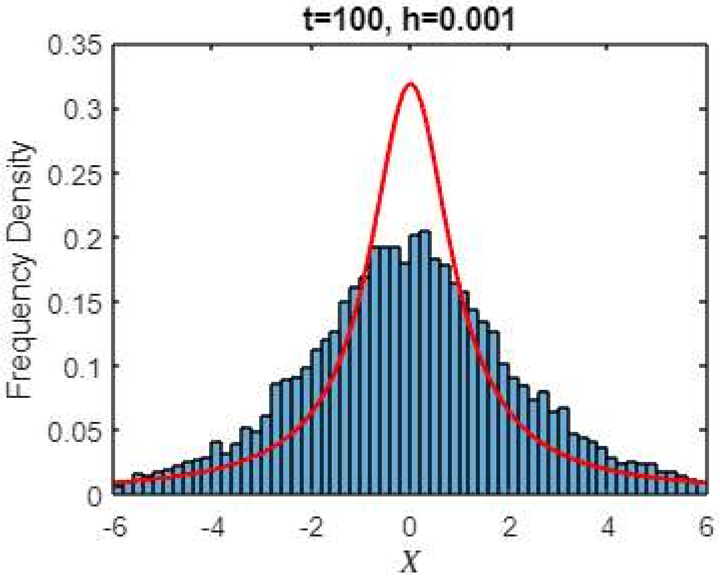
```
clear

N=100000;
h =.001;
D = 10^-4;

% Using batches makes the angles array smaller
batch=30;
batch_repeat=400;
trials=batch*batch_repeat;
angles = zeros(N,batch);
steps = zeros(N,2,batch);
```

Ideally, the walks all start at $(r_0, 0)$, but unfortunately the results are much worse this way. I am unsure why. Still, r_0 is used in plotting the distribution below.

```
r0 = 0.8;

T = zeros(1,trials);
for j = 1:batch_repeat
    % steps(1,:,:) = repmat([r0,0],1,1,batch);
    % steps(2:end,:,:) = sqrt(2*D*h)*randn(N-1,2,batch);
    steps = sqrt(2*D*h)*randn(N,2,batch);

    % Sum the rows to get the cumulative walk
    walk = cumsum(steps, 1);

    complex_walk = reshape(complex(walk(:,1,:),walk(:,2,:)),[N,batch]);

    % Get angles of each complex-valued location
    angles = angle(complex_walk);

    % Unwrap the angles to get the "unmodulated" angles for each trial
    angles = unwrap(angles,[],1);

    % Keep the final, total angle
    % We could instead bin these so we don't have to store angular values.
    T((j-1)*batch + 1 : j*batch)=angles(end,:);
end
```

```
% Apply scaling
t = N*h;
% Use the exact version of B(t) since t still isn't exceedingly large
X = (1/log(2*sqrt(t)/(r0*exp(.577/2))))*T;
%X = (2/log(t))*T;

histogram(X,'Normalization','pdf',BinWidth=0.2);
hold on;
% Compare with Spitzer's result
a = -6;
```

```
b = 6;
P=@(x)1./(pi*(1+x.^2));

x=linspace(a,b,1000);
plot(x,P(x),"LineStyle","-","Color","red",'LineWidth',1.5);
axis([a,b,0,0.35]);
title_str = ['t=',num2str(t),', h=',num2str(h)];
title(title_str);
xlabel("$X$","Interpreter","latex");
ylabel("Frequency Density");
hold off;
```


t=100, h=0.001
Frequency Density
X