Physics 3340/8300-005 - "Computational Physics" - Fall 2013
Homework 10
Assigned: Monday, 11 November
Due: Friday, 15 November 11:59pm by Blackboad upload

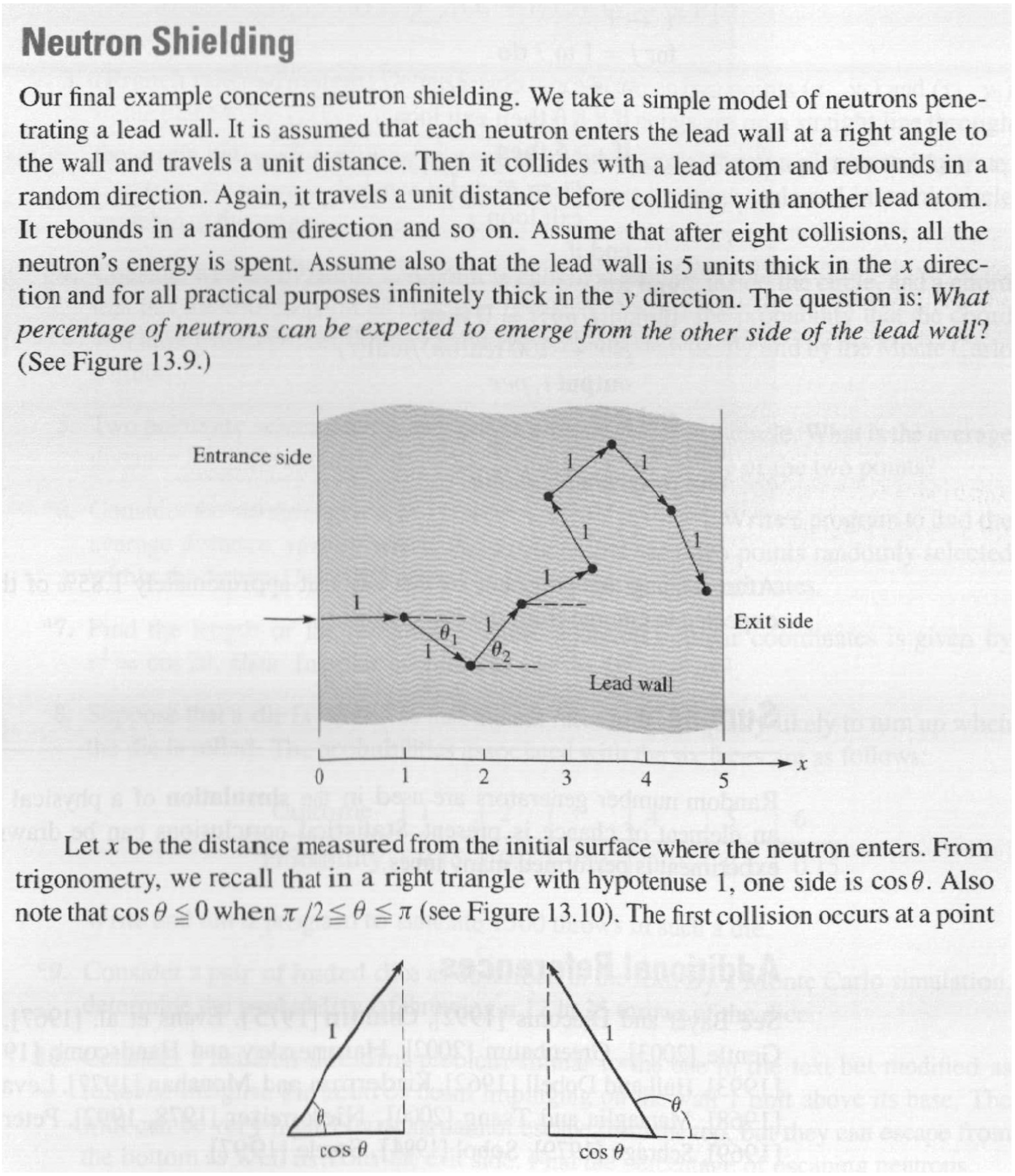
Reading:

Cheney and Kincaid sections 10.1, 10.2, 10.3 (6th ed. :13.1, 13.2, 13.3)
http://www.physics.smu.edu/fattarus/FittingLab.html
http://www.physics.smu.edu/fattarus/MonteCarloLab.html
And according to interest, http://en.wikipedia.org/wiki/Random_number_generation and http://en.wikipedia.org/wiki/Linear_congruential_generator

Problems:
For the following problems submit your solutions in your email report, with any C source code you write plus enough text or gnuplot graphics to trace the process of your work.

