Table of Contents

Reactor theory Monte Carlo modeling project

```
% Alexander W Safranek
% NE-6708 Reactor Theory, Dr. Richard Vasques.
% Last Edited: 12/01/2021

clear; clc; close all;
format short g;
```

Material parameters:

```
numParts = 1e6;                         % number of iterations
% Note: 100 is sufficient if NumBins is small
% Note: as n -> inf, shortest propogation distance -> dominant

BN.S_alpha = 38.4;           % absorption into alpha production [1/cm]
% BN.S_proton = 0.0944;        % absorption into proton production [1/cm]
% BN.S_gamma = 0;              % absorption into gamma production [1/cm]
BN.S_el = 0.73;                 % elastic scattering [1/cm]
BN.S_tot = BN.S_alpha + BN.S_el;     % total interaction [1/cm]

% SiC.S_alpha = 0;             % absorption into alpha production [1/cm]
% SiC.S_proton = 0;            % absorption into proton production [1/cm]
SiC.S_gamma = 0.1817;             % absorption into gamma production [1/cm]
SiC.S_el = 0.32;                % elastic scattering [1/cm]
SiC.S_tot = SiC.S_gamma + SiC.S_el;     % total interaction [1/cm]
```

Set layer thicknesses

```
BN.width = 0.0001;      % 1-um layer thickness [cm]
SiC.width = 0.01;      % 100-um layer thickness [cm]
```

Histogram parameters:

```
tsz = 16;
posit = [75 75 1200 600];
% posit = [75 75 600 600];
n_bins = 50000;    % number of bins in histogram (many are necessary)
```

Simulation (Boron Nitride):

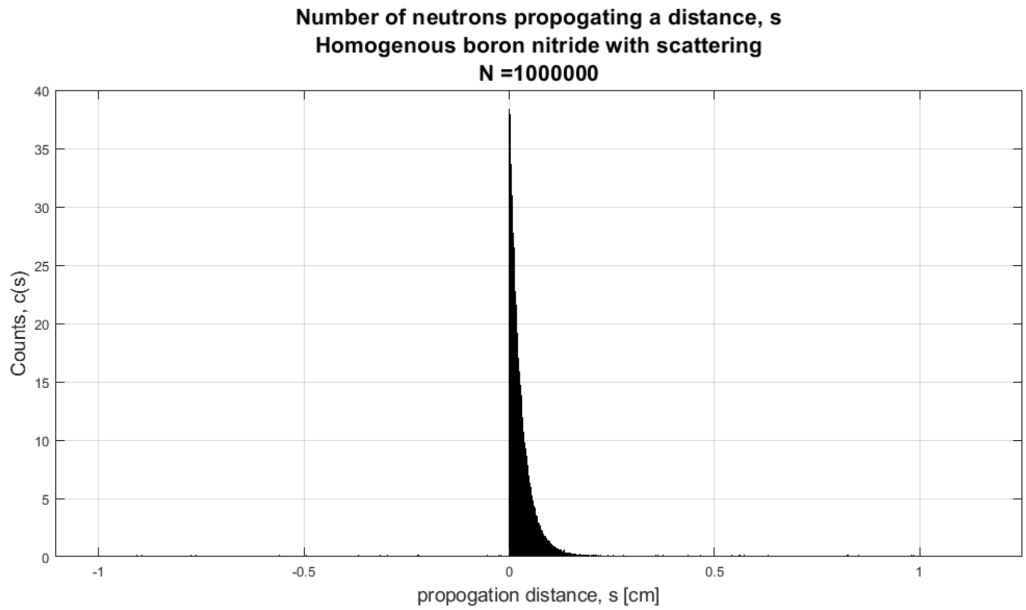
```
eta_1 = rand(numParts,1);                  % n random numbers btw [0,1] == F(s)
% X = -(1/BN.S_alpha)*log(1-eta_1);    % distance to any interaction
                                    % without scattering considered
for ii=1:numParts
    % neutron absorption, alpha production in BN:
    if eta_1(ii) < BN.S_alpha/BN.S_tot               % 98.1%
        x_1a(ii,1) = -(1/BN.S_alpha)*log(1-rand(1,1)); %#ok<SAGROW>
                      % propogation distance to absorption in BN [cm]
    % scattered interactions in BN:
    elseif eta_1(ii) >= BN.S_alpha/BN.S_tot
        mu_1(ii,1) = 1-2*rand(1,1);                   %#ok<SAGROW>
        x_1s(ii,1) = -(1/BN.S_alpha)*log(1-rand(1,1)) + mu_1(ii); %#ok<SAGROW>
%          phi_1(ii,1) = 2*pi*rand(1,1);
    else
        error('warning on condition 1')
    end
end

% sort particles
anonzero = find(x_1a~=0);
x_1a = x_1a(anonzero);  % absorbed (lost)
n_1a = length(x_1a);

snonzero = find(x_1s~=0);
x_1s = x_1s(snonzero);
n_1s = length(x_1s);       % only one scattered particle remains
                               %   in BN region (neglected)
% table(n_1a,n_1s)
```

