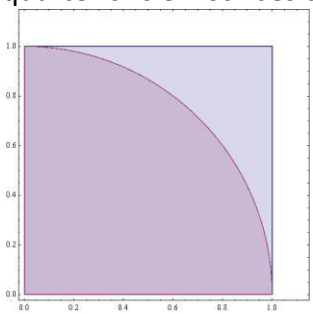
Monte Carlo (MC) and Markov chain Monte Carlo (MCMC) simulations

Problem 1 [Monte Carlo]

Generate n=100 samples of IID 2-dimensional uniform random variables in the unit-square. Count how many of these samples fall within the quarter unit-circle centered at the origin. This quarter circle inscribes the unit square as shown below:



1. Use these random samples to estimate the area of the inscribed quarter circle. Use this area estimate to estimate the value of pi. Do k=50 runs of these pi-estimations. Plot the histogram of the 50 pi-estimates.
2. Repeat the experiment with different numbers of uniform samples, n (using k=50 for all these runs). Plot the sample variance of the pi-estimates for these different values of n. Find the line of best-fit for the relationship between your estimate variance and Monte Carlo sample size.
3. Adapt your Monte Carlo solution to provide integral and error estimates for the function:

$g(x,y) = (x-1)^2 + 100(y-x^2)^2$ (x, y) in [-1, 1] for g(x,y)

Problem 2 [Variance Reduction Methods for Monte Carlo]

Use a total sample budget of n=1000 to obtain Monte Carlo estimates and sample MC estimate variances
for the definite integrals:

(a) Sin(x)/Ln(x), x in [2,10]
(b) Sin[x^2], x in [-2pi, 2pi]

Implement stratification and importance sampling (separately) in the Monte Carlo estimation procedures using the same sample budget n=1000. Compare the 3 different Monte Carlo integral estimates and their sample variances. Discuss the quality of the Monte Carlo estimates from each method.

Problem 3 [Markov Chain Monte Carlo]

The random variable X has a mixture distribution: 60% in a Beta (1,8) distribution and 40% in a Beta (9,1) distribution.

1. Implement a Metropolis-Hastings algorithm to generate samples from this distribution.
2. Run the algorithm multiple times from different initial points. Plot sample paths for the algorithm. Can you tell if/when the algorithm converges to its equilibrium distribution? (Hint: look up MCMC-specific convergence diagnostics. Goodness-of-fit tests on the Markov chain tail samples may also be useful).
3. Plot sample paths for the algorithm using different proposal pdfs. Comment on the effect of low-variance vs high-variance proposal pdfs on the behavior of your algorithm.