PHYS 325 Assignment 9:
Random Walks

Author: A. J. Ogle
Date: May 10th, 2015
PHYS 325 Computational Physics

1 Problem 7.1: Random Walks in Two Dimensions

The diffusion rate for a one dimensional random walker with equal probability of moving ± 1 in the x direction was investigated. The diffusion constant was found to be 0.9104, with a linear fit ($R^2 = 0.9894$).

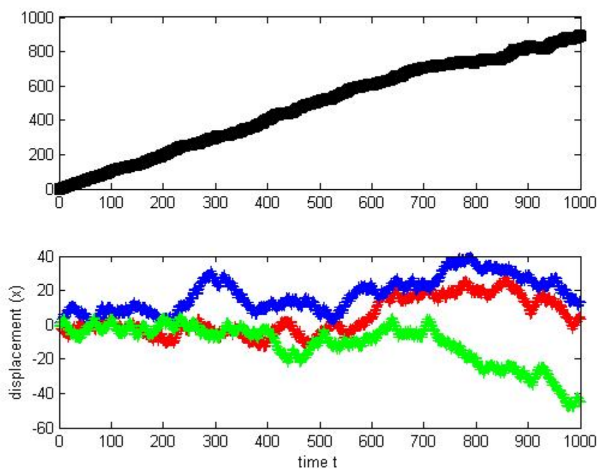

Figure 1: Three random walkers are moving in one dimension, x, with 1000 steps of equal probability towards either +1 or -1. The top part of the figure shows the increase of $< x^2 >$ as the step size increases. The diffusion constant, D, was found to be 0.9104, with a standard error of $SE = 0.002984$.

2 Problem 7.2: Random Walkers in 3 Dimensions

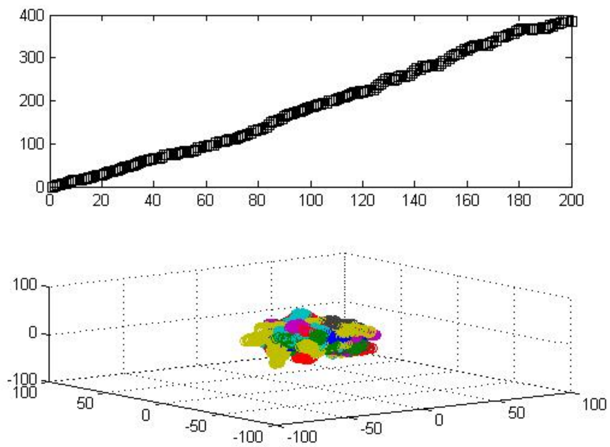

Figure 2: 100 walkers are moving in 3 dimensions (x, y, z) with 100 steps of equal probability for ± 1 in x, y, and z per each step. The top part of the figure shows the increase of $< r^2 >$ as the step size increases. The diffusion constant, D, was found to be 2.060926, with a standard error of $SE = 0.009894$.

The program for a two dimensional random walker was extended to three dimensions and the diffusion constant was found. It was shown to have a proportionality constant for the diffusion rate. That is, the diffusion occurred linearly, with $D = 2.060926$, with a standard deviation and R^2 of $SE = 0.009894$ and $R^2 = 0.9955$. The diffusion is shown in **??**.

3 Conclusions

Random walkers can model physical phenomena such as the diffusion of cream into coffee. In these systes, a statistical approach must be taken to get an idea of the behavior of the system. As opposed to calculating the motion of individual particles, there is more insight to be gained from dealing with these types of problems statistically then mechanically. This method is also more computationally efficient, as it only accounts for probabilities as opposed to accounting for the individual kinematics of every particle - a feat no computer can yet handle.

References

[1] Giordano, Nicholas J., and Hisao Nakanishi. "7. Random Walks" Computational Physics. Upper Saddle River, NJ: Pearson/Prentice Hall, 2006. N. pag. Print.

