PHYS 371 - Assignment IX

Celil Buğra Karacan
Deepartment of Physics, Bilkent University
(Dated: December 20, 2018)

I. INTRODUCTION

What assigned was that calculating number π with pseudo-random numbers (Monte Carlo's Methods to be specific). The setup involves random hits onto a square which encircles a circle inside. By assuming uniform hit possibility over the square region, one can compute π by basically calculating the ratio of the areas. Formula given is as follows:

$$\pi = 4\frac{c}{s}$$

, where c and s are number of hits within the circle and the square respectively.

II. METHOD

For creation of pseudo-random numbers MATLAB ®'s `rand()` function was used. By creating two random numbers representing Cartesian coordinates of the hit, one could check whether the hit point was within or out of the circle using the inequality $x^2 + y^2 < 1$. Throw numbers of $[10^3, 5\times10^3, 10^3, 10^4, 5\times10^4, ..., 10^7, 5\times10^7]$ were fed into the script. On top of that for error calculation, k many calculations per a given number of throws (e.g. 10^5) were utilized, which then have been used for calculating standard deviation (σ) per throws. Graphs of average (arithmetic mean) values per throw-nembers with errorbars are below. Lastly, code from MATLAB ®community by Frederic Moisy (`errorbarlogx.m`) was utilized for esthetical reasons.

III. VERIFICATION

On behalf of verification, formula was ,kindly, assigned by the professor. Which, as stated above, is using ratio of areas to compute π, legitimate method I would say. Moreover, graphs seem to converge 3.14... assumably π.

IV. INTERPRETATION

It is evident that as number of throws increase, error decreases significantly. As seen in the Figure 1, standard deviation falls somewhat parobolically.Also, approximation converges to π at the mean time. As k increases, there seems no difference in the error, present differences would better be correlated with the randomization, as seen for three different outcomes for $k = 15$.

V. DATA

Plots for standard deviation and mean values with errorbars are as follows. For comparison one plot of `errorbar()` function alone is included too.

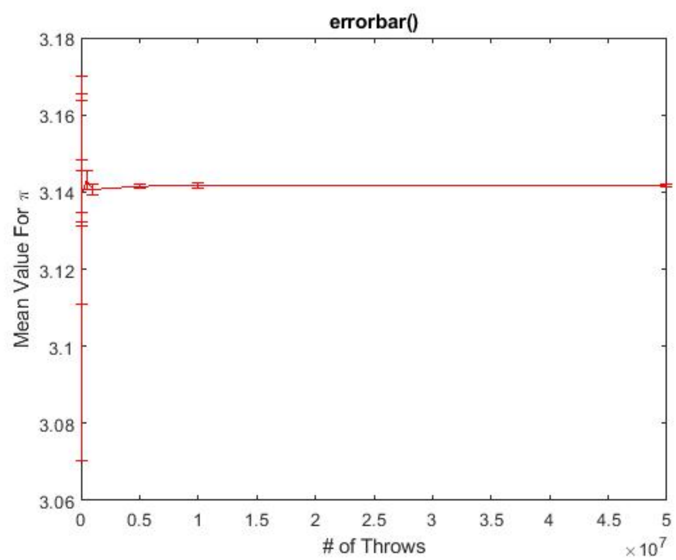

FIG. 1. k = 15 (first)

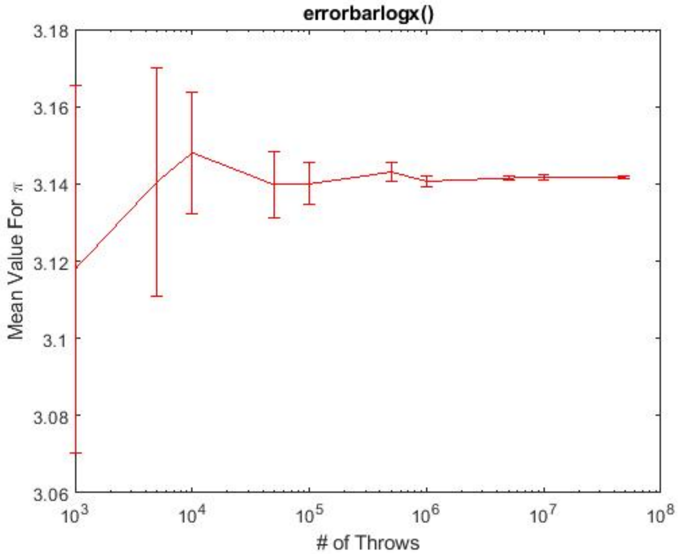

FIG. 2. k = 15 (first)

