Problem 0

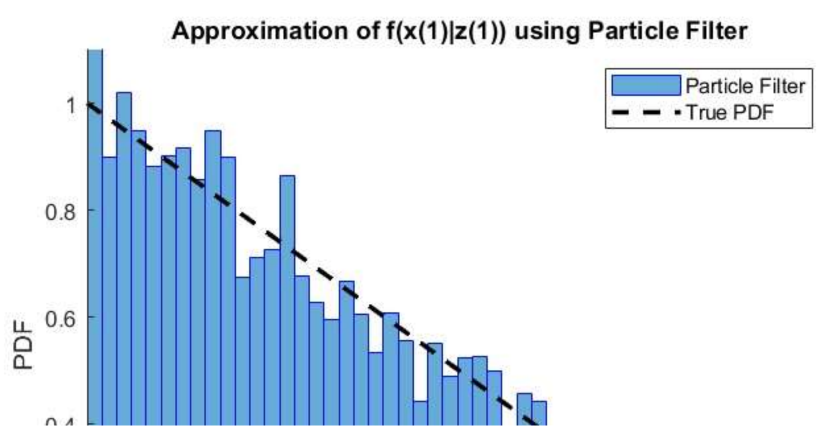
Tuesday, April 4, 2023 1:33 AM

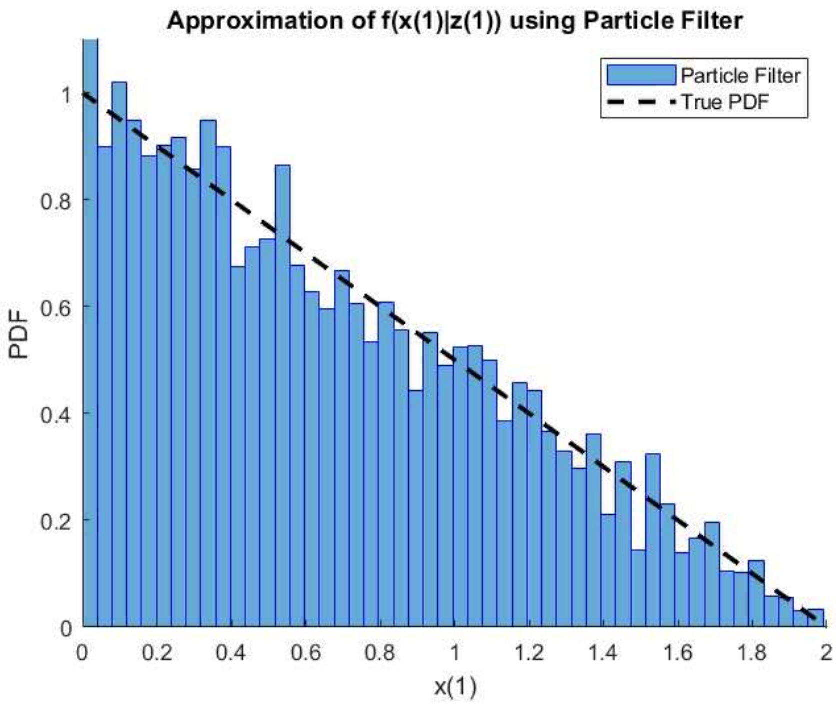
I worked on this homework assignment alone.

Problem 1

Tuesday, March 28, 2023 1:05 PM

```
N = 10^4; % Number of particles
xm_0 = unifrnd(-1,1,1,N);
v_0 = unifrnd(-1,1,1,N);
w_1 = unifrnd(-1,1,1,N);
xp_1 = zeros(1,N);
% f(z|x) = f_w(z-x|x) (trivial COV)
fzx = zeros(1,N);
for i = 1:1:N
   xp_1(i)= xm_0(i)+v_0(i);
   if xp_1(i)>=0 && xp_1(i)<=2
      fzx(i) = 0.5;
   else
      fzx(i) = 0;
   end
end
beta = fzx ./ sum(fzx);

resample_index = randsample(1:N, N, true, beta);  % Draw N samples with replacement based on the
weights
xm_1 = xp_1(resample_index);

% Set the number of histogram bins and the range of x values to plot
num_bins = 50;
x_min = 0;
x_max = 2;
% Plot the histogram of resampled particles
h = histogram(xm_1, num_bins, 'Normalization', 'pdf');
hold on
h.EdgeColor = 'Blue';
% Add a line plot of the true conditional PDF
x_vals = linspace(x_min, x_max);
y_vals = 0.5*(2-x_vals);
plot(x_vals, y_vals, 'LineWidth', 2, 'Color', 'k', 'LineStyle', '--')
% Add labels and adjust the plot appearance
xlabel('x(1)')
ylabel('PDF')
title('Approximation of f(x(1)|z(1)) using Particle Filter')
legend('Particle Filter', 'True PDF')
ylim([0 1.1])
xlim([x_min x_max])
box off
```





We can clearly see a very close agreement between the particle filter estimate and the true PDF of the system, as we would expect. We can also see the effects of the noise and randomness on the particle filter estimate by observing the overshoot and undershoot of some of the bins. If we were to increase the number of particles, we would be able to get a more accurate estimation of the true PDF