Plot for boron nitride region alone:

it would be far more efficient for thicker materials because few particles are absorbed within 1 um. Even fewer are scattered.

```
x_1 = [x_1a; x_1s]; % to see distribution in homogenous, thick BN
```

```
for ff = 1
    figure(ff)
    set(gcf,'position',posit)

    h = histogram(x_1,n_bins);
%     bin_limits = h.BinLimits;           % bounds of s
%     h.BinLimits = [0,BN.width + SiC.width];
%     h.BinLimits = [0,BN.width + SiC.width/2];
%     bin_width = h.BinWidth;             % width of each bin [cm]
%     bin_counts = h.Values;            % number of counts in each s-range

    set(h,'normalization','pdf')
    set(h,'FaceColor',0.5*[1,0,1])          % RGB-pink
    grid on
    hold on
%     plot(bin_limits)
    hold off
    title({'Number of neutrons propogating a distance, s'; ...
        'Homogenous boron nitride with scattering'; ...
        strcat('N = ',num2str(numParts))},'fontsize',tsz)
    xlabel('propogation distance, s [cm]','fontsize',tsz-2)
    ylabel('Counts, c(s)','fontsize',tsz-2)
end
```



Determine interactions in BN:

```
backBN = find(x_1<0);
n_1_back = length(backBN);
x_1_back = x_1(backBN);

% absorption -> alpha
absinBN = find(x_1a<=BN.width & x_1a>=0);
```

```
n_absinBN = length(absinBN);
x_1a_inBN = x_1a(absinBN);           % particle displacements in BN layer
                              % small percentage of particles created
scatinBN = find(x_1s<=BN.width & x_1s>=0);
n_scatinBN = length(scatinBN);
x_1s_inBN = x_1s(scatinBN);           % particle displacements in BN layer
                              % small percentage of particles created

x_1_inBN = [x_1a_inBN; x_1s_inBN];
```

Because scattered particles tend to scatter out of the first region.

```
table(n_absinBN,n_scatinBN)
% need to check number of scatters to improve robustivity

if n_scatinBN<1000
else
    error('error accumulating due to scattered particles in BN')
end

% forward particles (these are dominant):
fwdBN = find(x_1>BN.width);
n_fwdBN = length(fwdBN);     % number of particles to move to next region
x_1_fwd = x_1(fwdBN);             % these lengths become BN.width

% Note: particles moving in net negative direction are lost.


ans =

  1×2 table

    n_absinBN    n_scatinBN
    _________    __________

      3802           0
```

