

Simulations of Particle Interactions with Matter.

Homework #1.

Roman Shapovalov

September 13, 2011

1 Maxwell Boltzmann

Given the Maxwell-Boltzmann Distribution:

$$N(v) = 4\pi \left(\frac{m}{2\pi kT}\right)^{3/2} v^2 e^{-\frac{mv^2}{2kT}} \quad (1)$$

1.1 Show $\langle v \rangle$

Show that

$$\langle v \rangle = 4\pi \left(\frac{m}{2\pi kT}\right)^{3/2} \left(\frac{2kT}{m}\right)^2 \frac{\Gamma(2)}{2} \quad (2)$$

Solution:

$$\langle v \rangle = \int_0^\infty v N(v) dv = 4\pi \left(\frac{m}{2\pi kT}\right)^{3/2} \int_0^\infty v^3 e^{-\frac{mv^2}{2kT}} dv \quad (3)$$

using table integral (http://en.wikipedia.org/wiki/Lists_of_integrals):

$$\int_0^\infty x^3 e^{-ax^2} dx = \frac{1}{2a^2} \quad (4)$$

when $a > 0$ we can calculate integral 3 as

$$\langle v \rangle = 4\pi \left(\frac{m}{2\pi kT}\right)^{3/2} \frac{1}{2 \left(\frac{m}{2kT}\right)^2} = 4\pi \left(\frac{m}{2\pi kT}\right)^{3/2} \left(\frac{2kT}{m}\right)^2 \frac{1}{2} \quad (5)$$

and using the fact that $\Gamma(2) = (2-1)! = 1$ finally

$$\langle v \rangle = 4\pi \left(\frac{m}{2\pi kT}\right)^{3/2} \left(\frac{2kT}{m}\right)^2 \frac{\Gamma(2)}{2} \quad (6)$$

that is exactly the equation 2 above. End of prove.

1.2 Energy Fluctuation (Grad)

Show that the energy fluctuation is

$$\frac{1}{4} m^2 \langle (v^2 - \langle v^2 \rangle)^2 \rangle = \frac{3}{2} (kT)^2 \quad (7)$$

Solution:

$$\frac{m^2}{4} \langle (v^2 - \langle v^2 \rangle)^2 \rangle = \frac{m^2}{4} (\langle v^4 \rangle - 2 \langle v^2 \rangle \langle v^2 \rangle + \langle v^2 \rangle^2) = \frac{m^2}{4} (\langle v^4 \rangle - \langle v^2 \rangle^2) \quad (8)$$

Using table integral (http://en.wikipedia.org/wiki/Lists_of_integrals):

$$\int_0^{\infty} x^{2n} e^{-ax^2} dx = \frac{(2n-1)!!}{2^{n+1}} \sqrt{\frac{\pi}{a^{2n+1}}} \quad (9)$$

we can calculate:

$$\langle v^2 \rangle = \int_0^{\infty} v^2 N(v) dv = 4\pi \left(\frac{m}{2\pi kT} \right)^{3/2} \int_0^{\infty} v^4 e^{-\frac{mv^2}{2kT}} dv \quad (10)$$

as

$$\langle v^2 \rangle = 4\pi \left(\frac{m}{2\pi kT} \right)^{3/2} \frac{3!!}{2^3} \sqrt{\frac{\pi}{\left(\frac{m}{2kT}\right)^5}} = \frac{3}{2} \left(\frac{m}{2kT} \right)^{3/2} \left(\frac{2kT}{m} \right)^{5/2} = \frac{3}{2} \left(\frac{2kT}{m} \right) \quad (11)$$

Again using the table integral 9 we can calculate:

$$\langle v^4 \rangle = \int_0^{\infty} v^4 N(v) dv = 4\pi \left(\frac{m}{2\pi kT} \right)^{3/2} \int_0^{\infty} v^6 e^{-\frac{mv^2}{2kT}} dv \quad (12)$$

as

$$\langle v^4 \rangle = 4\pi \left(\frac{m}{2\pi kT} \right)^{3/2} \frac{5!!}{2^4} \sqrt{\frac{\pi}{\left(\frac{m}{2kT}\right)^7}} = \frac{15}{4} \left(\frac{m}{2kT} \right)^{3/2} \left(\frac{2kT}{m} \right)^{7/2} = \frac{15}{4} \left(\frac{2kT}{m} \right)^2 \quad (13)$$

Using equation 8 above

$$\frac{m^2}{4} \langle (v^2 - \langle v^2 \rangle)^2 \rangle = \frac{m^2}{4} (\langle v^4 \rangle - \langle v^2 \rangle^2) = \frac{m^2}{4} \left(\frac{15}{4} \left(\frac{2kT}{m} \right)^2 - \left(\frac{3}{2} \left(\frac{2kT}{m} \right) \right)^2 \right) = \frac{3}{2} \frac{kT}{m} \quad (14)$$

that is exactly the equation 7 above. End of prove.

