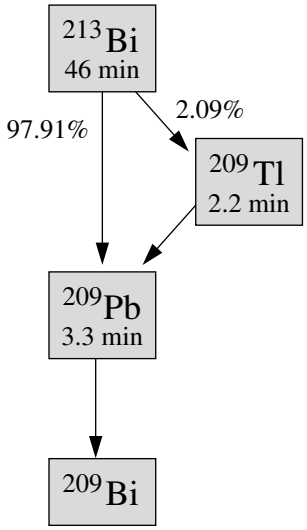
COMPUTATIONAL PHYSICS

EXERCISES FOR MONTE CARLO METHODS

1. **Radioactive decay chain**

   This exercise looks at a more advanced version of the simple radioactive decay we worked on in class. The isotope ^{213}Bi decays to stable ^{209}Bi via one of two different routes, with probabilities and half-lives as shown in below:



   (Technically, ^{209}Bi isn't really stable, but it has a half-life of more than 10^{19} years, a billion times the age of the universe, so it might as well be.)

   Starting with a sample consisting of 10 000 atoms of ^{213}Bi, simulate the decay of the atoms as in the code `decay.py` by dividing time into slices of length $\delta t = 1$ s each and on each step doing the following:

   (a) For each atom of ^{209}Pb in turn, decide at random, with the appropriate probability, whether it decays or not. (The probability can be calculated from $p(t) = 1 - 2^{-t/\tau}$) Count the total number that decay, subtract it from the number of ^{209}Pb atoms, and add it to the number of ^{209}Bi atoms.

   (b) Now do the same for ^{209}Tl, except that decaying atoms are subtracted from the total for ^{209}Tl and added to the total for ^{209}Pb.

   (c) For ^{213}Bi the situation is more complicated: when a ^{213}Bi atom decays you have to decide at random with the appropriate probability the route by which it decays. Count the numbers that decay by each route and add and subtract accordingly.

   Note that you have to work up the chain from the bottom like this, not down from the top, to avoid inadvertently making the same atom decay twice on a single step.

   Keep track of the number of atoms of each of the four isotopes at all times for 20 000 seconds and make a single graph showing the four numbers as a function of time on the same axes.

2. (a) Write a program to evaluate the integral

$$I = \int_0^2 \sin^2 \left[\frac{1}{x(2-x)} \right] dx$$

using the "hit-or-miss" Monte Carlo method, i.e.

$$I = \int_a^b f(x)dx \longrightarrow I \simeq \frac{kA}{N}$$

where k are the number of points below the curve defined by the integrant, N are the total number of random points and A is the area being integrated. Use 10 000 points.

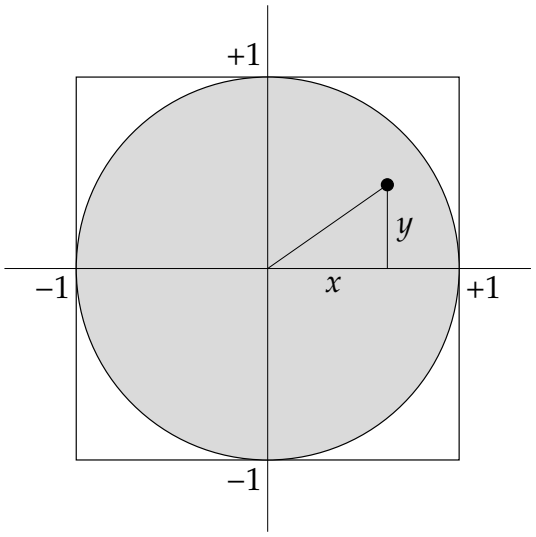
(b) Now estimate the integral again using the mean value method, i.e.

$$I = \int_a^b f(x)dx \longrightarrow I \simeq \frac{b-a}{N} \sum_{i=1}^{N} f(x_i)$$

with 10 000 points. Also evaluate the error.

3. **Volume of a hypersphere**

This exercise asks you to estimate the volume of a sphere of unit radius in ten dimensions using a Monte Carlo method. Consider the equivalent problem in two dimensions, the area of a circle of unit radius:



The area of the circle, the shaded area above, is given by the integral

$$I = \int\int_{-1}^{+1} f(x,y)\,\mathrm{d}x\,\mathrm{d}y,$$

where $f(x,y) = 1$ everywhere inside the circle and zero everywhere outside. In other words,

$$f(x,y) = \begin{cases} 1 & \text{if } x^2+y^2 \le 1, \\ 0 & \text{otherwise.} \end{cases}$$

So if we didn't already know the area of the circle, we could calculate it by Monte Carlo integration. We would generate a set of N random points (x,y), where both x and y are in the range from -1 to 1. Then the two-dimensional version of

$$I \simeq \frac{V}{N}\sum_{i=1}^{N} f(\mathbf{r}_i)$$

for this calculation would be

$$I \simeq \frac{4}{N}\sum_{i=1}^{N} f(x_i,y_i).$$

Generalize this method to the ten-dimensional case and write a program to perform a Monte Carlo calculation of the volume of a sphere of unit radius in ten dimensions.

If we had to do a ten-dimensional integral the traditional way, it would take a very long time. Even with only 100 points along each axis (which wouldn't give a very accurate result) we'd still have $100^{10} = 10^{20}$ points to sample, which is impossible on any computer. But using the Monte Carlo method we can get a pretty good result with a million points or so.

4. **Monte Carlo Simulation of an ideal gas**

   Starting with the coved `mcsim.py` modify it to do the following:

   (a) Start the with all the particles in the first excited state, i.e. $n_x = n_y = n_z = 2$. What is the value of the internal energy of the system?

   (b) Rewrite the original code so that you can include realistic values of the physical constants:

   $$k_B \quad m \quad \hbar \quad L$$

   however, do not include the correct values of these constants in the code but set then equal to 1.

   (c) Rewrite the original code to simulate a 2-dimensional gas.