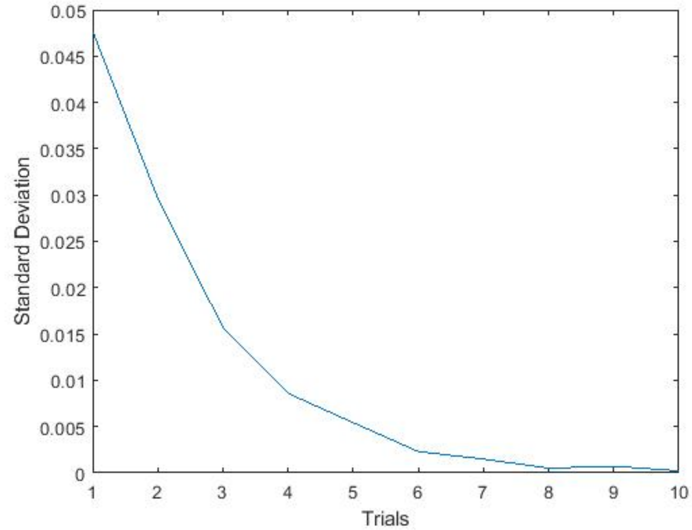

FIG. 4. k = 15 (first)

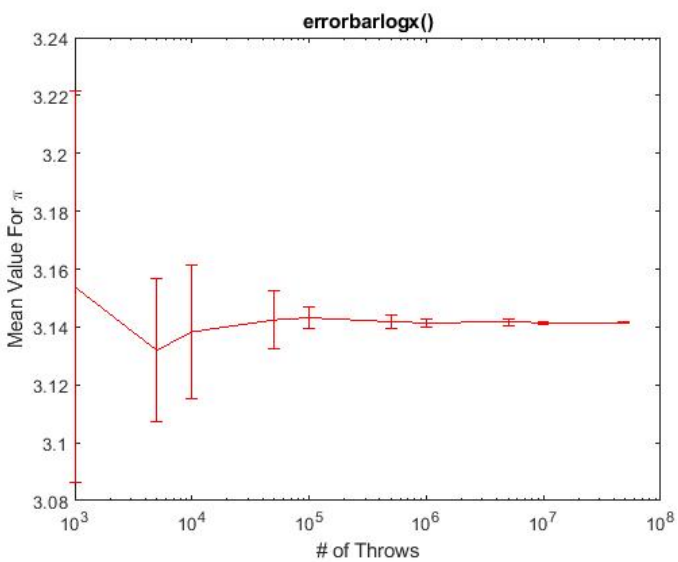

FIG. 3. k = 15 (second)

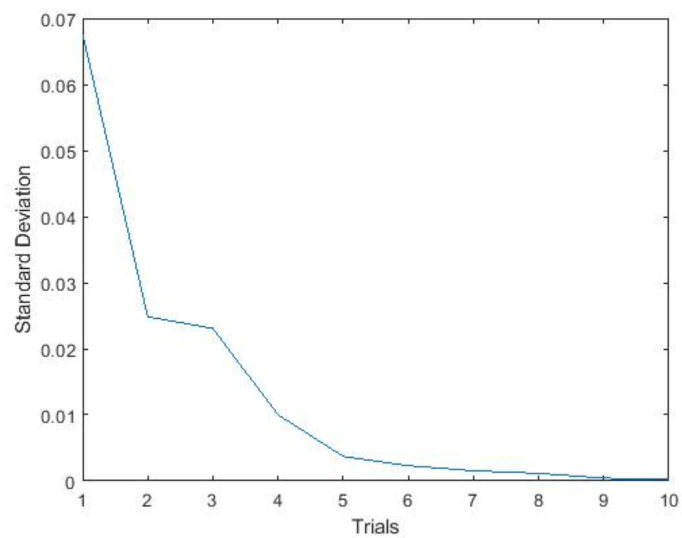

FIG. 5. k = 15 (second)

