a reasonable approximation just by making $\delta$ small.

(a) Write a program that defines a function `f(x)` returning the value $x(x-1)$, then calculates and prints the derivative of the function at the point $x = 1$ using the formula above with $\delta = 10^{-2}$. Calculate the true value of the same derivative analytically and compare with the answer your program gives. The two will not agree perfectly. Why not?

(b) Repeat the calculation for $\delta = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}$, and $10^{-14}$. You should see that the accuracy of the calculation initially gets better as $\delta$ gets smaller, but then gets worse again. Why is this?

☑ **For full credit**, turn in a printout of your program, the results from the various calculations, and your answer to the question in part (b).

We will look at numerical derivatives in more detail later in the course, when we will study techniques for dealing with these issues.

3. **Heat capacity of a solid:** Debye's theory of solids gives the heat capacity of a solid at temperature $T$ to be

$$C_V = 9V\rho k_B \left(\frac{T}{\theta_D}\right)^3 \int_0^{\theta_D/T} \frac{x^4 e^x}{(e^x - 1)^2}\, dx,$$

where $V$ is the volume of the solid, $\rho$ is the number density of atoms, $k_B$ is Boltzmann's constant, and $\theta_D$ is the so-called *Debye temperature*, a property of solids that depends on their density and speed of sound.

(a) Write a Python function `cv(T)` that calculates $C_V$ for a given value of the temperature, for a sample consisting of 1000 cubic centimeters of solid aluminum, which has a number density of $\rho = 6.022 \times 10^{28}\,\mathrm{m}^{-3}$ and a Debye temperature of $\theta_D = 428\,\mathrm{K}$. Use the trapezoidal rule to evaluate the integral with $N = 1000$ sample points. Hint: The value of the integrand at $x = 0$ is zero.

(b) Use your function to make a graph of the heat capacity as a function of temperature from $T = 5\,\mathrm{K}$ to $T = 500\,\mathrm{K}$.

☑ **For full credit** turn in a printout of your program and the plot it produces.

4. **Simpson's rule:**

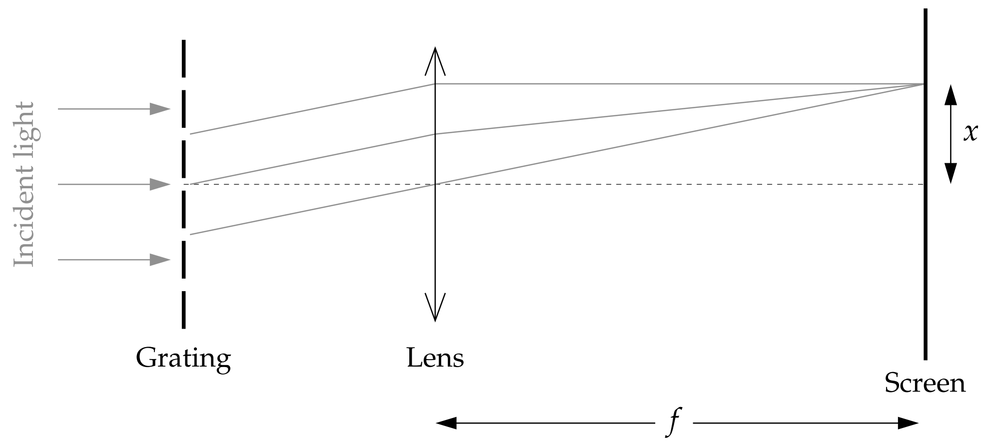
(a) Write a program to calculate an value for the integral $\int_0^2 (x^4 - 2x + 1)\, dx$ from Example 5.1, but using Simpson's rule with ten slices instead of the trapezoidal rule.

(b) Run the program and compare your result to the known correct value of 4.4. What is the fractional error on your calculation?

(c) Modify the program to use a hundred slices instead, then a thousand. Note the improvement in the result. How do the results compare with those from Example 5.1 for the trapezoidal rule with the same number of slices?

☑ **For full credit** turn in a printout of your program, plus your results and a brief discussion of how they compare with the trapezoidal rule.

5. **Diffraction gratings:** Light with wavelength $\lambda$ is incident on a diffraction grating of total width $w$, gets diffracted, is focused by a lens of focal length $f$, and falls on a screen:




Theory tells us that the intensity of the diffraction pattern on the screen, a distance $x$ from the central axis of the system, is given by

$$I(x) = \left| \int_{-w/2}^{w/2} \sqrt{q(u)}\, \mathrm{e}^{\mathrm{i}2\pi xu/\lambda f}\, \mathrm{d}u \right|^2,$$

where $q(u)$ is the intensity transmission function of the diffraction grating at a distance $u$ from the central axis.

(a) Consider a grating with transmission function $q(u) = \sin^2 \alpha u$. What is the separation of the "slits" in this grating, expressed in terms of $\alpha$?

(b) Write a Python function `q(u)` that returns the transmission function $q(u) = \sin^2 \alpha u$ as above at position $u$ for a grating whose slits have separation $20\,\mu$m.

