NE506: Homework 6

Authored by Matthew Klebenow

Contents

- Part 1: Tally Routine
- Derivation of Scores in III
- Estimator of the particle current on the surface
- Estimator of the particle flux on the surface
- Estimator of the particle flux in the sphere
- Estimator of the particle energy deposition in the sphere
- Pulse height tally in the sphere
- Script

Part 1: Tally Routine

The provided CSV file has a set of many instances of $x_0, y_0, z_0, E_0, x_c, y_c, z_c, E', x_f, y_f, z_f$. Write a `MATLAB` script to read these (see `csvread`), calculate the scores as derived in III, and provide:

- A histogram of the results for each tally type
- The mean value for each tally type
- The relative error for each tally type

Derivation of Scores in III

Estimator of the particle current on the surface

First we need the velocity vector of the particle before and after the collision, $\bar{r}_0$ and $\bar{r}_f$. (Note that we define $\bar{x}_i = (x_i, y_i, z_i)$)

$$\bar{r}_0 = (u_0, v_0, w_0) = (\bar{x}_c - \bar{x}_0)$$

$$\bar{r}_f = (u_f, v_f, w_f) = (\bar{x}_f - \bar{x}_c)$$

Now we solve the surface equation for s:

$$(x_p + su)^2 + (y_p + sv)^2 + (z_p + sw)^2 = 100$$

This should be solved for both the pre- and post-collision velocities (although the post-collision equation should substitute $\bar{x}_p$ with $\bar{x}_c$). Note that only values of s such that $0 < s < 1$ should be accepted.

The final score for the particle current estimator is the number of valid solutions for s that are found.

Estimator of the particle flux on the surface

Particle surface flux will need a score calculated at each valid solution for s found in the previous calculation. First, we determine the point at which the particle crosses the surface, $\bar{p}$.

$$\bar{p}_i = \bar{x}_i + s_i \bar{r}_i$$

Since we know $\bar{p} \cdot \bar{r} = |\bar{p}| \cdot |\bar{r}| \cdot \cos\theta$, we can solve for $\cos\theta$ easily:

$$\cos\theta = \frac{\bar{p} \cdot \bar{r}}{|\bar{p}| \cdot |\bar{r}|}$$

Finally, we calculate the score as the sum of the absolute value of the inverse of each value of $\cos\theta$, normalized by the product of the number of histories and the area of the surface:

$$\bar{\phi}_A = \frac{1}{NA} \sum_i \frac{1}{|\cos\theta_i|}$$

Estimator of the particle flux in the sphere

Particle volume flux will require calculation of the track length within the sphere. Once the track length is calculated, the tally can be computed.

$$\bar{\phi}_V = \frac{1}{NV} \sum_i T_{l,i}$$

The track length (T_l) will be calculated by first determining if the beginning or ending position of the particle is inside of the sphere and then applying the appropriate distance s.

Estimator of the particle energy deposition in the sphere

The energy desposition estimator uses the value of track length (T_l) from the particle flux estimation. The estimator that follows is:

$$\bar{Q}_V = \frac{\rho_a}{N\rho_g V} \sum_i H_T\left(E_i\right) T_{l,i}$$

Where $H_T\left(E\right) = E^2$, $\rho_a = 0.02$, $\rho_g = 1$, and E is the energy lost by the particle.

Pulse height tally in the sphere

The pulse height tally is essentially a measure of how much energy is deposited in the cell as a result of all particle histories.