4 MATLAB code

```
%%Problem 7.1
% find the position and average position squared of 1D random walkers

clear all
% total number of steps (in time) = nsteps
nsteps = 1000;
step = [1:1:nsteps];

% total number of random walkers = nwalkers
nwalkers = 500;
% initialize variables
r = 0.0;
x2=0.0;
for i = 1:nsteps
    x2a(i)=0.0;
    for j=1:nwalkers
    x(i,j)=0.0;
    end;
end;

% perform random walk, find positions at each step,
% calculate the average of position squared for the group
for i = 2:nsteps
    x2=0.0;
    for j=1:nwalkers
    r=rand;
    if (r<0.5)
        x(i,j) = x(i-1,j)-1.0;
    else
        x(i,j) = x(i-1,j)+1.0;
    end;
    x2=x2+(x(i,j)).^2;    %add up all of the x^2 of each walker
    end;
    x2a(i)=x2/nwalkers;
end;

% plot data
subplot(2,1,1); plot(step, x2a, 'ks')
subplot(2,1,2); plot(step,x(:,2),'r+')
hold on
plot(step,x(:,100),'b+')
xlabel('time t');
ylabel('displacement (x)');
plot(step,x(:,266),'g+')


%%Problem 7.2
```

```
% find the position and average position squared of 1D random walkers

clear all
% total number of steps (in time) = nsteps
nsteps = 3000;
dx = 1;
step = [1:dx:nsteps];

% total number of random walkers = nwalkers
nwalkers = 100;
% initialize variables
r = 0.0;

%for the x dimension
x2=0.0;
for i = 1:nsteps
    x2a(i)=0.0;
    for j=1:nwalkers
    x(i,j)=0.0;
    end;
end;

%for the y dimension
y2=0.0;
for i = 1:nsteps
    y2a(i)=0.0;
    for j=1:nwalkers
    y(i,j)=0.0;
    end;
end;

%for the z dimension
z2=0.0;
for i = 1:nsteps
    z2a(i)=0.0;
    for j=1:nwalkers
    z(i,j)=0.0;
    end;
end;

% perform random walk, find positions at each step,
% calculate the average of position squared for the group

%for x
for i = 2:nsteps
    x2=0.0;
    for j=1:nwalkers
    r=rand;
    if (r<0.5)
        x(i,j) = x(i-1,j)-1.0;
```

```
    else
        x(i,j) = x(i-1,j)+1.0;
    end;
    x2=x2+(x(i,j)).^2;    %add up all of the x^2 of each walker
    end;
    x2a(i)=x2/nwalkers;
end;


%for y
for i = 2:nsteps
    y2=0.0;
    for j=1:nwalkers
    r=rand;
    if (r<0.5)
        y(i,j) = y(i-1,j)-1.0;
    else
        y(i,j) = y(i-1,j)+1.0;
    end;
    y2=y2+(y(i,j)).^2;    %add up all of the x^2 of each walker
    end;
    y2a(i)=y2/nwalkers;
end;

%for z
for i = 2:nsteps
    z2=0.0;
    for j=1:nwalkers
    r=rand;
    if (r<0.5)
        z(i,j) = z(i-1,j)-1.0;
    else
        z(i,j) = z(i-1,j)+1.0;
    end;
    z2=z2+(z(i,j)).^2;    %add up all of the x^2 of each walker
    end;
    z2a(i)=z2/nwalkers;
end;

for i = 1:nsteps
    r2a(i) = sqrt(x2a(i)^2 + y2a(i)^2 + z2a(i)^2);
end

subplot(2,1,1); plot(step, r2a, 'ks');
subplot(2,1,2);
for i = 1:nwalkers
    scatter3(x(:,i),y(:,i),z(:,i));
    hold on;
end
axis([-100 100 -100 100 -100 100]);
```

```
% % plot x data
% subplot(6,1,1); plot(step, x2a, 'ks')
% subplot(6,1,2); plot(step,x(:,2),'r+')
% hold on
% plot(step,x(:,100),'b+')
% xlabel('time t');
% ylabel('displacement (x)');
% plot(step,x(:,266),'g+')
%
% %plot the y data
% subplot(6,1,3); plot(step, x2a, 'ks')
% subplot(6,1,4); plot(step,x(:,2),'r+')
% hold on
% plot(step,x(:,100),'b+')
% xlabel('time t');
% ylabel('displacement (x)');
% plot(step,x(:,266),'g+')
%
% %plot the z data
% subplot(6,1,5); plot(step, x2a, 'ks')
% subplot(6,1,6); plot(step,x(:,2),'r+')
% hold on
% plot(step,x(:,100),'b+')
% xlabel('time t');
% ylabel('displacement (x)');
% plot(step,x(:,266),'g+')
```