Problem 2

Tuesday, March 28, 2023 1:09 PM

```
Np = [10, 10^2, 10^3];
NumRuns = 10^3;
estim = zeros(NumRuns, 4);
z_1 = 0.5;
for i = 1:3
   for k = 1:NumRuns
      xm_0 = unifrnd(-1, 1, 1, Np(i));
      v_0 = unifrnd(-1, 1, 1, Np(i));
      w_1 = unifrnd(-1, 1, 1, Np(i));
      xp_1 = zeros(1, Np(i));
      fzx = zeros(1, Np(i));

      for j = 1:Np(i)
         xp_1(j) = xm_0(j)^3 + v_0(j);
         if xp_1(j)^3 >= -1+z_1 && xp_1(j)^3 <= 1+z_1
            fzx(j) = 1/(2*z_1+2);  % Uniform PDF
         else
            fzx(j) = 0;
         end
      end

      beta = fzx ./ sum(fzx);
      resample_index = randsample(1:Np(i), Np(i), true, beta);
      xm_1 = xp_1(resample_index);
      estim(k, i) = mean(xm_1);
   end
end
% Plot histograms
figure;
histogram(estim(:,1), 50, 'Normalization', 'pdf', 'FaceAlpha', 0.5, 'EdgeColor', 'blue');
hold on;
histogram(estim(:,2), 50, 'Normalization', 'pdf', 'FaceAlpha', 0.5, 'EdgeColor', 'red');
histogram(estim(:,3), 50, 'Normalization', 'pdf', 'FaceAlpha', 0.5, 'EdgeColor', 'yellow');
legend('Np = 10', 'Np = 100', 'Np = 1000');
xlabel('x');
ylabel('PDF');
title('Distribution of Final Estimate for Different Numbers of Particles');
for i = 1:3
   fprintf('Np = %d:\n', Np(i))
   fprintf('Mean = %.4f\n', mean(estim(:, i)))
   fprintf('Std Deviation = %.4f\n\n', std(estim(:, i)))
end
```

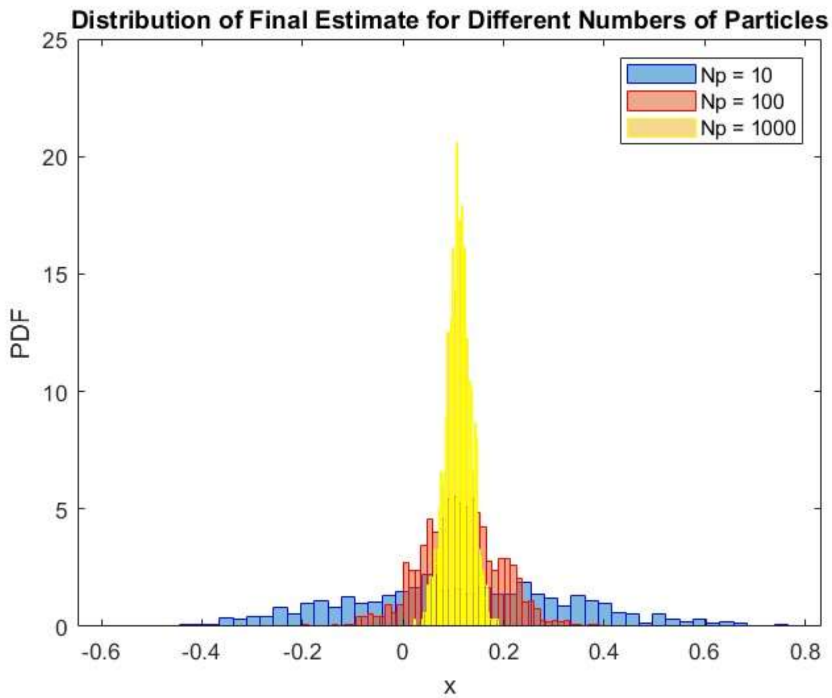
Output:

Np = 10:
Mean = 0.1143
Std Deviation = 0.2307

Np = 100:
Mean = 0.1134
Std Deviation = 0.0779

Np = 1000:
Mean = 0.1107

Std Deviation = 0.0243



Increasing the number of particles leads to a more accurate representation of the PDF under study, hence a more accurate estimate and a smaller sample deviation. This is reflected in the results above.