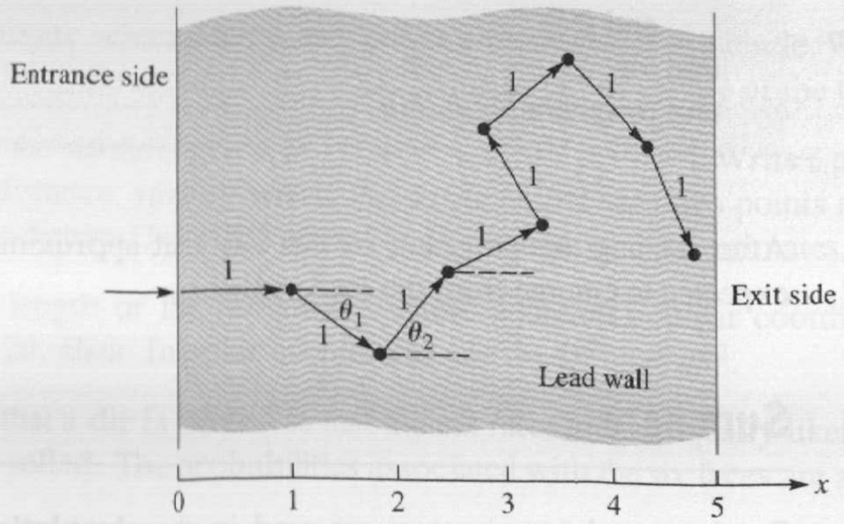
1) Simulate the fundamental process of radioactive decay with the Monte Carlo method as discussed in class. Write a C program that uses the routines in random.c from the Monte Carlo class lab. The program should use a printf() statement to output the decay rate for each time step so that you can capture the output into a data file and plot the decay rate versus time in gnuplot. Assume the decay probability λ is 0.1 for the material being simulated, and start with an initial undecayed population of 10000 nuclii. Then use the simulated data to find the best linear fit following the method in problem 1 of Homework 9. Hint: You will probably want to modify the printf() statement in your program to produce an output list tailored to the type needed by that fitting procedure. Report the value of λ that the fit routine finds, and compare it to the value of 0.1 that your modeling program used.

2) Simulate the neutron shielding problem from Cheney and Kincaid, section 13.3:

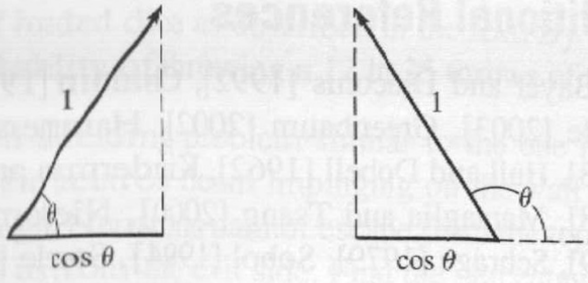


Neutron Shielding

Our final example concerns neutron shielding. We take a simple model of neutrons penetrating a lead wall. It is assumed that each neutron enters the lead wall at a right angle to the wall and travels a unit distance. Then it collides with a lead atom and rebounds in a random direction. Again, it travels a unit distance before colliding with another lead atom. It rebounds in a random direction and so on. Assume that after eight collisions, all the neutron's energy is spent. Assume also that the lead wall is 5 units thick in the x direction and for all practical purposes infinitely thick in the y direction. The question is: *What percentage of neutrons can be expected to emerge from the other side of the lead wall?* (See Figure 13.9.)



