Undergraduate Final Project

Due May 6, 11:59pm

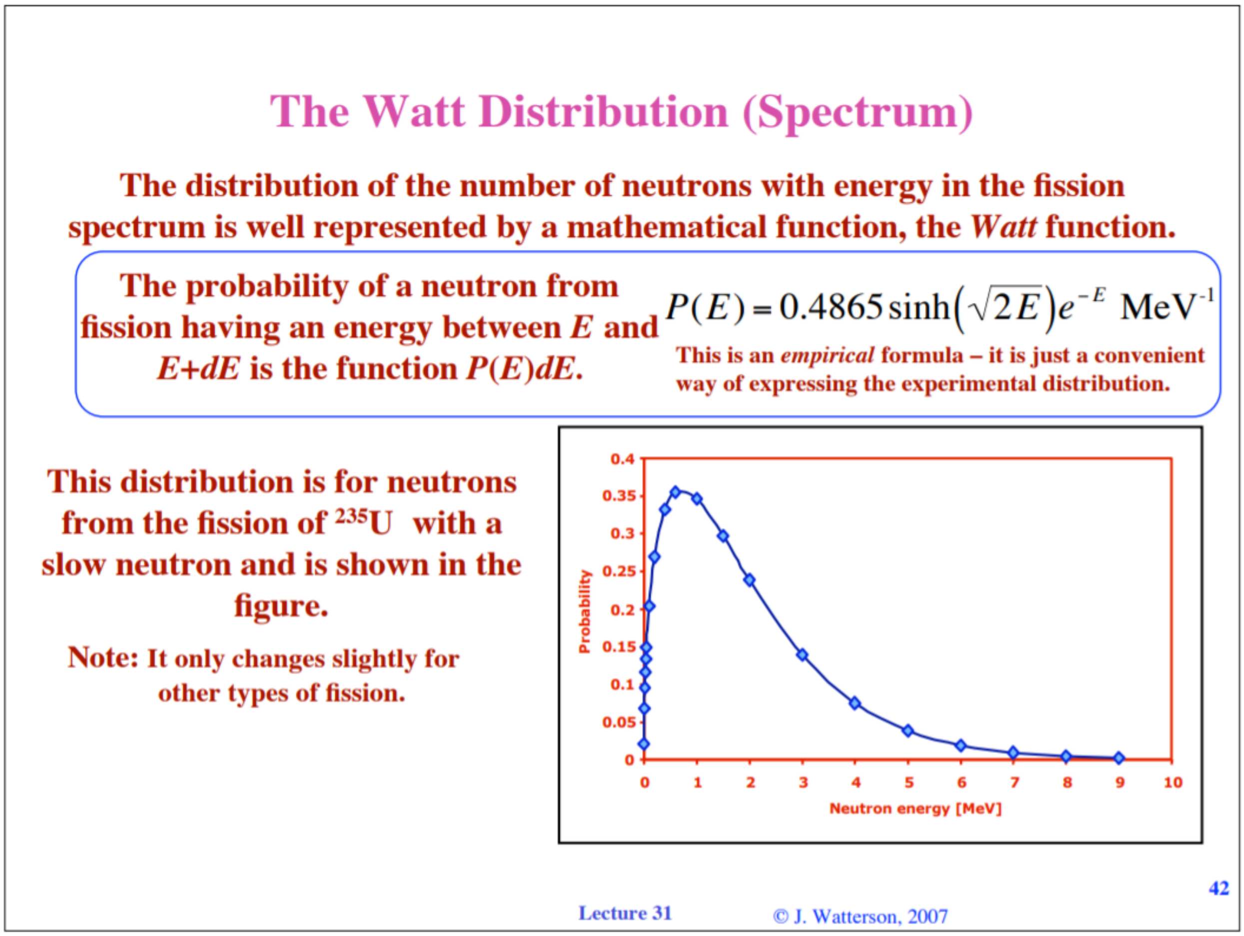
150 points

CS 4499/5531

Scientific Computing

Dr. Leslie Kerby

Write a C++ program that randomly samples from the Watt Fission Spectrum.



Say you have a Monte Carlo neutronics code. Part of this code will involve randomly sampling what energy a new neutron created from fission will have. This can be done using the Watt distribution. For this problem, we will assume the probability distribution of the energy is

$$P(E) \;=\; 0.4865\, sinh(\sqrt{2E})\, e^{-E}$$

Our plan will be to utilize rejection sampling as this is a powerful technique which can be used to randomly sample from essentially any distribution. The rejection method requires knowledge of the min/max and range, so we will calculate these along the way.

1. Set up f(x) (in this case *P(E)*).
    a. First create a function **fx** (either global or a lambda in main) which is passed **E** and returns the value of *P(E)* from the equation above.
    b. Check that you have **fx** correct by integrating, using an integration method from your HW 7, from [0,10] and showing your result is 1.00 (accurate to about the hundredth place). Print the integral to the screen.
2. Find the **maximum** of **fx** (*P(E)* **).**
    a. You can do this either by finding the roots of the derivative of **fx** (ie using **finite_difference_derivative** and **bisect**) and evaluating fx at the root, or by using **minima**. Print the (*E, P(E)*) for the maximum to the screen.
3. Create a random point.
    a. Randomly create a point in the range x=[0,8] and y=[0,*max from part 2*]. You will need to randomly select two values.
4. Use the rejection method.
    a. Check to see if the point selected in part 3 is under or equal to the *P(E)* curve. If it is, save the energy to a vector of doubles called **sample**. If it is not, reject the point and randomly create a new one and check again. Repeat until a valid point is found and the selected energy is stored.
5. Create a sample.
    a. Randomly sample 100,000 energies from the Watt Fission Spectrum. They should be stored in **sample**. Print the size of **sample** to the screen.
6. Binning.
    a. Create a vector of 100 energy bins, from 0 to 8 MeV, called **bins**. Find the number of energies in **sample** that are in each bin. Print to the screen the number of selections found in each of the 100 bins.
7. Histogram plot.
    a. Make sure your random sampler works by plotting a histogram of **sample,** by plotting the value in each bin of **bins**. You can use Excel, Python, C++, etc. If not in your C++ code, then attach the other file with the image of the plot.

Attach all relevant code and screenshots.