Problem 3

Tuesday, March 28, 2023 1:12 PM

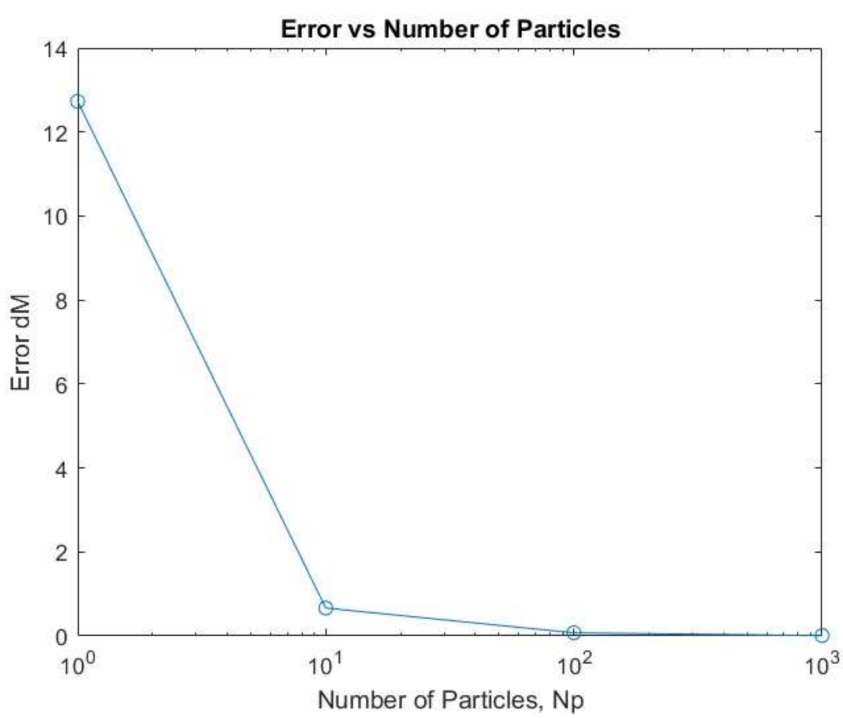
```
%% General to all
z = [1, 0.5, 1.5, 1, 1.5];
sigww = 1;
Np = [1,10,100,1000];
%% Kalman Filter, a-i
A = 1;
H = 1;
xm_k_1 = 0;
pm_k_1 = 1;
sigvv = 1;
for j = 1:1:5
        % Simulate System:
        xp_k = A*xm_k_1;
        pp_k = A*pm_k_1*A'+sigvv;
        k_k = pp_k*H/((H*pp_k*H'+sigww));
        xm_k = xp_k+k_k*(z(j)-H*xp_k);
        pm_k = (1-k_k*H)*pp_k*(1-k_k*H)'+k_k*sigww*k_k';
        xm_k_1 = xm_k;
        pm_k_1 = pm_k;
end
%% Particle Filter a-i
NumRuns = 10^2;
estim = zeros(NumRuns, 4);
time = estim;
MD = estim;
for i = 1:4
    for k = 1:NumRuns
        tic
        xm_0 = normrnd(0,1, 1, Np(i));
        xp_1 = zeros(1, Np(i));
        fzx = zeros(1, Np(i));
        for l = 1: 5
            v_0 = normrnd(0,1, 1, Np(i));
            w_1 = normrnd(0,1, 1, Np(i));
            for j = 1:Np(i)
                xp_1(j) = xm_0(j) + v_0(j);
                fzx(j) =  normpdf(z(l) - xm_0(j), 0, sqrt(1)); % Measurement update
            end

        beta = fzx ./ sum(fzx);
        resample_index = randsample(1:Np(i), Np(i), true, beta);
        xm_1 = xp_1(resample_index);
        xm_0 = xm_1;
        end
        time(k,i) = toc;
        estim(k, i) = mean(xm_1);
        MD(k,i) = (xm_k-estim(k,i))'*inv(pm_k_1)*(xm_k-estim(k,i));

    end
end
%% Plot a-i
dM = mean(MD,1);
comptime = mean(time,1);
figure(1)
semilogx(Np, dM, '-o')
xlabel('Number of Particles, Np')
ylabel('Error dM')
title('Error vs Number of Particles')
```