Script

```
% Clear
clear;
% Problem-specific data
r = 10; % sphere radius 10 cm
rho_a = 0.02; % atom density [atoms/b.cm]
rho_g = 1; % mass density [g/cm^3]
A = 4*pi*r^2; % area [cm^2]
V = (4/3)*pi*r^3; % volume [cm^3]
% Ht(E) = E^2 [MeV.b]

% Read data from csv
filename = 'trackData.csv';
data = csvread(filename);
[N,~] = size(data);
x_0 = data(:,1); % initial position
y_0 = data(:,2);
z_0 = data(:,3);
E_0 = data(:,4); % initial energy
x_c = data(:,5); % collision position
y_c = data(:,6);
```

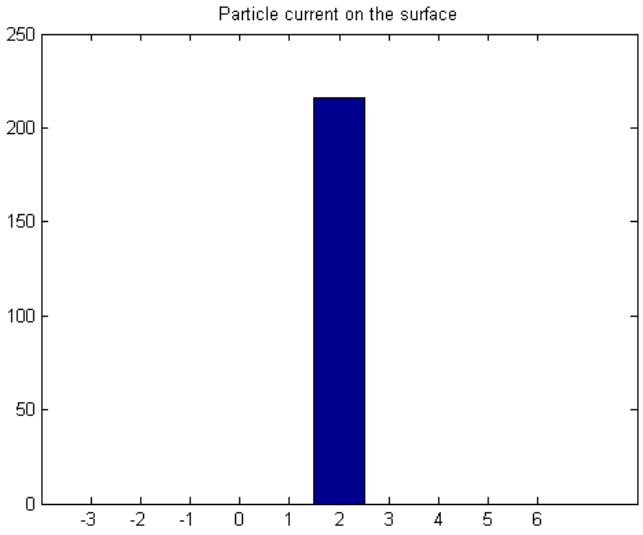
```
z_c = data(:,7);
E_l = data(:,8); % energy lost
x_f = data(:,9); % final position
y_f = data(:,10);
z_f = data(:,11);

% General data crunching
E_f = E_0 - E_l; % final particle energy

% Initialize estimators
curr   = zeros(N,1);
flux_s = zeros(N,1);
flux_v = zeros(N,1);
energy = zeros(N,1);
pulse  = zeros(N,1);

% Score each particle
for i=1:N
    % Declare syms
    syms s;
    % Define start, collision, and finish vectors
    start  = [x_0(i);y_0(i);z_0(i)];
    coll   = [x_c(i);y_c(i);z_c(i)];
    finish = [x_f(i);y_f(i);z_f(i)];
    % Define initial and final direction vectors
    r_0 = coll - start;
    r_f = finish - coll;
    % Solve initial intercept distance
    dist_0 = solve((start(1)+s*r_0(1))^2 + (start(2)+s*r_0(2))^2 + ...
        (start(3)+s*r_0(3))^2 == 100, s);
    dist_0 = double(dist_0);
    s_0 = [];
    for j=1:length(dist_0)
        if dist_0(j) > 0 && dist_0(j) < 1
            % Score if positive and less than 1
            s_0 = [s_0; dist_0(j)];
        end
    end
    % Solve final intercept distance
    dist_f = solve((coll(1)+s*r_f(1))^2 + (coll(2)+s*r_f(2))^2 + ...
        (coll(3)+s*r_f(3))^2 == 100, s);
    dist_f = double(dist_f);
    s_f = [];
    for j=1:length(dist_f)
        if dist_f(j) > 0 && dist_f(j) < 1
```