Let x be the distance measured from the initial surface where the neutron enters. From trigonometry, we recall that in a right triangle with hypotenuse 1, one side is $\cos\theta$. Also note that $\cos\theta \leqq 0$ when $\pi/2 \leqq \theta \leqq \pi$ (see Figure 13.10). The first collision occurs at a point

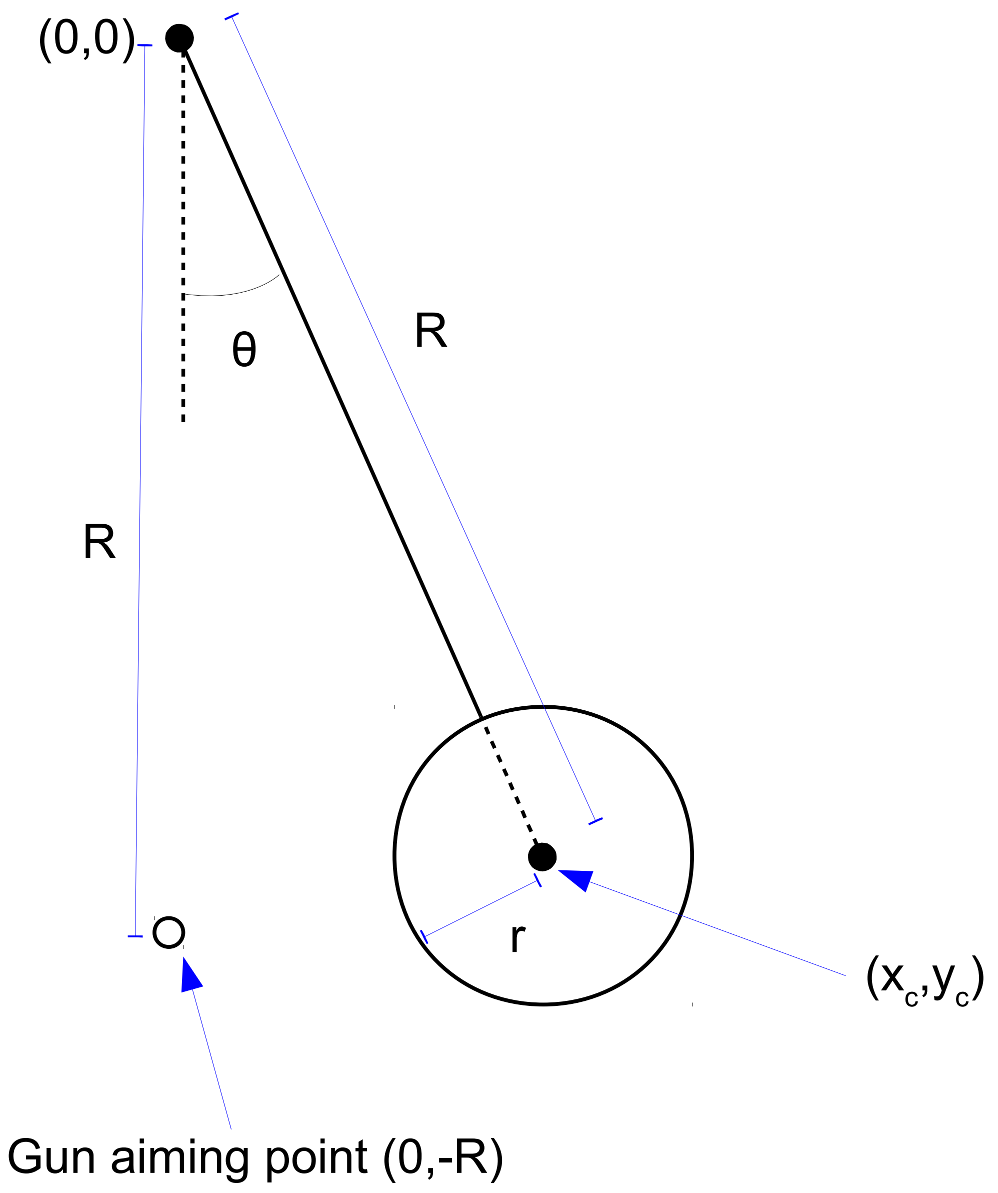


where $x = 1$. The second occurs at a point where $x = 1 + \cos\theta_1$. The third collision occurs at a point where $x = 1 + \cos\theta_1 + \cos\theta_2$, and so on. If $x \geqq 5$, the neutron has exited. If $x < 5$ for all eight collisions, the wall has shielded the area from that particular neutron. For a Monte Carlo simulation, we can use random angles θ_i in the interval $(0, \pi)$ because of symmetry. The simulation program then follows:

```
program Shielding
integer $i, j, m$;   real $x, per$;   real array $(r_{ij})_{1:n \times 1:7}$
integer $n \leftarrow 5000$, $iprt \leftarrow 1000$
$m \leftarrow 0$
call $Random((r_{ij}))$ for $i = 1$ to $n$ do
    $x \leftarrow 1$
    for $j = 1$ to $7$ do
        $x \leftarrow x + \cos(\pi r_{ij})$
        if $x \leqq 0$ then exit loop $j$