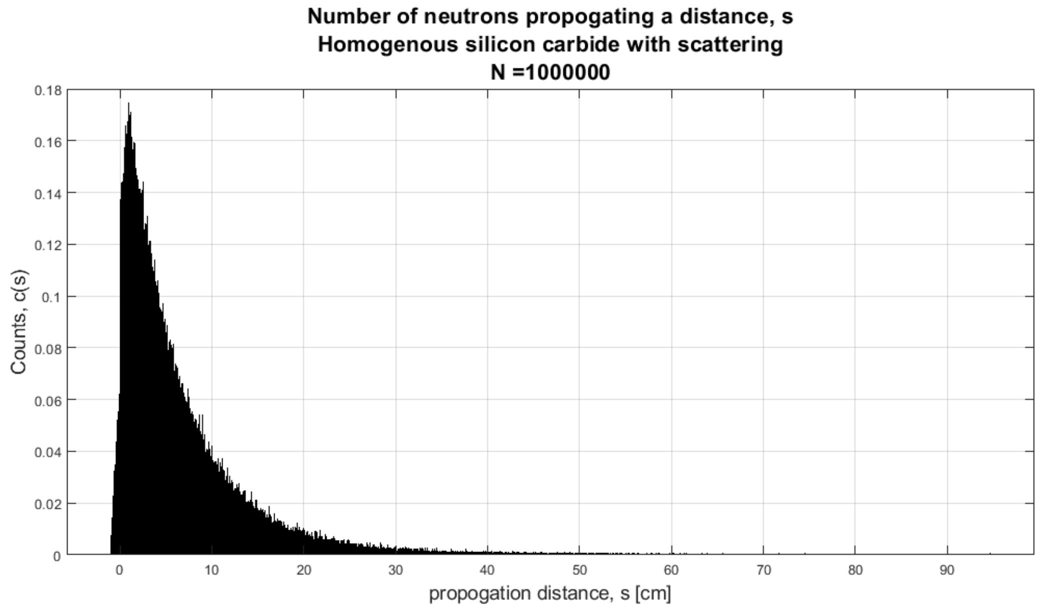
Simulation (Silicon Carbide):

```
eta_2 = rand(n_fwdBN,1);         % particles that do not interact with BN
% x_2 = zeros(n_fwdBN,1);

% new loop for different pdf:
for ii = 1:n_fwdBN
    % neutron absorption in SiC, gamma production:
    if eta_2(ii) < SiC.S_gamma/SiC.S_tot                % 36.2%
        x_2a(ii,1) = -(1/SiC.S_gamma)*log(1-rand(1,1)); %#ok<SAGROW>
                                % propogation distance [cm]
    % scattering interactions:
```

```
    elseif eta_2(ii) >= SiC.S_gamma/SiC.S_tot
        mu_2(ii,1) = 1-2*rand(1,1);                    %#ok<SAGROW> % size
 unknown
        x_2s(ii,1) = -(1/SiC.S_gamma)*log(1-rand(1,1)) +
 mu_2(ii); %#ok<SAGROW>
%         phi_2(ii,1) = 2*pi*rand(1,1);
    else
        error('warning on condition 2')
    end
end


% sort particles
anonzero2 = find(x_2a~=0);
x_2a = x_2a(anonzero2);     % absorbed (lost)
n_2a = length(x_2a);

snonzero2 = find(x_2s~=0);
x_2s = x_2s(snonzero2);
n_2s = length(x_2s);        % only one scattered particle remains
                                %   in BN region (neglected)
% table(n_2a,n_2s)
```

Plot silicon carbide region alone:

```
x_2 = [x_2a; x_2s];

for ff = 2
    figure(ff)
    set(gcf,'position',posit)

    h = histogram(x_2,n_bins);
%     bin_limits = h.BinLimits;           % bounds of s
%     h.BinLimits = [0,BN.width + SiC.width];
%     h.BinLimits = [0,BN.width + SiC.width/2];
%     bin_width = h.BinWidth;             % width of each bin [cm]
%     bin_counts = h.Values;            % number of counts in each s-range


    set(h,'normalization','pdf')
    set(h,'FaceColor',0.5*[1,0,1])          % RGB-pink
    grid on
    hold on
%     plot(bin_limits)
    hold off
    title({'Number of neutrons propogating a distance, s'; ...
        'Homogenous silicon carbide with scattering'; ...
        strcat('N = ',num2str(numParts))},'fontsize',tsz)
    xlabel('propogation distance, s [cm]','fontsize',tsz-2)
    ylabel('Counts, c(s)','fontsize',tsz-2)
end
```



Determine interactions in SiC:

backscattering into BN (zero particles)

```
backSiC = find(x_2<0);
n_2_back = length(backSiC);   % ~30000 backscattered into BN
x_2_back = x_2(backSiC);
% run simulation again for these particles and it will have increased
    % absorptions in BN

% absorption -> gamma in SiC
absinSiC = find(x_2a<=SiC.width & x_2a>=0);
n_absinSiC = length(absinSiC);    % ~683 absorbed in SiC
x_2a_inSiC = x_2a(absinSiC);         % particle displacements in SiC layer

% scattered, but remain in SiC
scatinSiC = find(x_2s<=SiC.width & x_2s>=0);  % index of scattered particles
 remaining in SiC
n_scatinSiC = length(scatinSiC);    % ~522 scattered
x_2s_inSiC = x_2s(scatinSiC);        % particle displacements in SiC layer

table(n_absinSiC,n_scatinSiC)

x_2_inSiC = [x_2a_inSiC; x_2s_inSiC];

% particles scattered forward, out of the region are lost.
fwdSiC = find(x_2>SiC.width);
n_fwdSiC = length(fwdSiC);                % move to next region
x_2_fwd = x_2(fwdSiC);              % ~ 959272 (most are lost)

x_2infregion = [x_1_inBN;
    (BN.width + x_2_inSiC);
```