(c) Use your function in a program to calculate and graph the intensity of the diffraction pattern produced by such a grating having ten slits in total, if the incident light has wavelength $\lambda = 500\,$nm. Assume the lens has a focal length of 1 meter and the screen is 10 cm wide. Use Simpson's rule for the integral. Decide for yourself how many sample points to use, so as to get an accurate answer. What criteria play into this decision?

Notice that the integrand in the equation for $I(x)$ is complex, so you will have to use complex variables in your program. As mentioned in Section 2.2.5 of the book, there is a version of the `math` package for use with complex variables called `cmath`. In particular you may find the `exp` function from `cmath` useful because it can calculate the exponentials of complex arguments.

(d) Now modify your program to create a visualization of how the diffraction pattern would look on the screen using a density plot. Your plot should look something like this:

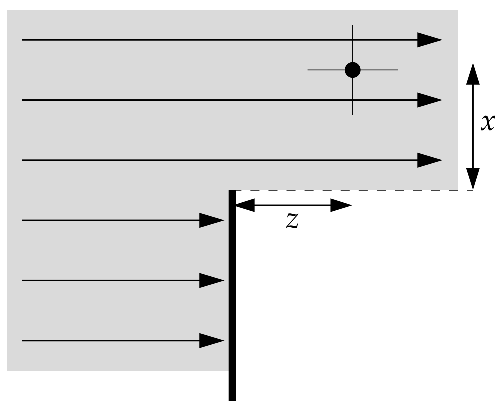
(e) Modify your program further to make pictures of the diffraction patterns produced by gratings with the following profiles:

   i. A transmission profile that obeys $q(u) = \sin^2 \alpha u \sin^2 \beta u$, with $\alpha$ as before and the same total grating width $w$, and $\beta = \frac{1}{2}\alpha$.

   ii. Two "square" slits, meaning slits with 100% transmission through the slit and 0% transmission everywhere else. Calculate the diffraction pattern for non-identical slits, one $10\,\mu$m wide and the other $20\,\mu$m wide, with a $60\,\mu$m gap between the two.

☑ **For full credit** turn in a printout of your program from part (d), your graph from part (c), and the three density plots from parts (d) and (e).

6. Suppose a plane wave of wavelength $\lambda$, such as a sound wave, is blocked by an object with a straight edge, represented by the solid line at the bottom of this figure:




The wave will be diffracted at the edge and the resulting intensity at the position $(x, z)$ marked by the dot is given by near-field diffraction theory to be

$$I = \frac{I_0}{8}\Big(\big[2C(u)+1\big]^2 + \big[2S(u)+1\big]^2\Big),$$

where $I_0$ is the intensity of the wave before diffraction and

$$u = x\sqrt{\frac{2}{\lambda z}}, \qquad C(u) = \int_0^u \cos \tfrac{1}{2}\pi t^2 \,\mathrm{d}t, \qquad S(u) = \int_0^u \sin \tfrac{1}{2}\pi t^2 \,\mathrm{d}t.$$

Write a program to calculate $I/I_0$ and make a plot of it as a function of $x$ in the range $-5$ m to 5 m for the case of a sound wave with wavelength $\lambda = 1$ m, measured $z = 3$ m past the straight edge. Calculate the integrals using any method of your choice. You should find significant variation in the intensity of the diffracted sound—enough that you could easily hear the effect if sound were diffracted, say, at the edge of a tall building.

☑ **For full credit** turn in a printout of your final program and the plot it produces.

7. Consider the integral

$$I = \int_0^1 \sin^2 \sqrt{100x}\, \mathrm{d}x$$

   (a) Write a program that uses the adaptive trapezoidal rule method described in Section 5.3 of the book—particularly Eq. (5.34)—to calculate the value of this integral to an approximate accuracy of $\epsilon = 10^{-6}$ (i.e., correct to six digits after the decimal point). Start with one single integration slice and work up from there to two, four, eight, and so forth. Have your program print out the number of slices, its estimate of the integral, and its estimate of the error on the integral, for each value of the number of slices $N$, until the target accuracy is reached. (Hint: You should find the result is around $I = 0.45$.)

   (b) Now modify your program to evaluate the same integral using Romberg integration. Have your program print out a triangular table of values, as on page 161 of the book, of all the Romberg estimates of the integral. Calculate the error on your estimates using Eq. (5.49) and again continue the calculation until you reach an accuracy of $\epsilon = 10^{-6}$. You should find that the Romberg method reaches the required accuracy considerably faster than the trapezoidal rule alone.

☑ **For full credit** turn in a copy of your final program from part (b) and printouts of the output of the programs from parts (a) and (b), showing that the Romberg method reaches the required accuracy in fewer steps than the adaptive trapezoidal method.

