22.05 P-SET 2

Jacob Miske

September 2018

1 Materials Table

The problem presented analysing the paths taken by one million neutrons in three different media. The random paths taken from an origin 0,0,0 were seeded in MatLab with the function 'rng(1)'. To sample the results of Monte Carlo simulation, instead of 1e6 neutrons, 1000 neutrons were simulated. This allowed for quick, iterative coding.

In the case of "Mean Distance Travelled" and "Sum, Mean Distance Travelled", the first refers to measuring the distance between the absorption point and the origin. The second refers to the additive motions between each scatter and the eventual absorption.

Material	Light Water	Graphite	Heavy Water
Neutron Absorp. Σ	$\Sigma a = 0.030$	$\Sigma a = 0.0025$	$\Sigma a = 0.0025$
Neutron Scatter Σ	$\Sigma s = 0.270$	$\Sigma s = 0.050$	$\Sigma s = 0.025$
Max # of Collisions	101	431	410
Mean # of Collisions	9.72	27.7	26.46
Mean Distance per Collision	4.14cm	20.64cm	35.31cm
1/6 of "m-s crow flight"	29.3cm^2	2,756cm^2	4,156cm^2
Mean Distance Travelled	4.14cm	65.26cm	75.31cm
Max Distance Travelled	72.4cm	474.0cm	754.4cm
Sum, Mean Distance Travelled	53cm	425.26cm	645.12cm
Sum, Max Distance Travelled	257.2cm	1,694.1cm	2,156.3cm

2 The 3D and 2D Projections

The projections provide an interesting look into the clustering of scatter events. As the total cross section increase, the distance between scatters decreases. For light water, the distance between angle changes is a much smaller fraction compared to graphite or heavy water.

Looking at figure 1 and 2, the range of each direction is about 70cm. There are a few rarer cases where a neutron will travel much further than normal in a given direction and continue to scatter locally in the new area that neutron has travelled into.

Looking at figure 3 and 4, graphite has a notable smaller cross section compared to light water. This results in larger bounds for the projections. In the case of the outer tracks, graphite has a higher prevalence for these due to a smaller absorption cross section.

Looking at figure 5 and 6, the clustering about the origin is more pronounced in heavy water than graphite. The heavy water track cloud appears more dense than that of graphite.

3 Characterization of Projections

By examining the resultant plots and table data, one can begin to understand the appearance that neutrons will take on randomly in a material. Viewing the scattering cloud's expansion into different directions is a way of assessing characteristics of each moderator. As the range of the most dense section of the neutron cloud increases, the amount of of moderator cooling flow needed will likely increase to cool a larger volume.

In the case of the AGR or CANDU reactors, the moderating section of the reactor is spatially much larger than that for a PWR or BWR core. The probability of interaction for light water is much higher. The cross sections themselves are functions of the microscopic cross section dependent on atomic principles and the number density of the material. In the case of AGR versus CANDU, the AGR graphite section is much larger due to neutrons that travel much farther than the mean distance that need to be captured in solid material. However, the CANDU reactor utilizes

4 Impacts on Reactor Core Design

The three given materials represent three major moderators for thermal reactors. Understanding how far neutrons can travel in an infinite system can be used to estimate how large the moderator will be. Looking at the distance between collisions can be used to estimate the damage rate on the moderator during reactor operation.

A reactor core designer will need to consider the relative measure of neutrons that will attenuate in different amounts about the neutron's point of origin. As the max number of collisions increases, the mean number of collisions will increase proportionally with respect to the absorption cross section. In the case of graphite versus heavy water, both materials are assumed to have the same absorption cross section but the mean distance per collision changes between the two materials. A small increase in the mean distance per collision will result in a much greater increase in mean distance travelled.