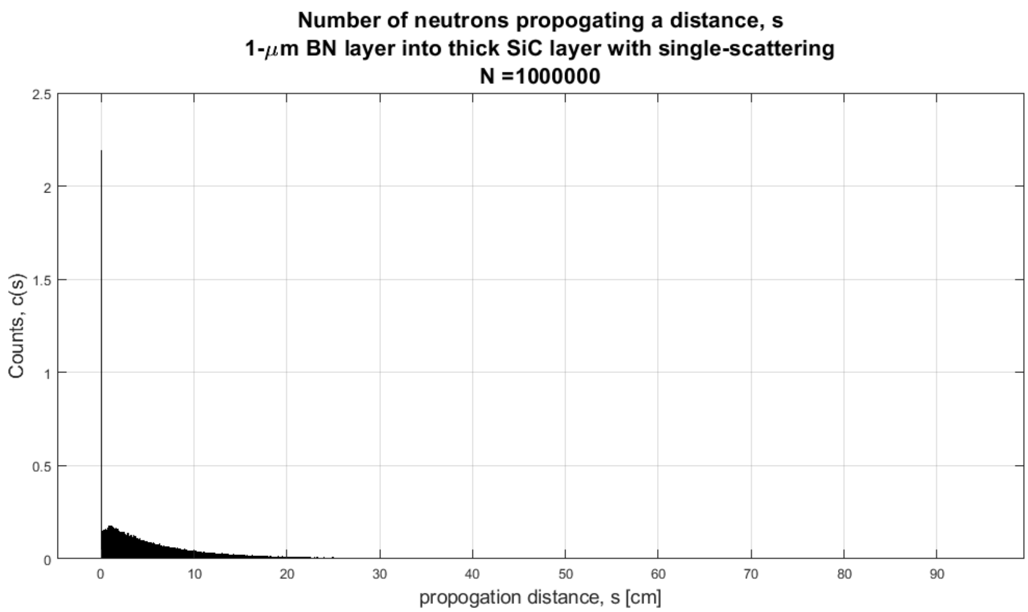
```
    (BN.width + x_2_fwd)]; % histogram data for two-region problem

x_2region = [x_1_inBN;
    (BN.width + x_2_inSiC)]; % histogram data for two-region problem
```

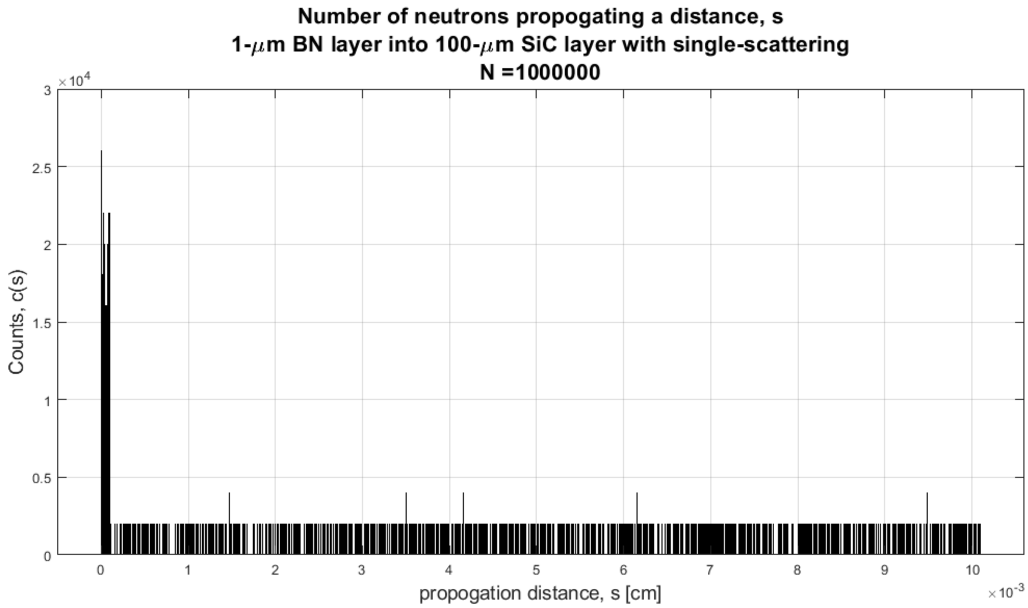
```
ans =

  1×2 table

    n_absinSiC    n_scatinSiC
    __________    ___________

       603            542
```