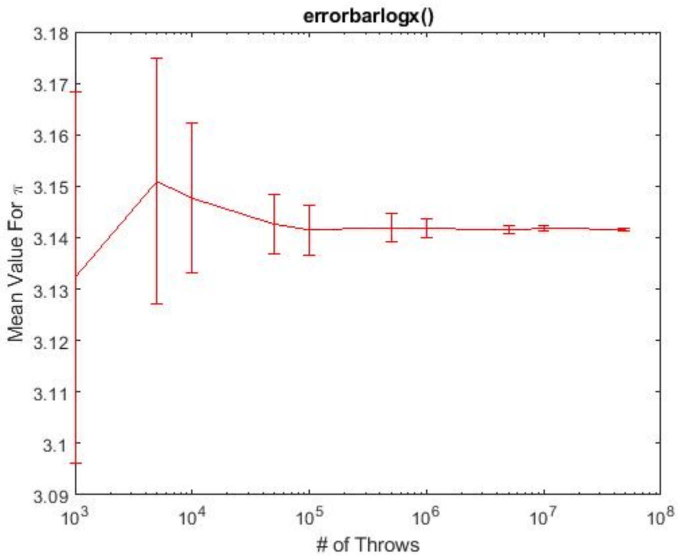

FIG. 6. k = 15 (third)

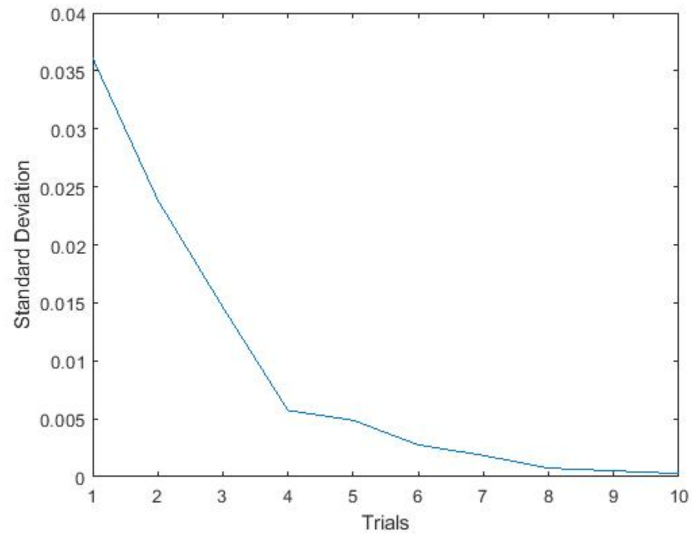

FIG. 8. k = 15 (third)