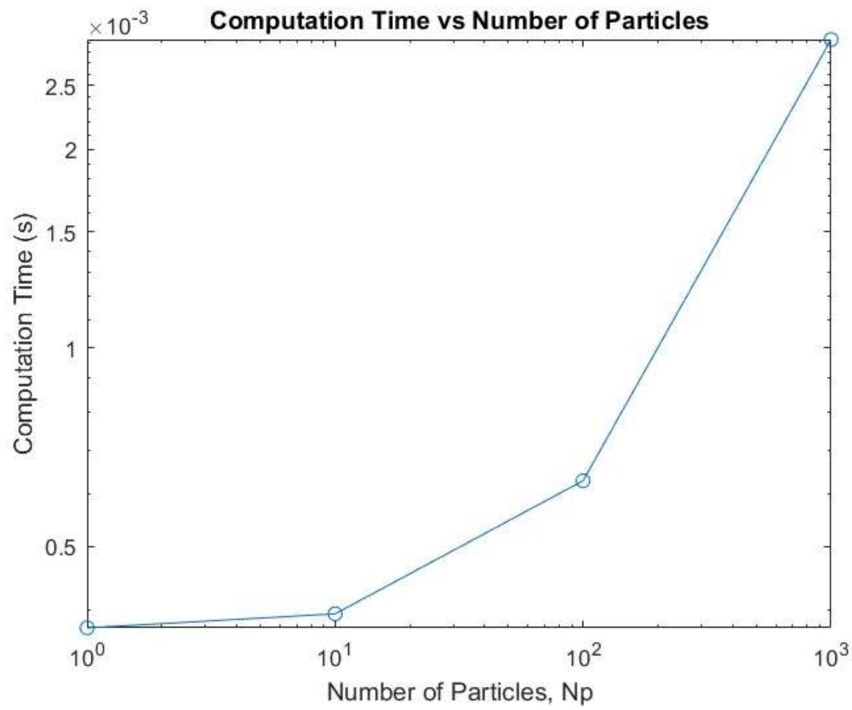
```
% Create the second plot
figure(2)
loglog(Np, comptime, '-o')
xlabel('Number of Particles, Np')
ylabel('Computation Time (s)')
title('Computation Time vs Number of Particles')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Kalman Filter, a-ii
A = [1,1;0,1];
H = [1,0];
xm_k_1 = [0;0];
pm_k_1 = [1,0;0,1];
sigvv = 1;
B = [0;1];
for j = 1:1:5
        % Simulate System:
        xp_k = A*xm_k_1;
        pp_k = A*pm_k_1*A'+B*sigvv*B';
        k_k = pp_k*H'*inv((H*pp_k*H'+sigww));
        xm_k = xp_k+k_k*(z(j)-H*xp_k);
        pm_k = (1-k_k*H)*pp_k*(1-k_k*H)'+k_k*sigww*k_k';
        xm_k_1 = xm_k;
        pm_k_1 = pm_k;
end
%% Particle Filter a-ii
NumRuns = 10^2;
estim1 = zeros(NumRuns, 4);
estim2 = estim1;
time = estim;
MD = estim;
for i = 1:4
    for k = 1:NumRuns
        tic
        xm_01 = normrnd(0,1, 1, Np(i));
        xm_02 = normrnd(0,1, 1, Np(i));
        xp_1 = zeros(2, Np(i));
        fzx = zeros(1, Np(i));
            for l = 1: 5
                v_0 = normrnd(0,1, 1, Np(i));
                w_1 = normrnd(0,1, 1, Np(i));
                for j = 1:Np(i)
                    xp_1(1,j) = xm_01(j) + xm_02(j);
                    xp_1(2,j) = xm_02(j)+v_0(j);
                    fzx(j) =  normpdf(z(l) - xm_01(j), 0, sqrt(1)); % Measurement update
                end

                beta = fzx ./ sum(fzx);
                resample_index = randsample(1:Np(i), Np(i), true, beta);
                xm_11 = xp_1(1,resample_index);
                xm_12 = xp_1(2,resample_index);
                xm_01 = xm_11;
                xm_02 = xm_12;
            end
        time(k,i) = toc;
        estim1(k, i) = mean(xm_11);
        estim2(k,i) = mean(xm_12);
        MD(k,i) = (xm_k-[estim1(k,i);estim2(k,i)])'*inv(pm_k_1)*(xm_k-[estim1(k,i);estim2(k,i)]);

    end
end
%% Plot a-ii
dM = mean(MD,1);
comptime = mean(time,1);
figure(3)
```

```
semilogx(Np, dM, '-o')
xlabel('Number of Particles, Np')
ylabel('Error dM')
title('Error vs Number of Particles')
% Create the second plot
figure(4)
loglog(Np, comptime, '-o')
xlabel('Number of Particles, Np')
ylabel('Computation Time (s)')
title('Computation Time vs Number of Particles')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%% Kalman Filter, a-iii
A = [1,1,0,0;0,1,1,0;0,0,1,1;0,0,0,1];
H = [1,0,0,0];
xm_k_1 = [0;0;0;0];
pm_k_1 = eye(4);
sigvv = 1;
B = [0;0;0;1];
for j = 1:1:5
        % Simulate System:
        xp_k = A*xm_k_1;
        pp_k = A*pm_k_1*A'+B*sigvv*B';
        k_k = pp_k*H'*inv((H*pp_k*H'+sigww));
        xm_k = xp_k+k_k*(z(j)-H*xp_k);
        pm_k = (1-k_k*H)*pp_k*(1-k_k*H)'+k_k*sigww*k_k';
        xm_k_1 = xm_k;
        pm_k_1 = pm_k;
end
%% Particle Filter a-iii
NumRuns = 10^2;
estim1 = zeros(NumRuns, 4);
estim2 = estim1;
estim3 = estim1;
estim4 = estim1;
time = estim;
MD = estim;
for i = 1:4
    for k = 1:NumRuns
        tic
        xm_01 = normrnd(0,1, 1, Np(i));
        xm_02 = normrnd(0,1, 1, Np(i));
        xm_03 = normrnd(0,1, 1, Np(i));
        xm_04 = normrnd(0,1, 1, Np(i));
        xp_1 = zeros(4, Np(i));
        fzx = zeros(1, Np(i));
            for l = 1: 5
                v_0 = normrnd(0,1, 1, Np(i));
                w_1 = normrnd(0,1, 1, Np(i));
                for j = 1:Np(i)
                    xp_1(1,j) = xm_01(j) + xm_02(j);
                    xp_1(2,j) = xm_03(j) + xm_02(j);
                    xp_1(3,j) = xm_03(j) + xm_04(j);
                    xp_1(4,j) = xm_04(j)+v_0(j);
                    fzx(j) =  normpdf(z(l) - xm_01(j), 0, sqrt(1)); % Measurement update
                end

                beta = fzx ./ sum(fzx);
                resample_index = randsample(1:Np(i), Np(i), true, beta);
                xm_11 = xp_1(1,resample_index);
                xm_12 = xp_1(2,resample_index);
                xm_13 = xp_1(3,resample_index);
                xm_14 = xp_1(4,resample_index);
                xm_01 = xm_11;
```

```
            xm_02 = xm_12;
            xm_03 = xm_13;
            xm_04 = xm_14;
          end
        time(k,i) = toc;
        estim1(k, i) = mean(xm_11);
        estim2(k,i) = mean(xm_12);
        estim3(k,i) = mean(xm_13);
        estim4(k,i) = mean(xm_14);
        MD(k,i) = (xm_k-[estim1(k,i);estim2(k,i);estim3(k,i);estim4(k,i)])'*inv(pm_k_1)*(xm_k-
[estim1(k,i);estim2(k,i);estim3(k,i);estim4(k,i)]);

    end
end
%% Plot a-iii
dM = mean(MD,1);
comptime = mean(time,1);
figure(5)
semilogx(Np, dM, '-o')
xlabel('Number of Particles, Np')
ylabel('Error dM')
title('Error vs Number of Particles')
% Create the second plot
figure(6)
loglog(Np, comptime, '-o')
xlabel('Number of Particles, Np')
ylabel('Computation Time (s)')
title('Computation Time vs Number of Particles')
```