Histogram for 2-region problem:

```
for ff = 3
    figure(ff)
    set(gcf,'position',posit)

    h = histogram(x_2infregion,n_bins);
    bin_limits = h.BinLimits;            % bounds of s
%     h.BinLimits = [0,BN.width + SiC.width];
%     h.BinLimits = [0,BN.width + SiC.width/2];
    bin_width = h.BinWidth;              % width of each bin [cm]
    bin_counts = h.Values;             % number of counts in each s-range


    set(h,'normalization','pdf')
    set(h,'FaceColor',0.5*[1,0,1])          % RGB-pink
    grid on
    hold on
%     plot(bin_limits)
    hold off
    title({'Number of neutrons propogating a distance, s'; ...
        '1-\mum BN layer into thick SiC layer with single-scattering'; ...
        strcat('N = ',num2str(numParts))},'fontsize',tsz)
    xlabel('propogation distance, s [cm]','fontsize',tsz-2)
    ylabel('Counts, c(s)','fontsize',tsz-2)
end


for ff = 4
    figure(ff)
    set(gcf,'position',posit)

    h = histogram(x_2region,n_bins*2);
    bin_limits = h.BinLimits;            % bounds of s
%     h.BinLimits = [0,BN.width + SiC.width];
%     h.BinLimits = [0,BN.width + SiC.width/2];
    bin_width = h.BinWidth;              % width of each bin [cm]
```

```
    bin_counts = h.Values;            % number of counts in each s-range


    set(h,'normalization','pdf')
    set(h,'FaceColor',0.5*[1,0,1])          % RGB-pink
    grid on
    hold on
%     plot(bin_limits)
    hold off
    title({'Number of neutrons propogating a distance, s'; ...
        '1-\mum BN layer into 100-\mum SiC layer with single-scattering'; ...
        strcat('N = ',num2str(numParts))},'fontsize',tsz)
    xlabel('propogation distance, s [cm]','fontsize',tsz-2)
    ylabel('Counts, c(s)','fontsize',tsz-2)
end
```





Additional interactions from scattered particles in SiC:

```
eta_3 = rand(n_scatinSiC,1);     % particles that interact twice

% new loop for pdf:
for ii = 1:n_scatinSiC
    % neutron absorption in SiC, gamma production:
    if eta_3(ii) < SiC.S_gamma/SiC.S_tot                % 36.2%
        x_3a(ii,1) = -(1/SiC.S_gamma)*log(1-rand(1,1)); %#ok<SAGROW>
                                % propogation distance [cm]
    % scattering interactions:
    elseif eta_3(ii) >= SiC.S_gamma/SiC.S_tot
        mu_3(ii,1) = 1-2*rand(1,1);                    %#ok<SAGROW> % size
 unknown
        x_3s(ii,1) = -(1/SiC.S_gamma)*log(1-rand(1,1)) +
 mu_3(ii); %#ok<SAGROW>
%         phi_3(ii,1) = 2*pi*rand(1,1);
    else
        error('warning on condition 2')
    end
end

% sort particles
anonzero3 = find(x_3a~=0);
x_3a = x_3a(anonzero3);     % absorbed (lost)
n_3a = length(x_3a);

snonzero3 = find(x_3s~=0);
x_3s = x_3s(snonzero3);
n_3s = length(x_3s);        % only one scattered particle remains
```

```
                                % in BN region (neglected)
% table(n_3a,n_3s)

x_3 = [x_3a; x_3s];
dx_3 = x_3 + x_2s_inSiC;
```

Histogram of second interaction lengths:

for ff = 5 figure(ff) set(gcf,'position',posit)

```
   h = histogram(x_3,n_bins);
%     bin_limits = h.BinLimits;           % bounds of s
%     h.BinLimits = [0,BN.width + SiC.width];
%     h.BinLimits = [0,BN.width + SiC.width/2];
%     bin_width = h.BinWidth;             % width of each bin [cm]
%     bin_counts = h.Values;            % number of counts in each s-range

   set(h,'normalization','pdf')
   set(h,'FaceColor',0.5*[1,0,1])          % RGB-pink
   grid on
   hold on
%     plot(bin_limits)
   hold off
   title({'Number of neutrons propogating a distance, s'; ...
       'Homogenous silicon carbide, 2nd interaction lengths'; ...
       strcat('N = ',num2str(numParts))},'fontsize',tsz)
   xlabel('propogation distance, s [cm]','fontsize',tsz-2)
   ylabel('Counts, c(s)','fontsize',tsz-2)
end
```