2 MC calculation of π

Calculate π using the Monte Carlo method described in the Notes: https://wiki.iac.isu.edu/index.php/Simulations_of_Particle_Interactions_with_Matter#Example_2_Calculation_of_.CF.80

Solution:

Below is the script (Roman_Pi.C) for Monte Carlo Pi calculation:

```
//This program to calculate the Pi number
//using Monte-Carlo method
//written for g++ compiler

//Created May 31, 2011 by Roman S.
//Modified Sep 12, 2011

using namespace std;
#include <iostream>
#include <iomanip>
#include <fstream>
#include <ctime>
#include <cstdlib>
#include <cmath>

main()
{

    fprintf(stderr, "\n*****Welcome to C++ program to calculate Pi using Monte Carlo method*****\n");
    fprintf(stderr, "*****Created May 31, 2011 by Roman S.\n");
    fprintf(stderr, "*****Modified Sep 12, 2011\n\n");

    double x,y;
    int i;
    double hit_max;
    double hit_circle;

    hit_max = 10;
    hit_circle = 0;
    i = 0;

    srand((unsigned)time(0));

    while(i<hit_max){
        x = double(rand())/double(RAND_MAX);
        y = double(rand())/double(RAND_MAX);
        if (y<=sqrt(1-pow(x,2)))
            hit_circle++;

        fprintf(stdout, "%g\t%g\n", x,y);
        i++;
    }

    fprintf(stderr, "\nhit_circle=%g\thit_max=%g\tPi=%g\n", hit_circle, hit_max, 4*hit_circle/hit_max);
}
```

To compile this script I have used the g++ compiler with the following command:

```
g++ Roman_Pi.C -o Roman_Pi
```

Below is the screenshot of the results of running the program. The `hit_max = 10` here to be able to fit the results into screen. I will re-run the program later with the `hit_max = 10,000` to more accurately calculate the Pi.

```
roman@roman-laptop:~/Classes_Manuals/Phys599_NucSim/Homework/hw1/MonteCarlo_Pi$ ./Roman_Pi
*****Welcome to C++ program to calculate Pi using Monte Carlo method*****
*****Created May 31, 2011 by Roman S.
*****Modified Sep 12, 2011

0.864968      0.102693
0.136565      0.143102
0.140669      0.284003
0.448752      0.208196
0.343608      0.861973
0.793586      0.467476
0.323104      0.120766
0.953589      0.759857
0.00992611    0.970646
0.728003      0.703635

hit_circle=8   hit=10   Pi=3.2
```

3 Histograms using ROOT

To create a file with the 2 columns of random numbers I re-run my program as following:

```
./Roman_Pi > data.txt
```

The program here was recompiled with `hit_max = 10,000` and below the screenshot of results:

```
roman@roman-laptop:~/Classes_Manuals/Phys599_NucSim/Homework/hw1/MonteCarlo_Pi$ ./Roman_Pi > data.txt
*****Welcome to C++ program to calculate Pi using Monte Carlo method*****
*****Created May 31, 2011 by Roman S.
*****Modified Sep 12, 2011

hit_circle=7830 hit=10000      Pi=3.132
```

Now I did the following from the command line:

- `root` (to start root). The following inside the root.
- `root [0] .L ascii2root.C`
- `root [1] ascii2nt(''data.txt'')`
- `root [2] new TBrowser();`
- inside the opened window found the `rns.root` file and navigated to the leaf "rnd1" and "rnd2" and tried the following command to manipulate with histogram:
- `rns->Draw(''rnd1*4'');` see figure 1
- `rns->Draw(''rnd1:rnd1'', ''rnd1>0.5'');` see figure 2
- `rns->Draw(''rnd1:rnd2'');` see figure 3
- `rns->Draw(''asin(rnd1)'');` see figure 4
- `rns->Draw(''asin(rnd1-rnd2)'');` see figure 5. It does look like a Normal/Gaussian Distribution.