Energy deposition is greatest in the part of the reactor where the density of collisions is greatest. While the angle at a given collision is distributed randomly in the simulation, neutrons will more likely scatter forward. Thus, this simulation is under estimating the overall distances. However, the data

5 Code Example

The following details my code for simulating the neutrons as they travel through light water. The full script details the slight differences in various materials with different cross sections.

```
    %All neutrons are born at 0,0,0. Medium is homogeneous. Only S and A.
%Cross sections independent of energy (monoenergetics)

%Clear variables
clc; clear all
%start time
tic
%Set problem constants
rng(1) %Select a random seed
%neutrons = 1000;
neutrons = 1e6; %number of neutrons to simulate
%Given cross section constants
waterSigmaA=0.030; waterSigmaS=0.270; waterSt=waterSigmaA+waterSigmaS;
```

```
graphiteSigmaA=0.0025; graphiteSigmaS=0.050; graphiteSt=graphiteSigmaA+graphiteSigmaS;
D2OSigmaA=0.0025;D2OSigmaS=0.025; D2OSt=D2OSigmaA+D2OSigmaS;
%Setting boundaries of computation
plottedN=100; scatLimit=200;
%Create placeholders
%% Run for water

for i=1:neutrons
    %Generate Angles of initial direction
    phi=2*pi*rand(); cosNz=(1-2*rand()); sinNz=sqrt(1-cosNz^2);
    cosNx=sinNz*cos(phi); cosNy=sinNz*sin(phi);
    distanceX=0.0; distanceY=0.0; distanceZ=0.0; %Start at zero

    for j=2:1:scatLimit
        %For each neutron up until scatter limit, calculate some random
        %distance until next interaction with medium. Define distance in x,
        %y, and z by random incidence angles.
        distanceTravel=-log(rand())/waterSt;
        distanceX=distanceX+cosNx*distanceTravel;
        distanceY=distanceY+cosNy*distanceTravel;
        distanceZ=distanceZ+cosNz*distanceTravel;
        if i<=neutrons
            x(j,i)=distanceX;y(j,i)=distanceY;z(j,i)=distanceZ;
        end
        if rand() < real(waterSigmaS/waterSt)
            phi=2*pi*rand(); cosNz=(1-2*rand()); sinNz=sqrt(1-cosNz^2);
            cosNx=sinNz*cos(phi); cosNy=sinNz*sin(phi);
        else
            break;
        end
    end
end

% % Plot location of water points in 3D
figure(1) %First fig
for i = 1:(plottedN-2) %run through all events (s and eventually a)
    plot3(x(:,i),y(:,i),z(:,i))
    hold on; grid on
end
%Create x at end of neutron travel
for i =(plottedN-1)
    plot3(x(:,i), y(:,i), z(:,i), '-bX')
end
title('100 Neutron Paths in Light Water 3D')
xlabel('X (cm)');ylabel('Y (cm)');zlabel('Z (cm)'); %cross section units are (cm^-1)
saveas(gcf,'Light Water Neutron Path 3D Projection.pdf')

%% Plot location of water points in 2D
figure(4);
for n=1:plottedN;
    plot(x(:,n),y(:,n)); hold on; grid on
```

```
end
xlabel('X')
ylabel('Y')
title('100 Light Water Neutron Paths on 2D Projection')
saveas(gcf,'Light Water Neutron Path on 2D Projection.pdf')


%% %% Table Calculations for water
%Max number of collisions
waterMaxCol = size(x,1);
waterX = x; waterY = y; waterZ = z;