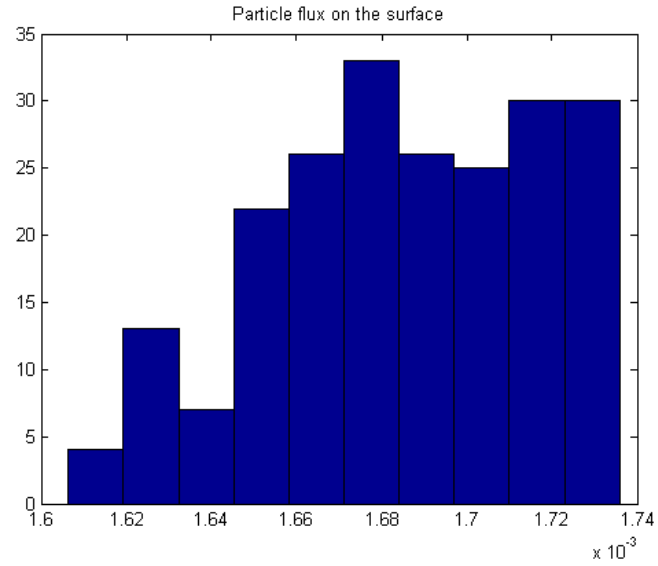
```
            % Score if positive and less than 1
            s_f = [s_f; dist_f(j)];
        end
    end
    % Score particle current
    curr(i) = length(s_0)+length(s_f);
    % Start surface flux
    mu = [];
    % Cycle intersections in s_0
    for j=1:length(s_0)
        s = s_0(j);
        p = start + s*r_0;
        mu = [mu; (p'*r_0) / (norm(p)*norm(r_0))];
    end
    % Cycle intersections in s_f
    for j=1:length(s_f)
        s = s_f(j);
        p = coll + s*r_f;
        mu = [mu; (p'*r_f) / (norm(p)*norm(r_f))];
    end
    % Score surface flux
    summation = 0;
    for j=1:length(mu)
        summation = summation + 1/abs(mu(j));
    end
    flux_s(i) = (1/(A))*summation;
    % Start volume flux
    p_0 = start + s_0(1)*r_0;
    p_f = coll  + s_f(1)*r_f;
    Tl = norm(coll - p_0) + norm(p_f - coll);
    % Score volume flux
    flux_v(i) = (1/(V))*Tl;
    % Energy deposition score
    energy(i) = (rho_a/(rho_g*V)) * (Tl*E_l(i)^2);
    % Pulse height tally
    pulse(i) = E_l(i);
end
% Plot histogram for each tally type
figure;
hist(curr);
title('Particle current on the surface');
figure;
hist(flux_s);
title('Particle flux on the surface');
figure;
```

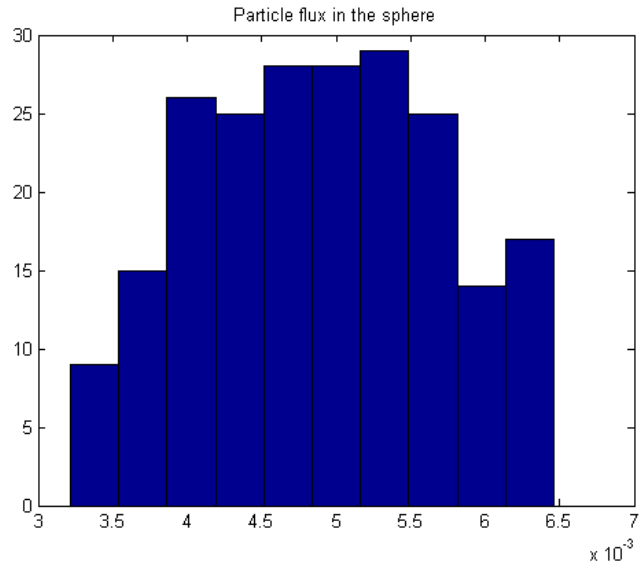
```
hist(flux_v);
title('Particle flux in the sphere');
figure;
hist(energy);
title('Energy deposition in the sphere');
figure;
hist(pulse);
title('Pulse height tally in the sphere');
% Mean value for each tally type
currMean   = mean(curr);
fluxsMean  = mean(flux_s);
fluxvMean  = mean(flux_v);
energyMean = mean(energy);
pulseMean  = mean(pulse);
% Relative error for each tally type
currRel   = sumsqr(curr)/sum(curr)^2 - 1/N;
fluxsRel  = sumsqr(flux_s)/sum(flux_s)^2 - 1/N;
fluxvRel  = sumsqr(flux_v)/sum(flux_v)^2 - 1/N;
energyRel = sumsqr(energy)/sum(energy)^2 - 1/N;
pulseRel  = sumsqr(pulse)/sum(pulse)^2 - 1/N;
% Print all statistics
fprintf('Particle current on the surface:\n');
fprintf('\tMean      = %1.5E\n\tRel Error = %1.5E\n',currMean,currRel);
fprintf('Particle flux on the surface:\n');
fprintf('\tMean      = %1.5E\n\tRel Error = %1.5E\n',fluxsMean,fluxsRel);
fprintf('Particle flux in the sphere:\n');
fprintf('\tMean      = %1.5E\n\tRel Error = %1.5E\n',fluxvMean,fluxvRel);
fprintf('Energy deposition in the sphere:\n');
fprintf('\tMean      = %1.5E\n\tRel Error = %1.5E\n',...
    energyMean,energyRel);
fprintf('Pulse height tally in the sphere:\n');
fprintf('\tMean      = %1.5E\n\tRel Error = %1.5E\n',pulseMean,pulseRel);
```

