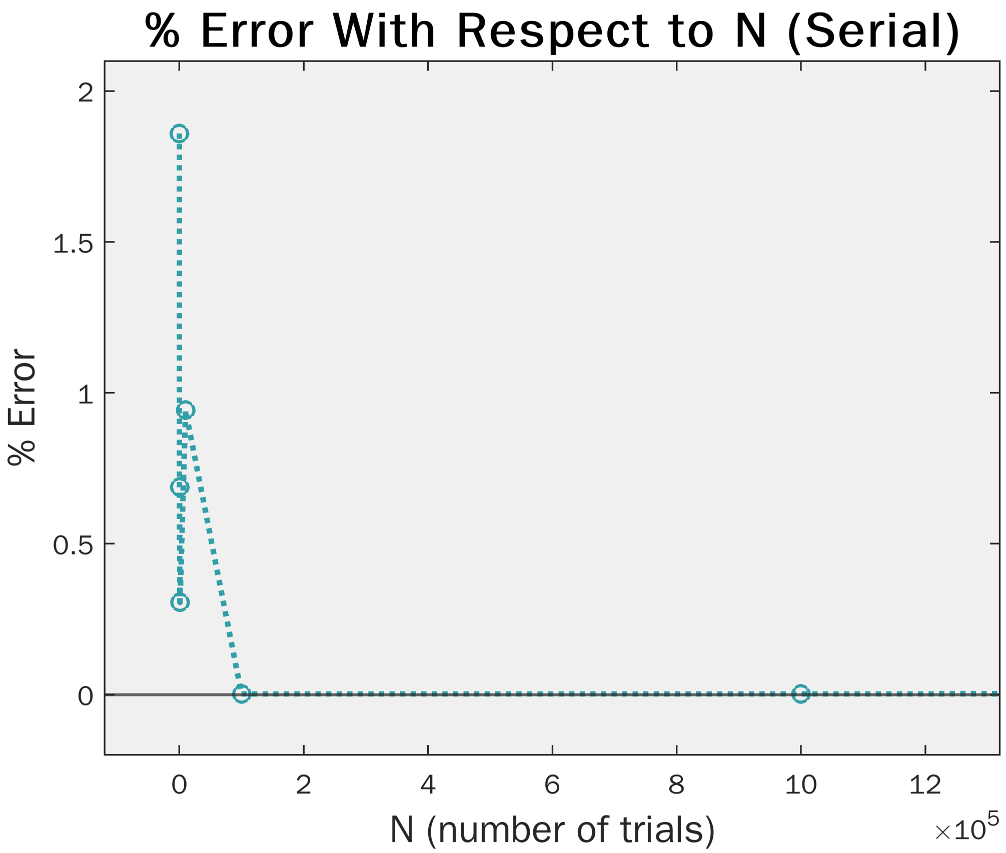
To test the logical validity of my Monte Carlo algorithm, I first implemented it serially to prove that as trials increase, percentage of error from the actual value of Pi decreases.

```
%Read serial trials from csv
sData=readmatrix('serialData.csv');
col1=sData(:,1);
col2=sData(:,2);

%Plot data with a line on the X-axis to visualize convergence
plot(col1,col2,':o','Color',1/255*[50 158 168],'LineWidth',2);
yline(0,'LineWidth',1);
set(gca,'color',1/255*[240 240 240])

%Details
title('% Error With Respect to N (Serial)','FontSize',18);
xlabel('N (number of trials)','FontSize',14);
ylabel('% Error','FontSize',14);
xlim([-120000 1320000])
ylim([-0.2 2.1])
```



It can be observed that as the number of trials increases exponentially, the error flattens at arounf 0 after a value of 10,000 N is read. It is safe to assume that it takes 100,000 trials to consistently generate Pi with accuracy through the C rand() function using a consistent seed. This is a principal aspect of the Monte Carlo method: as the number of trials increases, the higher likelihood the outcome will converge to a value (in this case, Pi).