EE2012 Project Work

Instructions

You will form a project group of **at most** two persons. You are encouraged to find your group partner within your EE2012 class. Each group will submit **ONE PDF(portable document format) FILE** as the report, and **ONE Matlab/Python FILE** to EE2012 IVLE project submission folder by the deadline. Please use matric number(s) in the file names, such as

• ***matric1_matric2.pdf*** (up to 10 pages)

• ***matric1_matric2.m*** if you use Matlab or ***matric1_matric2.py*** if you use Python (one single file with clear instructions for others to run your script)

The teaching assistant may need to review of your Matlab/Python script together with you if unable to run your script as explained. Therefore, it is important that you provide clear instructions in your Matlab/Python script. In the case if you cannot find a partner to form a group, you may submit the project work individually.

• EE2012 Teaching Assistant: Ms. Berrak Sisman

• Email: berraksisman@u.nus.edu

Submission deadline: 11:59 pm, 12 Nov 2018 (Monday Week 13)

Marking policy: A 10% penalty will be imposed for submissions delayed within one day. A 30% penalty will be imposed for submissions delayed by one to two days. A 50% penalty for submissions delayed by two to three days. Submissions with delay more than three days will not be accepted. Each group should submit only one report.

Project Introduction

In this project, you need to formulate and then write a program to obtain random numbers that have an exponential probability density function (pdf) $f(x) = \lambda e^{-\lambda x} u(x)$ from random numbers that are uniformly distributed over the interval [0,1]. Uniformly distributed random numbers can be generated through the Matlab function "rand" [1], or their Python equivalent, where each call to these functions will return a single random number within the interval [0,1].

To obtain an exponentially distributed random number from the uniformly distributed random numbers obtained from the generator, a transformation based on the inverse transformation method given in [2] can be used. Essentially, given that X is uniformly distributed, one needs to work out $g(X)$ such that $Y = g(X)$ has the desired pdf. Passing any realization of X through $g(X)$ will then give rise to another r.v. with the desired distribution. In this project, you need to develop a computer program (Matlab/ Python script) that gives random numbers with an exponential distribution.

Problems

(1) Exponentially distributed random variables are often encountered in modelling the time between occurrence of events. Here, an engineer is building a model on failure analysis, and it is assumed that the lifetime T (in year) for an electronic component follows an exponential distribution with a pdf of $f(t) = \lambda e^{-\lambda t} u(t)$, where λ is a parameter related to the average lifespan E[T]. For E[T] = 3, 4 and 5, please report the following:

a. Workout the pdf and cdf.

b. Calculate λ.

c. With Matlab or Python, plot the pdf and cdf for E[T]=3,4,5 in the same figure.

d. Work out the probability p that the component will lasts beyond 5 years for E[T]=3,4,5.

e. Discuss how and why p will change as the average lifetime is changed.

(2) Formulate and give explanation to the mathematical process needed to obtain the exponentially distributed random numbers in Problem (1) from random numbers that are uniformly distributed over [0,1]. Please report the following:

a. Formulate the inverse transformation method and explain the processes involved.

b. Develop a script that implements these processes, and then print out 100 exponentially distributed random numbers in a table.

c. Mark these random numbers on the x-axis of the pdf plot in the answer to Problem (1).

d. Elaborate the results that you obtained in 2.b and 2.c (for example the location of the random numbers).

Note that you must generate uniformly distributed random numbers first, and then 'transform' these into random numbers that have an exponential distribution.

References

[1] https://www.mathworks.com/help/matlab/ref/rand.html
[2] https://en.wikipedia.org/wiki/Inverse_transform_sampling