        if $x \geqq 5$ then
            $m \leftarrow m + 1$
            exit loop $j$
        end if
    end for
    if $\text{mod}(i, iprt) = 0$ then
        $per \leftarrow 100\,\text{real}(m)/\text{real}(i)$
        output $i, per$
    end if
end for
end program Shielding
```

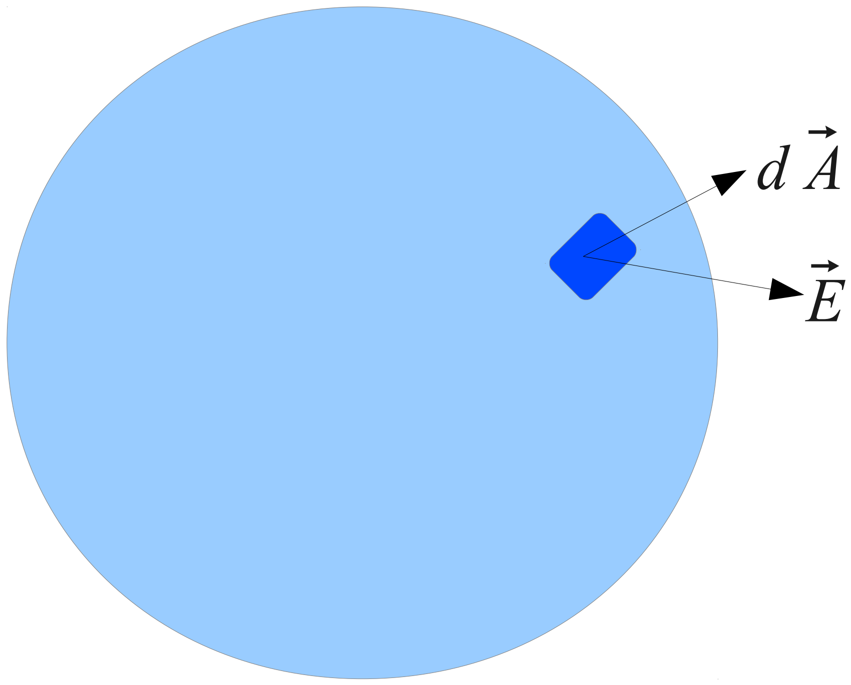
After running this program, we can say that approximately 1.85% of the neutrons can be expected to emerge from the lead wall.

3) You are on the Midway of the State Fair of Texas trying out a target shooting game. A circular disc target is mounted at the end of a pendulum arm, with the surface of the disc in the same vertical plane as the pendulum swinging motion. The pendulum swings back and forth with a maximum excursion θ_0 of 20 degrees each side of vertical. Assume for this small angular excursion that the angular motion is essentially sinusoidal, that is, $\theta=\theta_0 \sin(\omega t)$ where the angular frequency ω is 2π radians/sec. The distance from the pendulum pivot at coordinates (0,0) to the center of the target disc R=1 meter, and the radius of the disc r=0.1 meter.