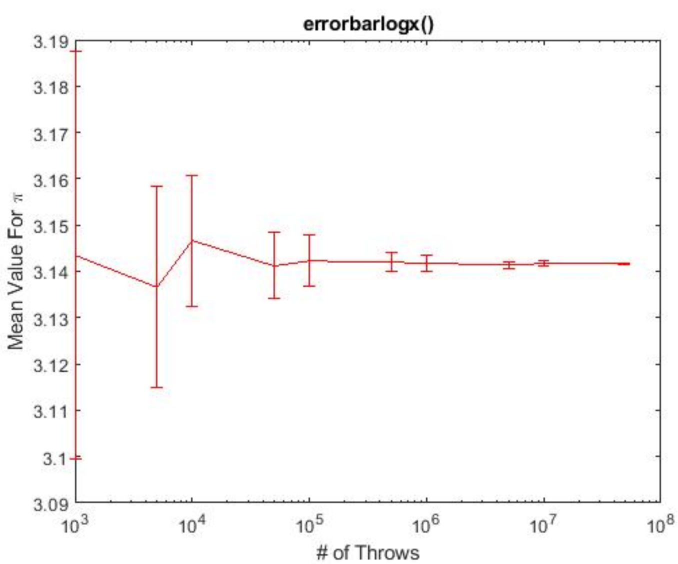

FIG. 7. k = 30

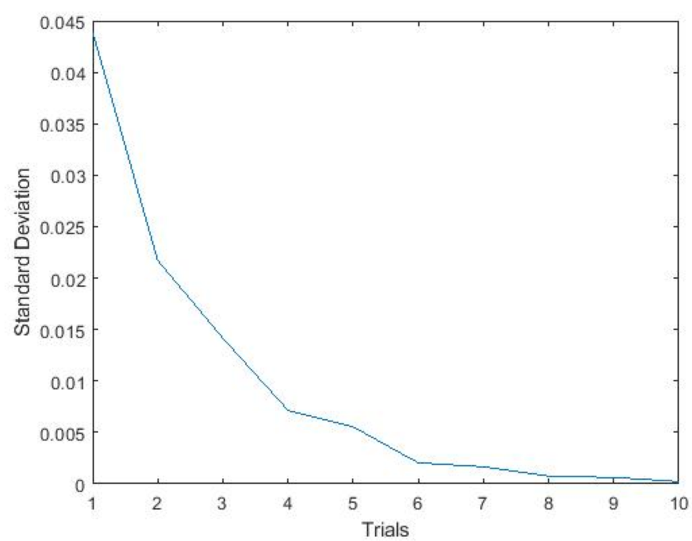

FIG. 9. k = 30

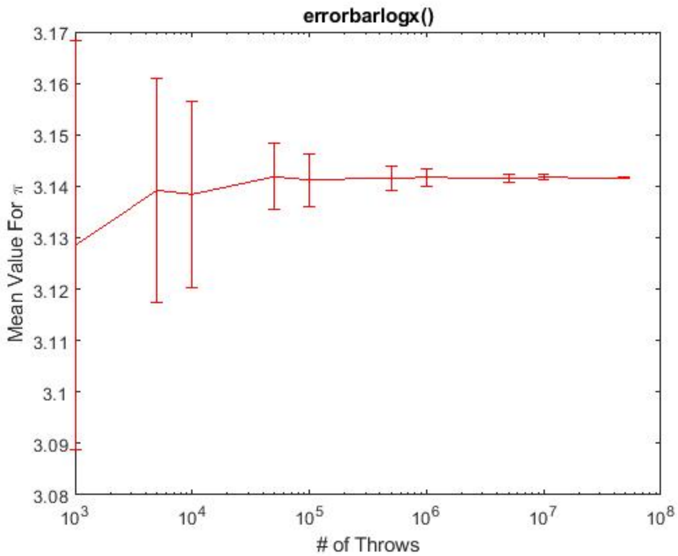

FIG. 10. k = 50

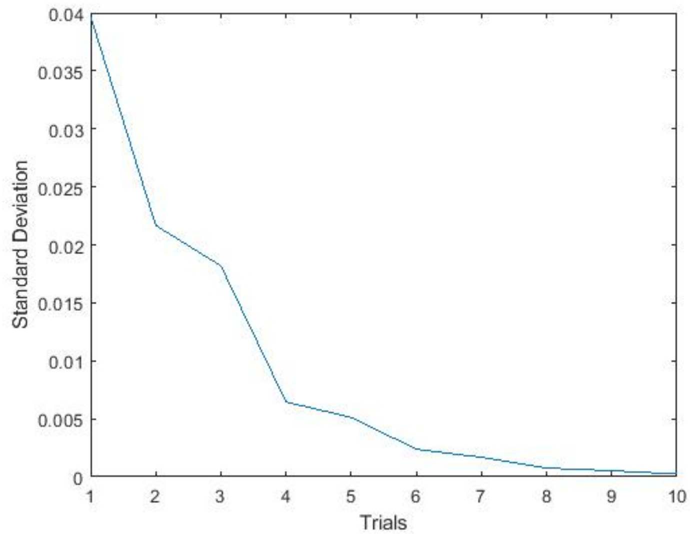

FIG. 11. k = 50

VI. REFERENCES

Moisy, Frederic. ERRORBARLOGX. Reconstructing an Image from Projection Data - MATLAB Simulink Example, MathWorks, 24 Jan. 2006, www.mathworks.com/matlabcentral/fileexchange/9715-errorbarlogx-m.