Physics 40: Laboratory Nine

Tuesday, April 28, 2020

Today's Goals: Discussion of diffusion equation code;
Discussion of connection between diffusion and random walks;
Random numbers;
Analytic solution of diffusion equation.

[0] Review the diffusion equation code. Better notation:

```
    double rho[1000],newrho[1000],D,dt,dx,Ddtodx2,norm,realx;
[...]
    printf("\nEnter D,dt,dx   ");
    scanf("%lf %lf %lf",&D,&dt,&dx);
    Ddtodx2=D*dt/(dx*dx);
[...]
            newrho[x]=rho[x]+Ddtodx2*(rho[x+1]+rho[x-1]-2.0*rho[x]);
[...]
        realx=dx*x;
        fprintf(fileout,"\n %8.4lf %8.4lf",realx,rho[x]/dx);
```

[1] Diffusion and random walks.

[2] Random numbers and their moments.

[3] Analytic solution of diffusion equation; Separation of variables; Fourier integrals.

[HW5-1] Modify your diffusion equation code to keep track of the 'normalization', that is the total number of particles in all the boxes by inserting an extra loop inside the 'time' loop:

```
        norm=0.0;
        for (x=1; x<999; x=x+1)
        {
            norm=norm+rho[x];
        }
        printf("\n norm= %8.3lf  ",norm);
```

What do you observe about the behavior of 'norm'? Why is this good or bad? What physical principle is involved?

[HW5-2] Run your diffusion equation code for $D = 0.02, dt = 0.0001, dx = 0.01, N = 100000$. Make a plot of rho[x] versus x. What is the total time t elapsed? How does your plot compare to the analytic solution?

[HW5-3] What does your original diffusion equation code give for rho[x] after one time step if $D = 0.125$? Work it out by hand (ie don't use your program). Suppose instead you wrote the code without using the variable newrho[x]:

```
#include <stdio.h>
#include <math.h>
int main(void)
{
    FILE * fileout;
    int x,t,Nt;
    double rho[1000],D;

    fileout=fopen("slughorn.txt","w");

    printf("\nEnter D*dt/dx^2   ");
    scanf("%lf",&D);
    printf("\nEnter number of time steps   ");
    scanf("%d",&Nt);

    for (x=0; x<1000; x=x+1)
    {
        rho[x]=0.0;
    }
    rho[500]=1.0;

    for (t=0; t<Nt; t=t+1)
    {
        for (x=1; x<999; x=x+1)
        {
            rho[x]=rho[x]+D*(rho[x+1]+rho[x-1]-2.0*rho[x]);
        }
    }

    for (x=0; x<1000; x=x+1)
    {
        fprintf(fileout,"\n %6d %8.4lf",x,rho[x]);
    }

    fclose(fileout);
    printf("\n");
    return 0;
}
```

Work out by hand what this new code would give after one time step. Discuss whether the original version of your code or the new version is more reasonable. In one of the cases you will find the particles spread to the right ($x > 500$) differently from their spread to the left ($x < 500$). Is there a way you could modify your code so that the left/right spread patterns are interchanged?

[3] Type in the following code, which generates N random numbers and prints them to the screen.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

int main()
{
        double R;
        int i,N;
        unsigned int seed;

        printf("\nEnter number of iterations and seed");
        printf("\n");
        scanf("%i %u",&N,&seed);
        srand(seed);

        for(i=0;i<N;i=i+1)
        {
            R=(double)rand()/RAND_MAX;
            printf("%12.8lf \n",R);
        }
        printf("\n");

        return 0;
}
```

[HW5-4] Run your code for $N = 20$ and $seed = 12345$. Run it again with the same N and $seed$. Run it with $N = 20$ and $seed = 54321$. Comment on what you observe? In what sense are the numbers random? In what sense are they not random?

[HW5-5] Modify your code to compute the first six 'moments' of the random numbers. What do you get for $N = 1000000$? What pattern do you observe?

[HW5-6] (extra credit) Prove the result you found in **[HW5-5]** is correct. That is, derive it 'by hand'.