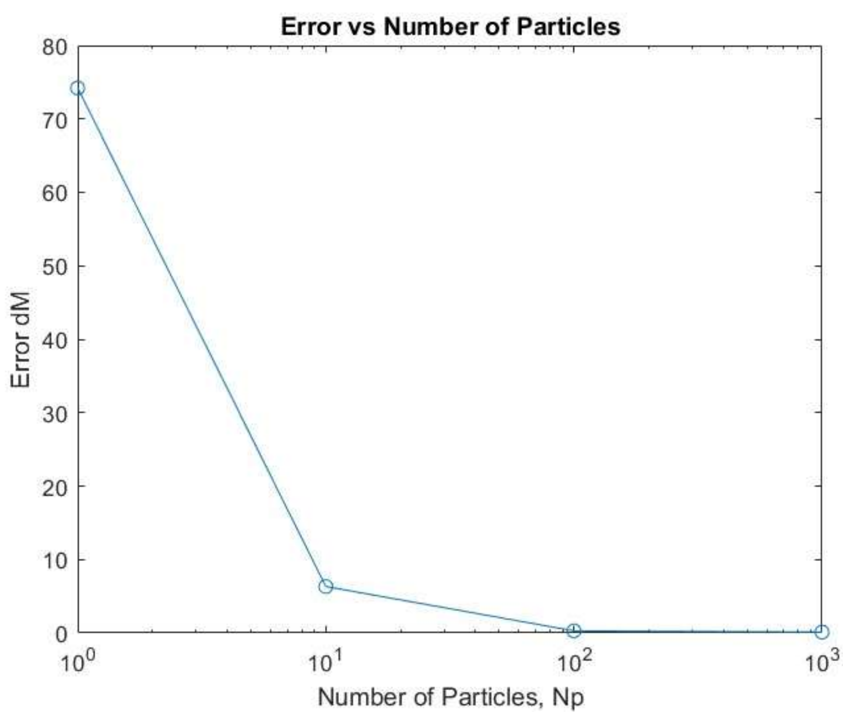
a-i:



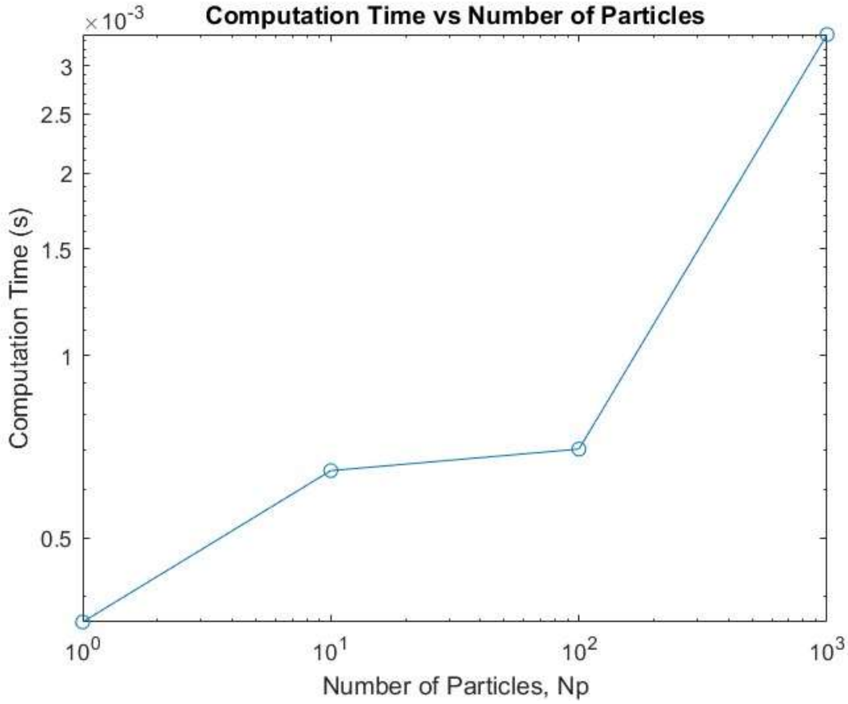


Computation Time vs Number of Particles
×10^-3
2.5
2
1.5
1
0.5
Computation Time (s)
10^0
10^1
10^2
10^3
Number of Particles, Np

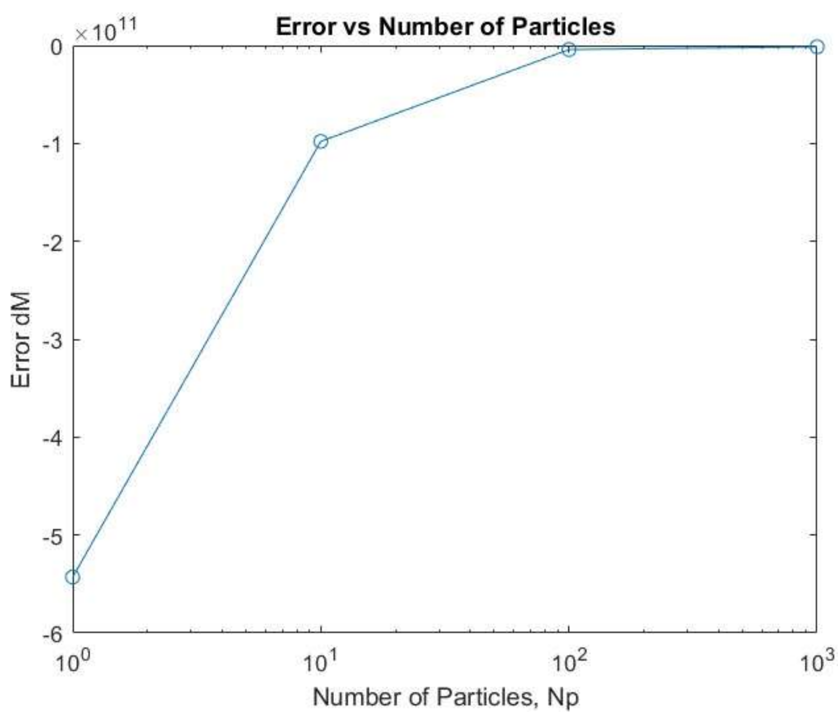
a-ii:

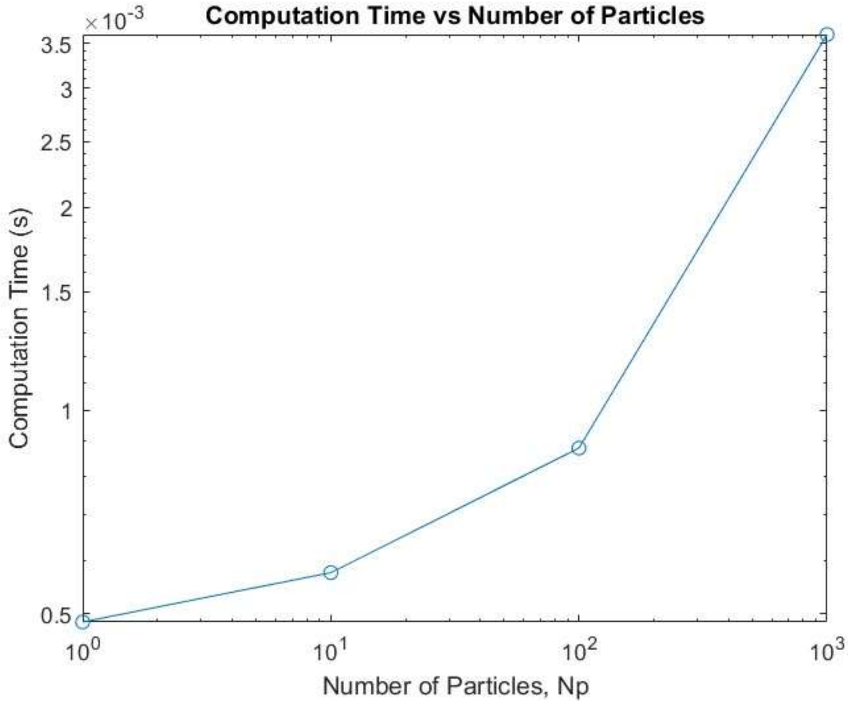


Error vs Number of Particles
80
70
60
50
40
30
20
10
0
Error dM
10^0
10^1
10^2
10^3
Number of Particles, Np

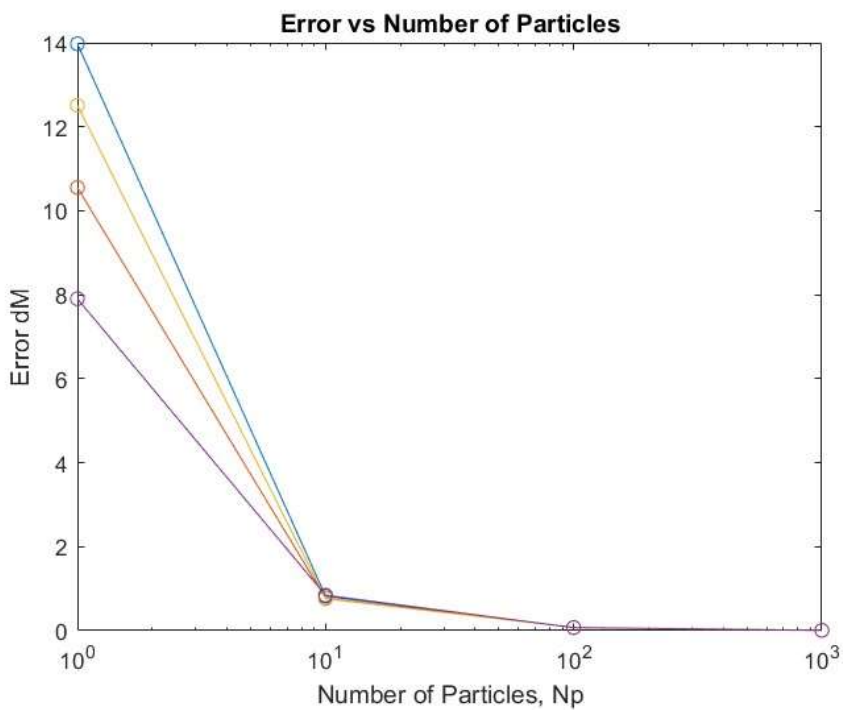

Computation Time vs Number of Particles
$\times 10^{-3}$
3
2.5
2
1.5
1
0.5
Computation Time (s)
10^0
10^1
10^2
10^3
Number of Particles, Np