%Mean and Max # of collisions to absorption
%Assume neutron doesn't ever return to 0,0,0
collisionCounter = 0;
collisionsPerN = [];
%Relate each point in trace to previous point
distanceBetweenPoints = 0;
distanceToOrigin = 0; originPoint=[0,0,0]
distanceBetweenN = [];
distanceFromOriginTracker = [];
% Reminds which point is under watch
currentPoint=[]; previousPoint=[]; twoPoints=[];

%Run through waterX, ...Y, ...Z
for j = 1:size(waterX,2) %1:100 number of saved neutrons
    collisionCounter=0; %reset iterator

    for i = 2:size(waterX,1) %2:37 (37 happens to be max(col.) -1 (origin))
        if waterX(i,j) ~= 0
            collisionCounter=collisionCounter+1;
        end
        %Define point at each spot in arrays
        currentPoint=[waterX(i,j),waterY(i,j),waterZ(i,j)]; previousPoint=[waterX(i-1,j),waterY(i-1,j),w
        twoPoints=[currentPoint;previousPoint];
        %Use pdist function to figure distance along tracks
        distanceBetweenPoints = pdist(twoPoints);
        distanceToOrigin = pdist([currentPoint;originPoint]);
        %Fill arrays
        distanceBetweenN(i-1,j) = distanceBetweenPoints;
        distanceFromOriginTracker(i-1,j) =distanceToOrigin;

    end
    %Save collisionCounter
    collisionsPerN(j) = (collisionCounter);
end
%Using written lists
waterMeanCollisions = mean(collisionsPerN);
waterMaxCollisions = max(collisionsPerN);

%% Calculating mean distance for the individual points, and for each
```

```
%neutron's path
%Runs a find function for non-zeros, only accounts them for each column
[~,ii,v] = find(distanceBetweenN);
waterMeanPntPntDistancePerN = accumarray(ii,v,[],@mean);
waterMeanPntPntDistance = mean(waterMeanPntPntDistancePerN);

%%  Derive 1/6 "mean square crow flight distance"
%Take list of distances from origin , square them, and find the mean (*1/6)

%Run through ...X, ...Y, ...Z
for j = 1:size(waterX,2) %1:100 number of saved neutrons
    collisionCounter=0; %reset iterator
    for i = 2:size(waterX,1) %2:37 (happens to be max(col.) -1 (origin))
        if waterX(i,j) ~= 0
            collisionCounter=collisionCounter+1;
        end
    end
    %Save collisionCounter
    collisionsPerN(j) = (collisionCounter);
    waterMeanSquareCrowFlight(j) =  distanceFromOriginTracker(collisionCounter,j);

end
%Run square root and then mean all values
waterMeanSquareCrowFlight = sqrt(waterMeanSquareCrowFlight);
waterMeanSquareCrowFlight = mean(waterMeanSquareCrowFlight);
waterMeanSquareCrowFlight=waterMeanSquareCrowFlight*1/6;
%% Mean distance travelled between birth and absorption
%Runs a find function for non-zeros, only accounts them for each column
[~,ii,v] = find(distanceFromOriginTracker);
waterMeanPntOriginDistancePerN = accumarray(ii,v,[],@mean);
waterMeanPntOriginDistance = mean(waterMeanPntOriginDistancePerN);
%% Maximum distance of any neutron during flight (cm)
waterMaxFromOriginDistancePerN = max(distanceFromOriginTracker);
waterMaxFromOriginDistance = max(waterMaxFromOriginDistancePerN);
```