8. **A more advanced adaptive method for the trapezoidal rule:** In Problem 7(a) above you used the adaptive version of the trapezoidal rule in which the number of steps is increased—and the width $h$ of the slices correspondingly decreased—until the calculation gives a value for the integral accurate to some desired level. Although this method varies $h$, it still calculates the integral at any individual stage of the process using slices of equal width throughout the domain of integration. In this exercise we'll look at a more sophisticated form of the trapezoidal rule that uses different step sizes in different parts of the domain, which can be useful particularly for poorly behaved functions that vary rapidly in certain regions but not others. Remarkably, this method is not much more complicated to program than the ones we've already seen, if one knows the right tricks. Here's how the method works.

   Suppose we wish to evaluate the integral $I = \int_a^b f(x)\, \mathrm{d}x$ and we want an error of no more than $\epsilon$ on our answer. To put that another way, if we divide up the integral into $N$ slices of width $h$ then we require an accuracy per slice of

$$\frac{\epsilon}{N} = \frac{h}{b-a}\,\epsilon = h\delta,$$

   where $\delta = \epsilon/(b-a)$ is the target accuracy per unit interval.

   We start by evaluating the integral using the trapezoidal rule with just a single slice of width $h_1 = b - a$. Let us call the estimate of the integral from this calculation $I_1$. Usually $I_1$ will not be very accurate, but that doesn't matter. Next we make a second estimate $I_2$ of the integral, again using the trapezoidal rule but now with two slices of width $h_2 = \frac{1}{2}h_1$

each. Equation (5.28) tells us that the error on this second estimate is $\frac{1}{3}(I_2 - I_1)$ to leading order. If the absolute value of this error is smaller than the required accuracy $\epsilon$ then our calculation is complete and we need go no further. $I_2$ is a good enough estimate of the integral.

Most likely, however, this will not be the case; the accuracy will not be good enough. If so, then we divide the integration interval into two equal parts of size $\frac{1}{2}(b-a)$ each, and we repeat the process above in each part separately, calculating estimates $I_1$ and $I_2$ using one and two slices respectively, estimating the error, and checking to see if it is less than the required accuracy, which is now $\frac{1}{2}(b-a)\delta = \frac{1}{2}\epsilon$.

We keep on repeating this process, dividing each slice in half and in half again, as many times as necessary to achieve the desired accuracy in every slice. Different slices may be divided different numbers of times, and hence we may end up with different sized slices in different parts of the integration domain. The method automatically uses whatever size and number of slices is appropriate in each region.

(a) Write a program using this method to calculate the integral

$$I = \int_0^{10} \frac{\sin^2 x}{x^2}\,\mathrm{d}x,$$

to an accuracy of $\epsilon = 10^{-4}$. Start by writing a function to calculate the integrand $f(x) = (\sin^2 x)/x^2$. Note that the limiting value of the integrand at $x = 0$ is 1. You'll probably have to include this point as a special case using an if statement.

The best way to perform the integration itself is to make use of the technique of recursion, the ability of a Python function to call itself, which we looked at in Homework 2. Write a function `step(x1,x2,f1,f2)` that takes as arguments the beginning and end points $x_1, x_2$ of a slice and the values $f(x_1), f(x_2)$ of the integrand at those two points, and returns the value of the integral from $x_1$ to $x_2$. This function should evaluate the two estimates $I_1$ and $I_2$ of the integral from $x_1$ to $x_2$, calculated with one and two slices respectively, and the error $\frac{1}{3}(I_2 - I_1)$. If this error meets the target value, which is $(x_2 - x_1)\delta$, then the calculation is complete and the function simply returns the value $I_2$. If the error fails to meet the target, then the function calls itself, twice, to evaluate the integral separately on the first and second halves of the interval and returns the sum of the two results. (And then *those* functions can call themselves, and so forth, subdividing the integral as many times as necessary to reach the required accuracy.)

Hint: If you've done the calculation right, you should get a value of around 1.5 for the integral.

Further comment: As icing on the cake, when the error target is met and the function returns a value for the integral in the current slice, it can, in fact, return a slightly better value than the estimate $I_2$. Since you will already have calculated the value of the integrand $f(x)$ at $x_1$, $x_2$, and the midpoint $x_m = \frac{1}{2}(x_1 + x_2)$ in order to evaluate $I_2$, you can use those results to compute the improved Simpson's rule estimate, Eq. (5.7), for this slice. You just return the value $\frac{1}{6}h[f(x_1) + 4f(x_m) + f(x_2)]$ instead of the trapezoidal rule estimate $\frac{1}{4}h[f(x_1) + 2f(x_m) + f(x_2)]$ (where $h = x_2 - x_1$). This involves very little extra work, but gives a value that is more accurate by two orders in $h$. (Technically, this is an example of the method of "local extrapolation." We'll discuss local extrapolation again when we study adaptive methods for the solution of differential equations later in the semester.)

(b) Why does the function `step(x1,x2,f1,f2)` take not only the positions $x_1$ and $x_2$ as arguments, but also the values $f(x_1)$ and $f(x_2)$? Since we know the function $f(x)$, we could just calculate these values from $x_1$ and $x_2$. Nonetheless, it is a smart move to include the values of $f(x_1)$ and $f(x_2)$ as arguments to the function. Why?

☑ **For full credit** turn in your final program and a printout of it in action, showing clearly your final value for the integral.