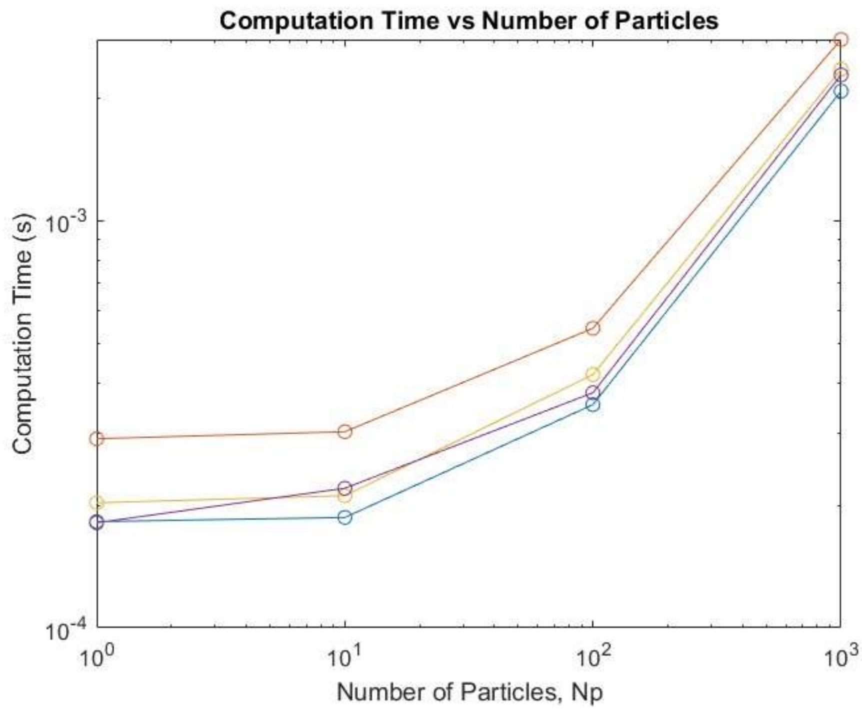
a-iii:

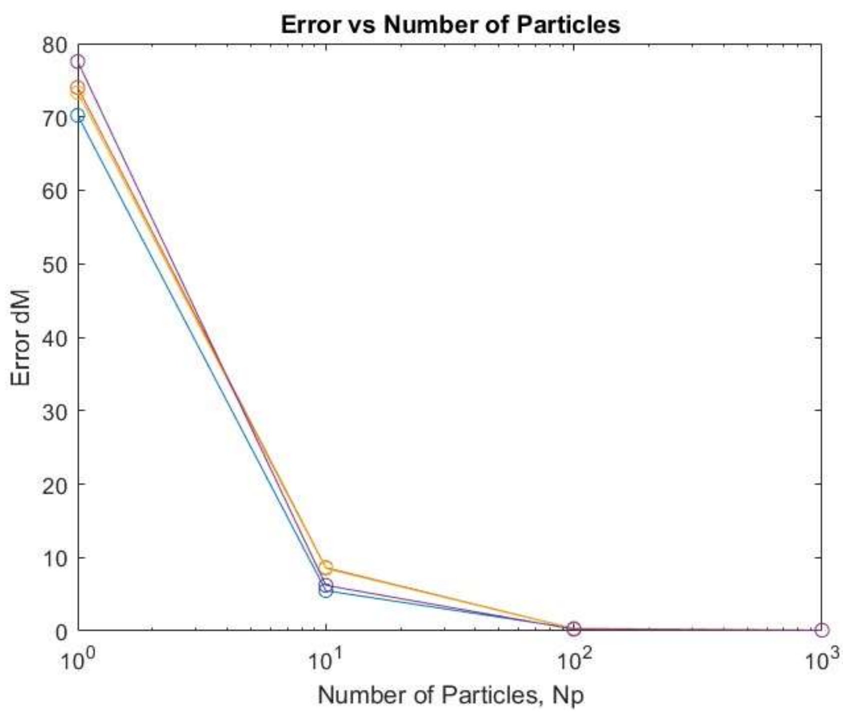

Error vs Number of Particles
$\times 10^{11}$
0
-1
-2
-3
-4
-5
-6
Error dM
10^0
10^1
10^2
10^3
Number of Particles, Np