Bibliography

22.05 Lecture 3

Lamarsh chapter 2

Water Neutron Path 3D Projection.pdf

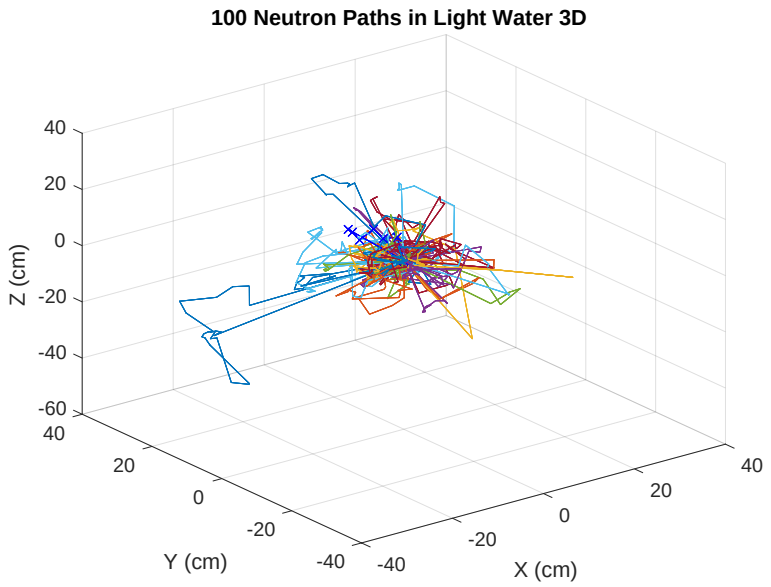

Figure 1: Light Water 3D

Water Neutron Path on 2D Projection.pdf

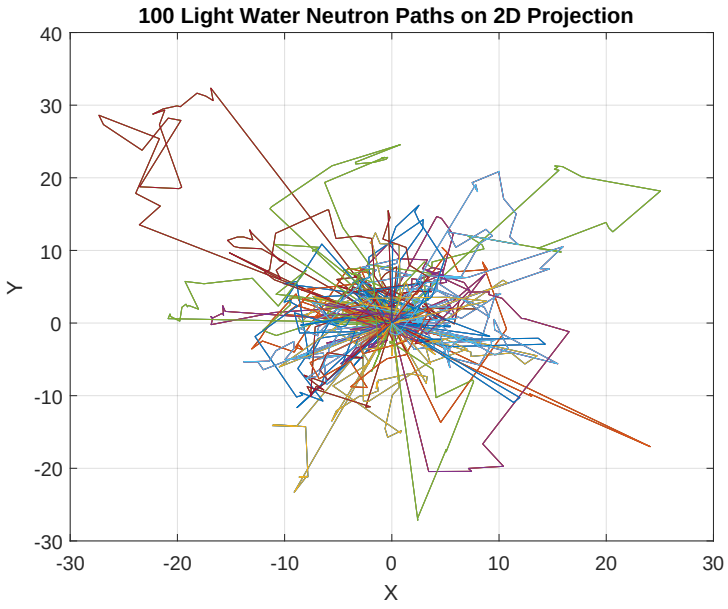

Figure 2: Light Water 2D

Neutron Path 3D Projection.pdf

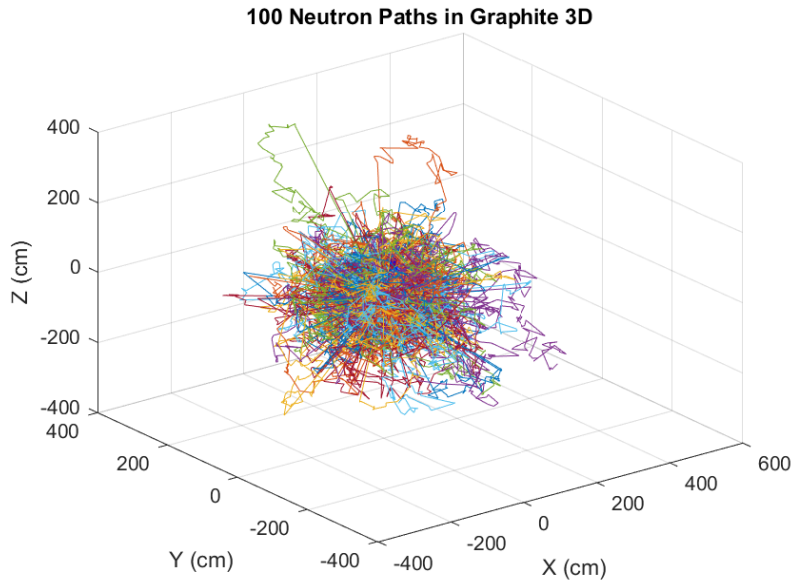