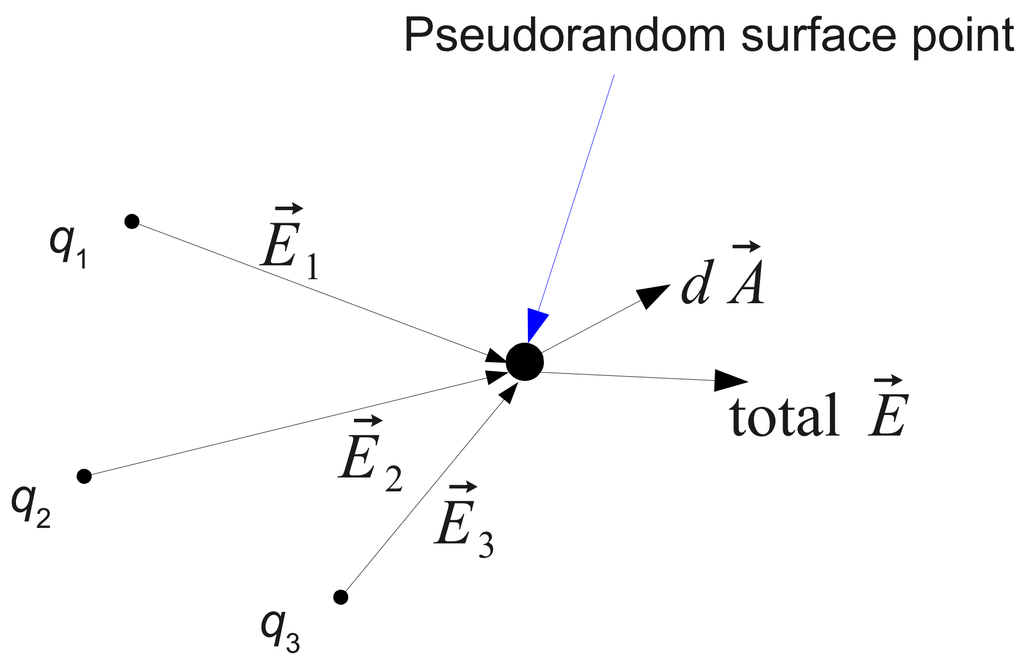


You aim the gun to fire exactly at the center of the disc as it passes by the exact bottom of its swing at a reference time of t=0. This aiming point would be at coordinates (0,-R). Each shot you take, however, is subject to a random timing error with a normal distribution. The mean is -t_0, exactly the negative time needed with the velocity of the bullet to hit the plane of the target at time t=0, so if there were no variance you would hit the target exactly in the center each time. But your shot timing has a standard deviation σ=0.07 seconds. (Note that the standard deviation is the square root of the variance!) Write a C program similar to the pi.c lab exercise that will calculate the probability of hitting the target.

4) Use the Monte Carlo method to verify Gauss' Law for electrostatic charges. Consider a spherical surface of radius 1 unit. Choose three arbitrary locations within the sphere to locate three electrostatic charges of arbitrary strengths. For example, you might choose one of your charges to be of 0.75 charge units at (x,y,z) coordinates of (0.5,0.2,0.6). For simplicity you can assume that in your arbitrary system of units, the value of permittivity is unity.



Use the random_gen() function to generate a large number of random points on the unit sphere. Use the sampling algorithm discussed in class for generating random points on a sphere. For each point on the sphere, calculate the dot product of $\vec{E} \cdot d\vec{A}$, where the **E** field is the sum of the fields from each of your three interior charges. Then form a sum of the dot product values over all your random points on the sphere, and use that to estimate the surface integral. Compare the value of the surface integral with the sum of the charge strengths you have included inside the sphere. Submit your source code and include your choice of charge locations and strengths. Hint: You probably will need quite a large number of random points to get a close correlation, like 10^6.


Pseudorandom surface point
q_1
$\vec{E}_1$
$d\vec{A}$
total $\vec{E}$
$\vec{E}_2$
q_2
$\vec{E}_3$
q_3