# Homework #3

Tuesday, February 9

You have been hired by the secretive $\pi$-society to analyze a new strategy for calculating the value of $\pi$ through the development of a Python program. Here is the basic format of the strategy provided to you by the $\pi$-society. Begin by recognizing that the area of a unit square (height = 1.0, width = 1.0) is 1.0 and the area of a quarter circle, which lies entirely within the unit square is $\pi/4$. Now imagine that we choose a random point $(x, y)$ within the unit square. The probability of that random point being within the quarter unit circle is $\frac{\pi}{4} \approx 79\%$ because 79% of the area of the unit square is covered by the quarter unit circle! Now, if we choose 'many' random points, we can estimate the value of $\pi$ because $4 \times (fraction\ inside\ the\ circle) = \pi$. Hence, we just need to select 'many' or $n$ random points (i.e., a point is two random floating point numbers between 0-1.0, see numpy.random.random()), then loop through the $n$ points and check to see if each point is inside the quarter unit circle (i.e., is $\sqrt{x^2 + y^2} < 1.0$), and then count the fraction of the $n$ points inside the quarter unit circle. The figure below shows 3000 random points on the unit square – the red points are inside the quarter unit circle and the blue points are outside.

You should write a memo to the $\pi$-society that summarizes your analysis of the feasibility of this algorithm. The report should include a figure, which may be generated using Excel or matplotlib, that shows the accuracy of the proposed strategy on the y-axis (accuracy = $|\pi - approximation|$) versus the number of random points used on the x-axis. The Python program should gain accuracy with more points. The head of the $\pi$-society casually mentioned that you should try, "maybe 100, or 1000, or 10,000 random points" to see how the number of points impacts the accuracy. You consider possibly trying one or two different numbers of random points beyond those three values. You have also been reminded to include your Python code in the appendix of the memo/report.



$n = 3000 \ (\pi \approx 3.16667)$