Simulations of Particle Interactions with Matter

a course on Monte Carlo Simulations of common processes associated with radiation detection and measurement

I overview

A) Particle detection

Basically a device detects a particle only after the particle transfers energy to the device.

Energy intrinsic to a device
depends on the material used in a device.

Some device material of average atomic number (Z) is at some temperature (T). The materials atoms are in constant thermal motion (unless $T = 0°K$)

Statistical Thermodynamics tells us that the canonical energy distribution of the atoms is given by Maxwell-Boltzmann statistics such that

$$P(E) = \frac{e^{-E/kT}}{kT}$$ = Probability of any atom in the system having an energy "E"

I Overview

A.) particle detection

$$P(E) = \frac{e^{-E/kT}}{kT} \quad \text{where} \quad k = 1.38\times10^{-23}\ \frac{J}{mol-{}^{\circ}K}$$

= Boltzmann's Constant

<u>note</u>: you may be more familiar ~~wt~~ with The Maxwell - Boltzmann distribution in the form

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

$\Rightarrow$ $N(v)\,\Delta v$ = # of molecules in gas sample with speeds between v and $v+dv$

Example 1.) What is the probability that an atom in a 12.011 gram block of carbon ~~at~~ would have an energy of 5eV?

$\Rightarrow$ $P(5ev) = ?$

First check that probability distribution is normalized

ie: does $\int_0^\infty P(E)dE \stackrel{?}{=} 1$?

$$\int_0^\infty P(E)\,dE = \int_0^\infty \frac{e^{-E/kT}}{kT} = \frac{1}{kT}\left(\frac{1}{-1/kT}\right) e^{-E/kT}\Big|_0^\infty$$

$$= -\left[e^{-\infty} - e^{0}\right] = 1 \quad \checkmark$$

I overview

A.) particle detection

example 1.) $P(5ev) = ?$

$P(5ev)$ is calculated by integrating $P(E)$ over some energy interval (ie: $N(v)\,dv$), I will arbitrarily choose $4.9\ ev \rightarrow 5.1\ ev$ ~~just~~ as a starting point

$$\int_{4.9}^{5.1ev} P(E)\,dE = -\left[e^{-5.1ev/kT} - e^{-4.9ev/kT} \right]$$

$$k = 1.38 \times 10^{-23} \frac{J}{mole - {}^{\circ}K} \left(6.242 \times 10^{18} \frac{ev}{J} \right) = 8.614 \times 10^{-5} \frac{ev}{mole - {}^{\circ}K}$$

assume room temperature of $T = 300^{\circ}K$

Then $kT = .0258 \frac{ev}{mole}$

$$\int_{4.9}^{5.1ev} P(E)\,dE = e^{-4.9/.0258} - e^{-5.1/.0258}$$
$$= 4.48 \times 10^{-83} - \underbrace{1.9 \times 10^{-86}}$$

This is 1000 times smaller ∴ negligable

$$\therefore\ P(5ev) = e^{-\cancel{5ev}/.0258} \approx 10^{-85}$$

Since we have 12.011 grams of carbon and
1 mole of carbon = 12.011 gram = 6×10^{23} carbon atom.

Then ~~P(5ev)~~ probability of finding an atom in a sample size of 6×10^{23} carbon atoms is very small!

$M_E \sim 10^{27} g \sim 10^{50}$ atoms

I overvrew

A) particle detection

example 1.)

In other words; if a carbon atom in a block of carbon absorbs 5eV of energy and is detected it would be very noticeable compared to the typical energy of a carbon atom in the block.

(signal well above noise!)

silicon detectors & ionization chambers are two commonly used devices for detecting radiation

~1eV of energy is typically needed to create an electron-ion pair in Silicon

$$P(1eV) \approx e^{-\frac{1eV}{.0258}} \approx 10^{-17}$$

~10 eV of energy in needed to ionize an atom in a gas chamber

$$P(10eV) \approx 10^{-169}$$

$\Rightarrow$ ~~any radiation which~~ the SNR for detecting 10eV radiation will be better in ionization chamber than silicon, below that energy you need silicon and a low noise detector.

I overview

A.) Particle Detection

One trick with silicon is to lower the temperature of the bulk material thereby decreasing the intrinsic noise

If $T = 200^\circ K$ then

$$P(\text{1 eV}) = e^{-1/kT} = 10^{-26} \text{ instead of } 10^{-17} \text{ at } T = 300^\circ K$$

~~Another is to~~

also, If the radiation flux is large, more electron-hole pairs are created and thus the signal increases unfortunately silicon can be damaged by radiation.

B.) The monte carlo method

Def.

stochastic: from the greek work "stachos", a means of, relating to, or characterized by conjecture and randomness.

A stochastic process is one whose behavior is non-deterministic in that the next state is ~~only~~ partially determined.

~~In other words if you can't~~

Physics has many such non-deterministic systems (Quantum Mechanics, Thermodynamics, ...)

