II. QUADRATURE: NEUTRON FLUX FROM A REACTOR

Physics 580, SDSU. Due 11:59 pm, Friday, September 30, 2011

Consider a reactor of height H, width W and depth D. Assume the reactor emits neutrons uniformly throughout its volume and each volume element emits uniformly in all directions. Assume the emission rate is 1 neutron per second per cubic meter.

Place a neutron detector a distance x_0 in front of the reactor, and a distance y_o along the front from one corner (by symmetry, which corner does not matter), as seen in Figure 1.

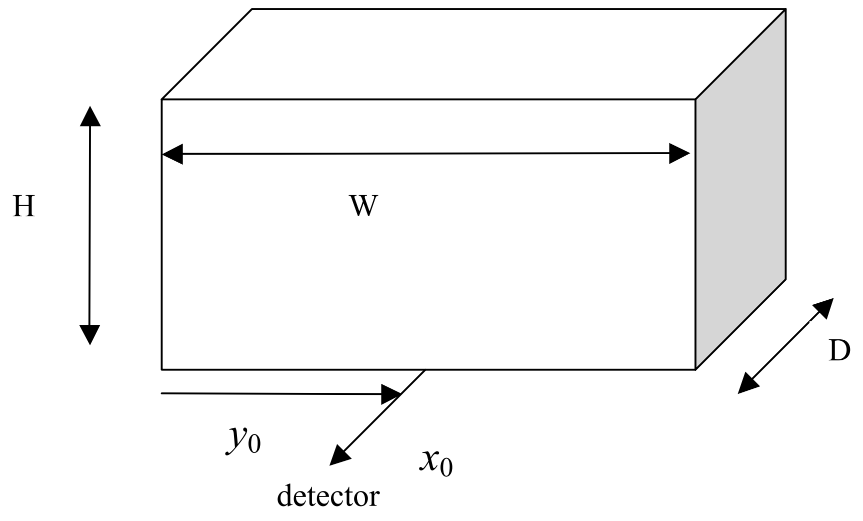

FIG. 1: A neutron-emitting reactor and detector.

A. Basic Project (worth 75%)

Input W, H and D and compute the total flux at position (x_0, y_0). Use Boole's rule. Note: user input needs to be W, H, D, x_0 and y_0, as well as the number of lattice points. These should not be hardwired into the code.

The flux of neutrons at a detector, ϕ_n, from a volume element, dV, a distance r away from the detector is

$$\phi_n = \lambda_n \frac{dV}{4\pi r^2}, \tag{1}$$

where λ_n is the neutron emission rate in neutrons per unit time per unit volume. For convenience, without loss of generality, we place the origin at the front corner of the reactor.

The flux is

$$\phi_n = 1\frac{\text{neutron}}{\text{s} \cdot \text{m}^3} \int_0^D dx \int_0^W dy \int_0^H \frac{dz}{4\pi} \left[(x+x_0)^2 + (y-y_0)^2 + z^2\right]^{-1} . \tag{2}$$

(In the future, you may not be given all this information, so be certain to understand how this result was derived. The + and − signs are important so be sure to understand those specifics.)

When writing a long program, it is a very good idea to break up the problem into several intermediate projects that can be tested.

1. Construct a one-dimensional quadrature routine using Boole's rule. Be sure the subroutine is appropriately documented. Be sure to put in an error trap to test that N, the number of points, is a multiple of 4. Test your subroutine against the Simpson's rule routine presented in class for some finite integral whose exact answer you know. Boole's rule should have a smaller error than Simpson's rule.

2. Write a master routine that reads in the information you need (*e.g.*, W, H, D, x_0 and y_0). Be sure to make it clear what each variable is. You will also need to ask for the number of lattice points in each direction. It is allowed (but not required) to use the same number of lattice points in each direction, even when W, H and D are not equal.

3. Write a subroutine that will do the three dimensional integral. To do the 3-D integral, write it as nested one-dimensional integrals. To wit,

   $$\phi_n = \int_0^D dx\ f(x),$$

   where

   $$f(x) = \int_0^W dy\ g(x,y),$$

   and where

   $$g(x,y) = \int_0^H \frac{dz}{4\pi} \left[(x+x_0)^2 + (y-y_0)^2 + z^2\right]^{-1} .$$

   In other words, your outermost loop is over x (it doesn't matter here in which order you integrate), then over y, then over z. The pseudo code looks like this:

```
loop over x
     loop over y
```

```
            loop over z
                compute, for x,y,z, $1/4\pi r^2$
                fill array for z-integral
            end loop over z
            pass array to Boole's rule to get g(x, y)
            fill array for y-integral
        end loop over y
        pass array to Boole's rule to get f(x)
        fill array for x-integral
    end loop over x
    pass array to Boole's rule to get flux
```

4. Testing. In order to test your program, one must think of a result that can be done analytically. In this case, for large x_0, when the detector is very far away from the source, you can approximate the reactor as a point source. If $x_0 \gg W, H, D$, then

$$\phi_n \approx \frac{WHD}{4\pi x_0^2},$$

where ϕ_n is in neutrons per second per square meter when W, H, D and x_0 are given in meters. Confirm that as x_0 becomes much larger than any of the spatial dimensions of the reactor, your calculated result is close to this estimate.

Remember, that for full credit your program should be well-commented and the input and output is clear and easy to use.

You should provide two output graphs,

- Illustrate the dependence on lattice spacing. Fix the spatial dimensions of the reactor (but not equal to each other) and the location of the detector (both non-zero). Use the same number of lattice points in each direction. Determine using regression (you may use the code provided on the course website), or estimate using a very large number of lattice points, the exact answer. Then vary the number of lattice points, again using the same number of points in each direction. Plot the absolute difference between the exact and numerical results as a function of N on a log-log plot. Be sure your graph includes your values for W, H, D, x_0 and y_0. Do you get what you expect?

  Extra credit (5%): use regression to determine the exact answer.

- Illustrate the $1/r^2$ law and deviations from it. Fix the spatial dimensions of the reactor (but not equal to each other) and y_0 (non-zero), but vary x_0. Plot both your answer and the approximate answer from a point source at the center of the reactor,

$$\frac{WHD}{4\pi}\left[\left(\frac{H}{2}\right)^2+\left(x_0+\frac{D}{2}\right)^2+\left(y_0-\frac{W}{2}\right)^2\right]^{-1}.$$

  Choose appropriate values of x_0 so that your graph clearly illustrates the limiting behaviors where the two curves approach each other and diverge.

B. Advanced project (worth 25%)

Use Monte Carlo integration to achieve the same ends. In this case, you do not break down the integral into 3 one-dimensional integrals, but simply sample points throughout the volume of the reactor. For full credit, you must compute the statistical uncertainty in the integral. For this, you may use the same master program as for your basic project, but write a different subroutine for the 3-D integration. For the Monte Carlo, you do not need to do any rejection techniques, simply generate uniformly-distributed random points within the volume of the reactor. You may use the subroutine `ranlib3` provided on the course webpage.

Add to the second graph (illustrate the inverse square law) the Monte Carlo results for at least five different values of x_0. Does the Monte Carlo integral and the deterministic integral (Basic project) agree within the error bars? Be sure to have the error bars for each point on the plot.

Submit the advanced project at the same time as the basic project. Either as a stand alone code or as a subroutine within the basic project.