Instructions:

Try to solve all problems on your own. If you have difficulties, ask the instructor or TAs.

Please follow the instructions given below to prepare your solution notebooks:

- Please use different notebooks for solving different Exercise problems.
- The notebook name for Exercise 1 should be `ROLLNUMBER-labLL-ex1.ipynb`. `ROLLNUMBER-labLL-ex2.ipynb`, etc for others. 'LL' is the two digit lab number (lab-3 is 03, etc).
- Please ask your doubts to TAs or instructors or post in Moodle Discussion Forum channel.
- You should upload on the .ipynb files on Moodle (one per exercise).

Only the questions marked **[R]** need to be answered on paper. Write legible and to-the-point explanations. The work-sheet on which you write needs to be submitted before leaving the session.

Some other questions require plotting graphs (histograms, trajectories, level-sets etc) or tables. Please make sure that the plots are present in the submitted ipython notebooks.

Submission Time: Please check the submission deadline as show on the assignment web-page in Moodle. Late submissions will be accepted upto 24 hours from the deadline. All late submissions will have a penalty of 3 marks. Submissions later than 24 hours after the deadline will not be accepted.

Objective of this Lab will be to introduce simulation of stochastic processes. We continue from where we stopped from the previous lab.

Exercise [5 Marks] Simulate the random variable with density $f(x) = \frac{(x(1-x)e^x}{3-e}$ on $[0,1]$ using the method of rejection sampling as per the following steps:

1. Generate a random number x with the density $h(x)$.
2. Generate a uniformly distribute random number u from the interval $[0,1]$. If $u \leq \frac{f(x)}{ch(x)}$, then output x; otherwise reject x and return to Step 1. Here c is a constant such that $f(x) \leq ch(x)$ for all x.

Exercise [5 Marks, **R**]

Simulate Normal random variable using the method of rejection sampling.

Exercise [5 Marks, **R**] Generating a stochastic matrix and then simulate a markov chain using this transition matrix. Simulate the hitting times.

Exercise [10 Marks, **R**]

Suppose you want to generate a discrete-time Markov chain whose invariant distribution is a given vector $\bar{\pi}$. The Metropolis-Hastings algorithms takes any Markov chain and uses it to generate the desired Markov chain. We'll look at the algorithm in this lab, and go over the theory of why it works in lecture during a later class.

To illustrate the algorithm, let $S = \{0, 1, 2, 3\}$ S= 0,1,2,3. Let the desired invariant probability vector be $\bar{\pi} =$ [0.10.20.30.4]. Take P to be the transition matrix of any convenient Markov chain, for example, random walk on the circle with $p = \frac{1}{2}$. To generate a new Markov chain, we need to specify the transitions. Suppose $X_m = i$, the next state X_{m+1} is determined by a two-step procedure:

1. Choose a proposal state j using the transition probabilities P_{ij}. For the random walk on a circle, the proposal state j will be either $i - 1$ or $i + 1$ mod 4.
2. Decide whether or not to accept the proposal state, according to the acceptance function

$$a_{ij} = \frac{\pi_j P_{ji}}{\pi_i P + ij}.$$

   Generate a uniformly distributed random number u from the interval $[0,1]$. The next state of the chain is

$$X_{m+1} = \begin{cases} j \text{ if } u \leq a_{ij} \\ i\text{if } u > a_{ij} \end{cases}$$

Exercise [5 Marks]

Try with three more examples with large invariant distributions.