I overview

B.) The monte-carlo method

Basically the monte-carlo method here uses a random number generator (RNG) to generate a distribution (uniform, gaussian, ...) which is used to solve a stochastic process base on an astochastic description

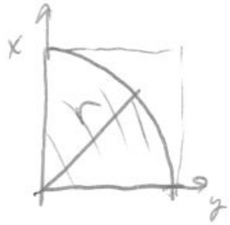
Example 1.) calculation of PI

astochastic description : PI may be measured as the ratio of the area of a circle of radius "r" divided by the Area of a square of length "2r".

$$\frac{A_{circle}}{A_{square}} = \frac{\pi r^2}{4r^2} = \frac{\pi}{4}$$

physically measure the area ~~ratio~~ ratios and you have π

stochastic description: treat the area like a dart board and throw random darts at it.



$$\frac{A_{1/4\ circle}}{A_{square}} = \frac{\frac{1}{4}\pi r^2}{r^2} = \frac{\pi}{4}$$

I overview

B.) Monte carlo method

Example 1.)

M.C. method: This is an outline of a program which would ~~give~~ calculate π

```
begin loop
      x = rnd                  (0 ≤ x ≤ 1)
      y = rnd
      dist = sqrt(x^2 + y^2)
      if dist ≤ 1.0  then  numbCircHits += 1.0
      numbSquareHits += 1.0
end loop
      π = 4 * numbCircHits / numbSquareHits
```

$$\pi = \frac{4 * numbCircHits}{numbSquareHits}$$

C.) UNIX Primer

To get our feet wet using the UNIX operating system we will try to solve example 1.) above using a RNG under unix.

Quick list of important commands:

1.) ls
2.) pwd
3.) cd
4.) df
5.) ssh
6.) scp
7.) mkdir
8.) printenv
9.) emacs, vim
10.) make, gcc
11.) man
12.) less
⋮
9999.) rm

I overview

c.) Unix primer

most ~~Ro~~ commands have a "-h", "--h", or "--help" ~~Sw~~ switch which you can pass to the command line.

ie; ls --help ; * the
ssh -h / switch depends on your flavor of unix

if using a switch doesn't help you can try "man" (if its installed) this program prints ~~of~~ out the manual for a given command

try man -k pwd

↳ ~~Find~~ search manual using key word "pwd"

Example 1.) compile the Marsaglia RNG

step. 1.) login to inca (see brems.physics.isu.edu/~tforest/NucSim/Day1/XwindowOnwindows.html)

2.) ~~mkdir "src"~~ mkdir src

3.) cd src

4.) ~~cp -R ~tforest/src/~~ cp -R ~tforest/NucSim/Day1 ./

5.) ls (should see directory called "Day1")

6.) cd Day1

7.) make 8.) ~~RT~~ ./rnd test

I overview

0.) ROOT primer

ROOT is an ~~B~~ Object Oriented ~~B~~ (C^{++}) data analysis frame work distributed freely by CERN at

http://root.cern.ch

it is predominantly used by the high energy physics and nuclear physics communities.

You can get Binary versions for Windows, MacOSX, and several Unix flavors.

~~The~~ It's "open source" so you can download and compile ~~the~~ latest version for your particular OS if you wish. (I'd recommend only attempting this under UNIX)

I would recommend that you try to download it via cvs on inca just for the experience. I took a while to compile but it was painless. (see root.cern.ch/root/CVS.html)

```
cvs -d:pserver:cvs@root.cern.ch:/user/cvs login
        password: cvs
cvs -d:pserver:cvs@root.cern.ch:/user/cvs -z3 co -Proot
```

I overview

D.) Root primer.

Important commands, under the root[#] prompte

1.) .q : quik root
2.) new TBrowser ( ); : launches a GUI
3.) .L filename.C : loads in a script file defining functions.
4.) .X filename.C : executes via the CINT interpretor the code in the file (ie; running program without compiling)
actually CINT compiles it for you on the fly

<u>Example</u>: Create an ntuple ; draw a Histogram.

This inclass example uses ~~Ran~~ the output of ~~ran~~ RN's from Example 1 in section C.).

step 1.) create an ascii file by diverting the stdout of rndtest using the ">" symbol

rndtest 500 > temp.dat

2) load the macro ascii2root.C

root

.L ascii2root.C

ascii2rt("temp.dat")

I overview

a.) Root primer

Example: create ntuple & Draw histogram

3.) Draw histogram using GUI

New TBrowser();

click on 3rd folder in left side window pane which looks like the path to your subdirectory. This should show a list of files. Double click on "rns.root". It will then appear in the same left side pane under the last folder listed called "ROOT files". Single click on the folder labeled "rns.root". A folder will appear in the right side pane labeled "rns;1". Double click "rns;1" in the right side window pane and it will change to look like 2 files called "rnd1" and "rnd2". Double click one of these and a histogram will be created. (assuming you can draw X-windows)

4.) Draw Histogram from command line:

Quit root and restart it passing filename on command line

root rns.root

rns → Draw("rnd1")

To use command line you need to know about the contents of the file. "rns" is a class/pointer in the file pointing to the stored #'s rnd1 & rnd2.