Sort second set of interactions in SiC (turn into a function):

absorption -> gamma in SiC

```
absinSiC2 = find(dx_3<=SiC.width & dx_3>=0);
n_absinSiC2 = length(absinSiC2);    % ~683 absorbed in SiC
x_3a_inSiC = dx_3(absinSiC2);         % particle displacements in SiC layer

% scattered, but remain in SiC
scatinSiC2 = find(dx_3<=SiC.width & dx_3>=0);
n_scatinSiC2 = length(scatinSiC2);    % ~522 scattered
x_3s_inSiC = dx_3(scatinSiC2);        % particle displacements in SiC layer

table(n_absinSiC2,n_scatinSiC2)
% can see that without more particles, only a few undergo a second
% reaction.

x_3_inSiC = [x_3a_inSiC; x_3s_inSiC];

% particles scattered forward, out of the region are lost.
fwdSiC2 = find(dx_3>SiC.width);
```

```
n_fwdSiC2 = length(fwdSiC2);                    % move to next region
x_3_fwd = dx_3(fwdSiC2);                        % ~ 959272 (most are lost)

x_3infregion = [x_1_inBN;
    (BN.width + x_2_inSiC);
    (BN.width + x_3_inSiC);
    (BN.width + x_3_fwd)]; % histogram data for two-region problem

x_3region = [x_1_inBN;
    (BN.width + x_2_inSiC);
    (BN.width + x_3_inSiC)]; % histogram data for two-region problem
```

```
ans =

  1×2 table

    n_absinSiC2    n_scatinSiC2
    ___________    ____________

         0              0
```

Histogram for 2-region problem:

for ff = 6 figure(ff) set(gcf,'position',posit)

```
   h = histogram(x_3infregion,n_bins);
   bin_limits = h.BinLimits;           % bounds of s
%     h.BinLimits = [0,BN.width + SiC.width];
%     h.BinLimits = [0,BN.width + SiC.width/2];
   bin_width = h.BinWidth;             % width of each bin [cm]
   bin_counts = h.Values;            % number of counts in each s-range

   set(h,'normalization','pdf')
   set(h,'FaceColor',0.5*[1,0,1])          % RGB-pink
   grid on
   hold on
%     plot(bin_limits)
   hold off
   title({'Number of neutrons propogating a distance, s'; ...
       '1-\mum BN layer into thick SiC layer with double-scattering'; ...
       strcat('N = ',num2str(numParts))},'fontsize',tsz)
   xlabel('propogation distance, s [cm]','fontsize',tsz-2)
   ylabel('Counts, c(s)','fontsize',tsz-2)
end

% for ff = 7
%     figure(ff)
%     set(gcf,'position',posit)
%
%     h = histogram(x_3region,n_bins*2);
%     bin_limits = h.BinLimits;           % bounds of s
% %     h.BinLimits = [0,BN.width + SiC.width];
```

```
% %     h.BinLimits = [0,BN.width + SiC.width/2];
%     bin_width = h.BinWidth;             % width of each bin [cm]
%     bin_counts = h.Values;            % number of counts in each s-range
%
%
%     set(h,'normalization','pdf')
%     set(h,'FaceColor',0.5*[1,0,1])          % RGB-pink
%     grid on
%     hold on
% %     plot(bin_limits)
%     hold off
%     title({'Number of neutrons propogating a distance, s'; ...
%         '1-\mum BN layer into 100-\mum SiC layer with double-
scattering'; ...
%         strcat('N = ',num2str(numParts))},'fontsize',tsz)
%     xlabel('propogation distance, s [cm]','fontsize',tsz-2)
%     ylabel('Counts, c(s)','fontsize',tsz-2)
% end
```

Post-simulation calculations:

```
total_counts = sum(bin_counts);     % total number of samples == n
ds = (bin_limits(2)-bin_limits(1))...
    /n_bins;                            % propogation range of plot

% plot(1:length(bin_counts),bin_counts)
```

Analysis: Efficiency estimation

```
particles_counted = n_absinBN + n_absinSiC + n_absinSiC2;
efficiency_metric = particles_counted/numParts;
table(efficiency_metric)    % ~0.5% of particles are counted
                            % Need ~1000 particles to count 5


ans =

  table

    efficiency_metric
    _________________

        0.004405
```

Published with MATLAB® R2021b