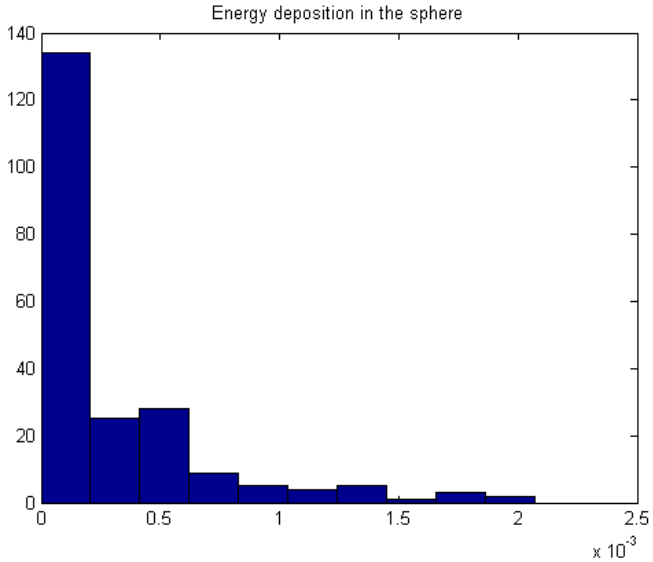
```
Particle current on the surface:
Mean      = 2.00000E+00
Rel Error = 0.00000E+00
Particle flux on the surface:
Mean      = 1.68535E-03
Rel Error = 1.67918E-06
Particle flux in the sphere:
Mean      = 4.89125E-03
Rel Error = 1.25487E-04
Energy deposition in the sphere:
Mean      = 2.82249E-04
```

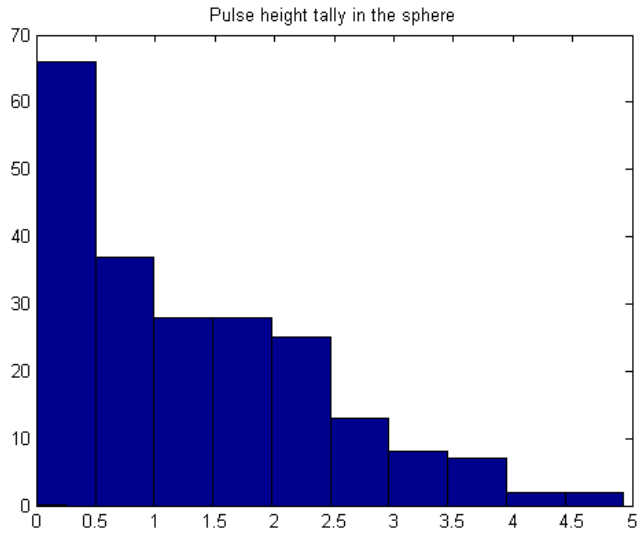
```
Rel Error = 9.40836E-03
Pulse height tally in the sphere:
Mean      = 1.30246E+00
Rel Error = 3.23178E-03
```






Particle flux in the sphere
30
25
20
15
10
5
0
3
3.5
4
4.5
5
5.5
6
6.5
7
x 10^-3


Energy deposition in the sphere
140
120
100
80
60
40
20
0
0
0.5
1
1.5
2
2.5
x 10^-3


Pulse height tally in the sphere
70
60
50
40
30
20
10
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5