Figure 3: Graphite 3D

Neutron Path on 2D Projection.pdf

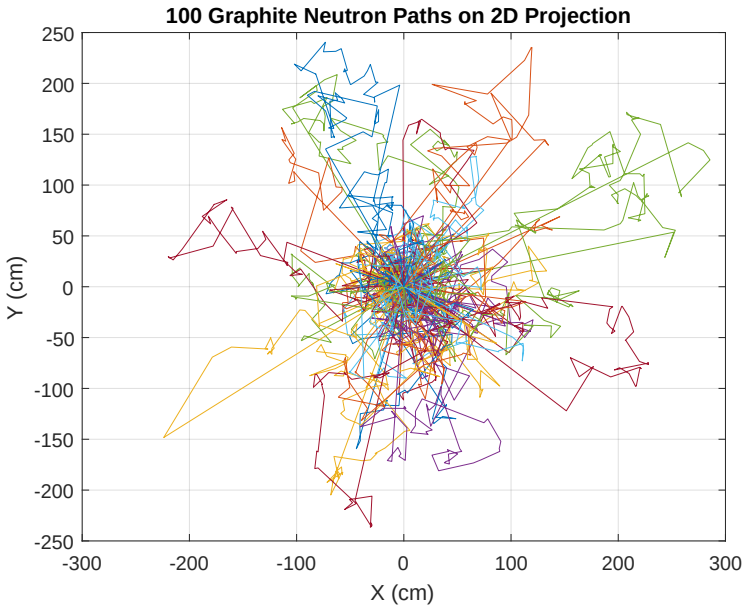

Figure 4: Graphite 2D

Water Neutron Path 3D Projection.pdf

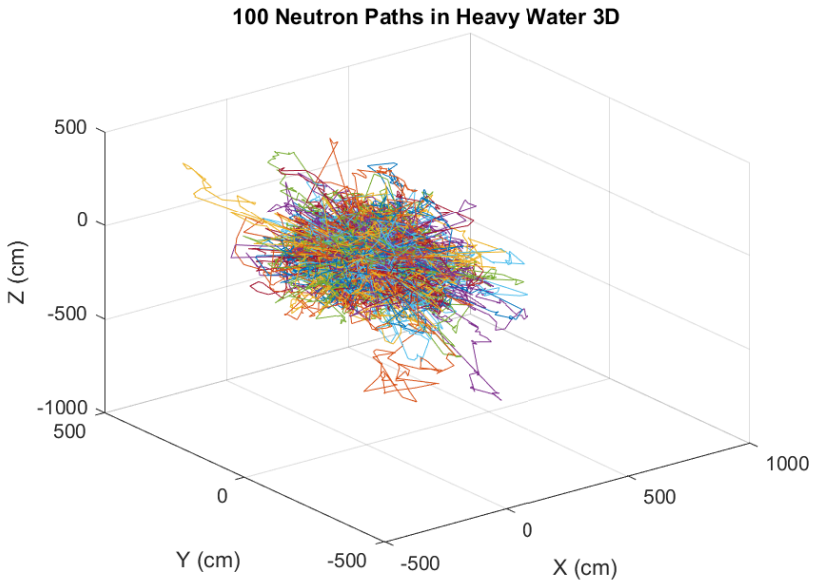

Figure 5: Heavy Water 3D

Water Neutron Path 2D Projection.pdf

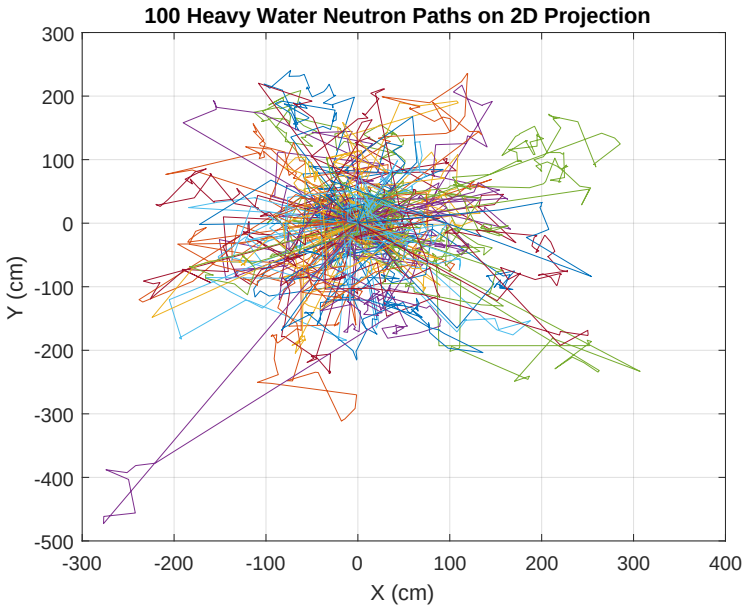

Figure 6: Heavy Water 2D