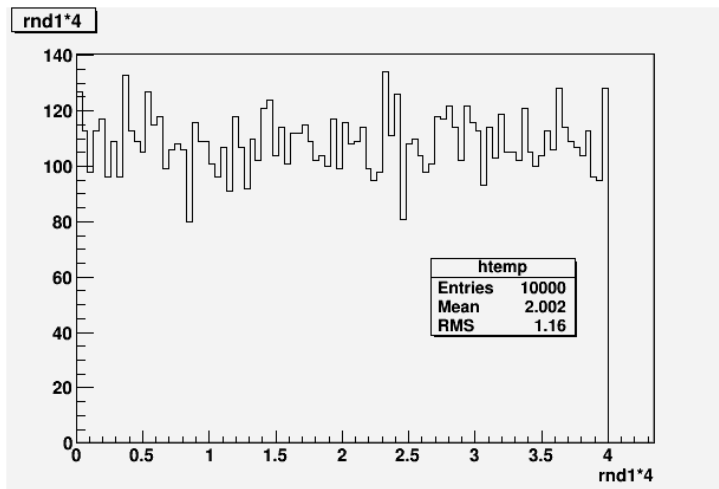


Figure 1: Histogram of rnd1 multiplied by 4

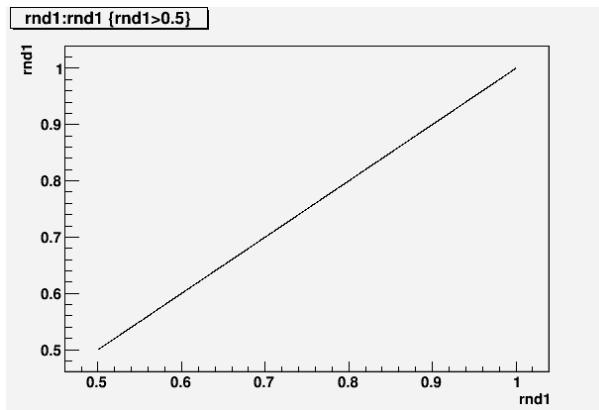


Figure 2: Convolution of rnd1 and rnd1 for $\text{rnd1} > 0.5$

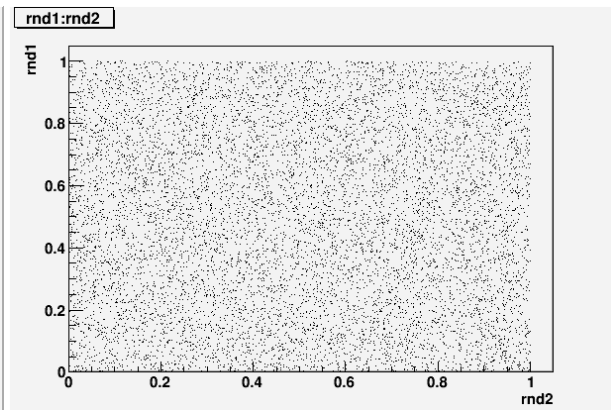


Figure 3: Convolution of rnd1 and rnd2

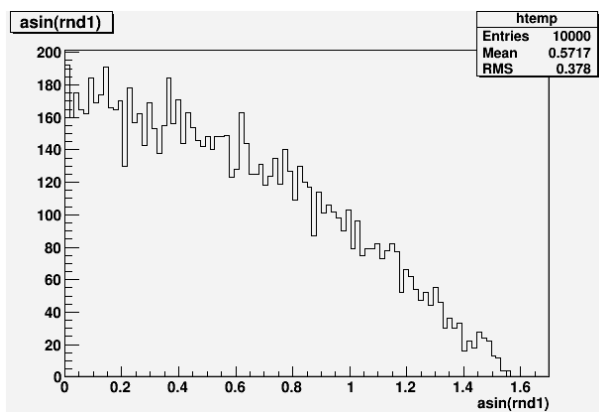


Figure 4: Histogram of asin(rnd1)

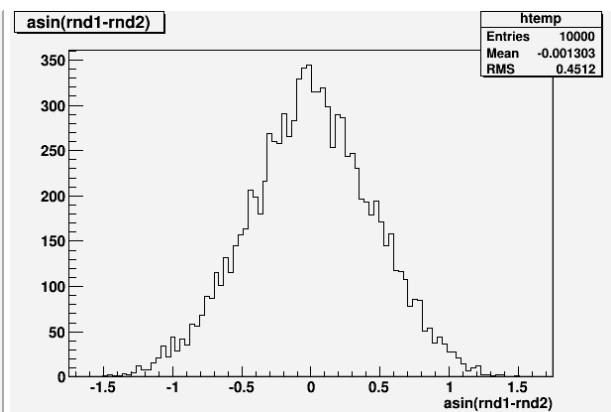


Figure 5: Histogram of asin(rnd1-rnd2)