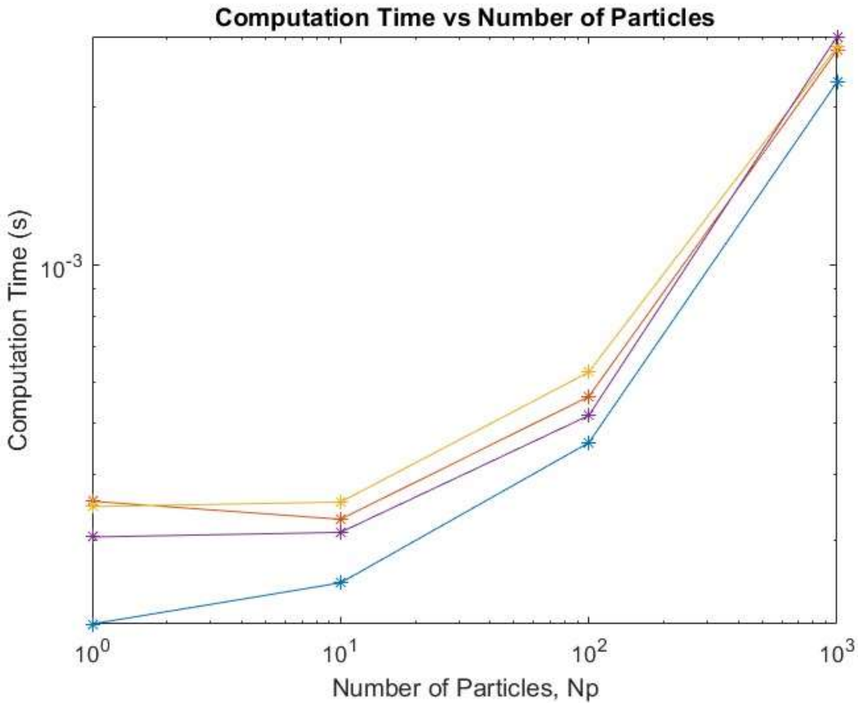


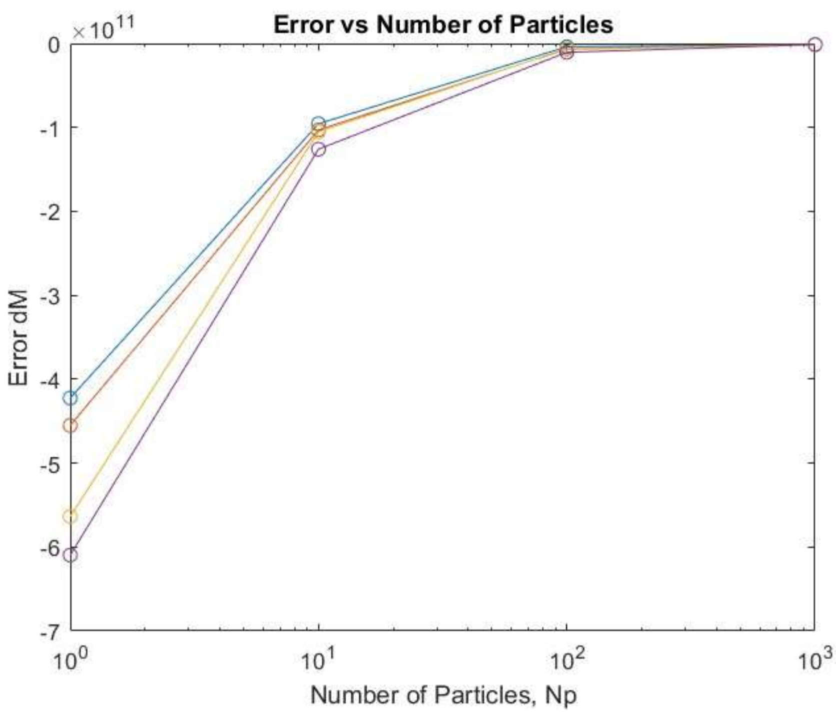
b-i-The graphs change when we rerun the scripts as seen below. They are clearly sensitive to noise, more drastically so when Np is smaller

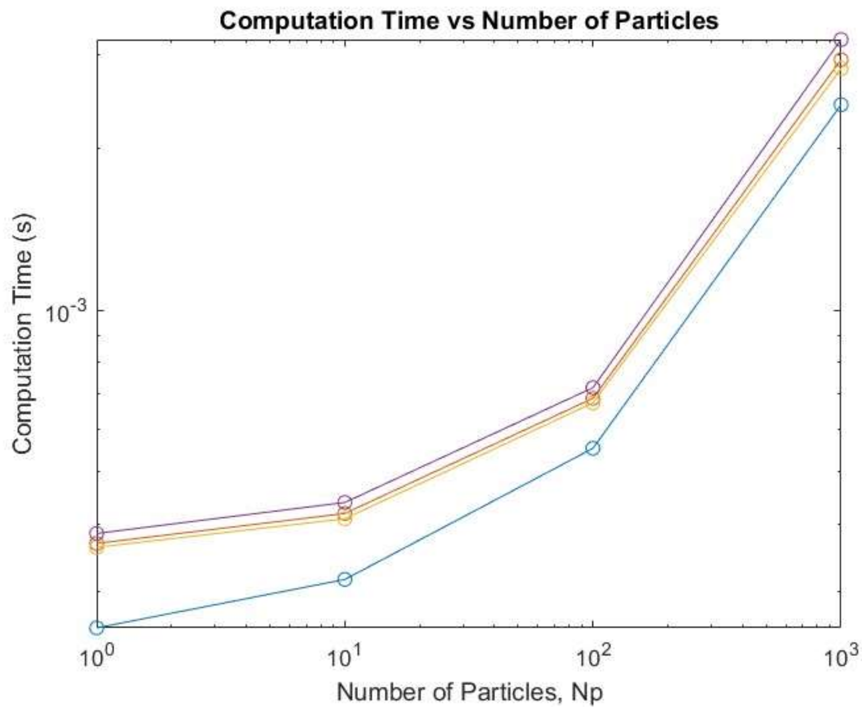



Computation Time vs Number of Particles
Computation Time (s)
10^{-3}
10^{-4}
10^0
10^1
10^2
10^3
Number of Particles, Np


Error vs Number of Particles
Error dM
80
70
60
50
40
30
20
10
0
10^0
10^1
10^2
10^3
Number of Particles, Np


Computation Time vs Number of Particles
Computation Time (s)
10^{-3}
10^0
10^1
10^2
10^3
Number of Particles, Np


Error vs Number of Particles
$\times 10^{11}$
0
-1
-2
-3
-4
-5
-6
-7
Error dM
10^0
10^1
10^2
10^3
Number of Particles, Np



b-ii- The trends match what one would expect, with lower error at higher number of particles and more compute time as well.