Problem 1. (35 points) Consider a queueing system with six parallel identical servers. Customers arrive to this queueing system according to a Poisson process and join a single queue. The service time for each customer is independent and identically distributed. Suppose that a single simulation run of this queueing system has been conducted to collect the following data: \( \{N(t), 0 \leq t \leq T\} \), where \( N(t) \) is the number of jobs in the system at time \( t \) and \( T \) is the total simulated time. This simulation was initialized at a state where all servers are busy but there are no other jobs waiting in the queue. Based on a prior warm-up period analysis, it is estimated that the stochastic process \( \{N(t), t \geq 0\} \), which is initialized as described, reaches steady state at around \( t = \tau \), where \( \tau \) is much smaller than \( T \).

(a) (12 points) Explain how you would apply the batch means method with \( b \) batches to construct a 95% confidence interval on the steady-state mean number of busy servers. Be specific about the upper and lower bounds of the confidence interval, i.e., express them as functions of \( \{N(t), 0 \leq t \leq T\}, T, \tau, \) etc.

(b) (6 points) Name one method to check the validity of the confidence interval obtained in part (a) using batch means.

(c) (7 points) If the confidence interval obtained in part (a) passes the test for validity in part (b) but is too large for the desired precision, what should be done to reduce the confidence interval length (to about one-third of its size) without compromising its validity?

(d) (10 points) The simulated process \( \{N(t), 0 \leq t \leq T\} \) enters state zero twenty times and \( N(T) = 7 \). Explain how the regenerative method can be used to provide a point estimator for the long-run average fraction of time the queue is empty. Be specific about the point estimator, i.e., express it in terms of \( \{N(t), 0 \leq t \leq T\}, T, \) etc.
Problem 2. (35 points) \( \{X(t), t \geq 0\} \) and \( \{Y(t), t \geq 0\} \) are two time-homogenous continuous-time stochastic processes defined on the same finite state space \( S \). For both processes, when the process leaves state \( i \in S \), it next enters state \( j \in S \) with some probability \( P_{ij} \), where \( P_{ii} = 0 \) and \( \sum_{j \in S} P_{ij} = 1 \). Each time \( \{X(t)\} \) enters state \( i \in S \), the amount of time it spends in that state before making a transition into a different state is exponentially distributed with mean \( \tau_i > 0 \). On the other hand, the sojourn time of \( \{Y(t)\} \) in state \( i \in S \) is uniformly distributed over \([\tau_i - \epsilon_i, \tau_i + \epsilon_i] \), where \( 0 < \epsilon_i < \tau_i \) for all \( i \in S \). For both processes, all sojourn times and transitions are independent. (Note that \( \{X(t), t \geq 0\} \) is a continuous-time Markov chain and \( \{Y(t), t \geq 0\} \) is a semi-Markov process.) In this question, we will design a simulation experiment to compare the fraction of time each process spends in state \( k \in S \) during \([0, T]\) if both processes just entered state \( k \) at time zero.

(a) (8 points) Provide an algorithm to simulate \( \{X(t), 0 \leq t \leq T\} \).

(b) (8 points) Provide an algorithm to simulate \( \{Y(t), 0 \leq t \leq T\} \).

(c) (7 points) Explain how you would synchronize random numbers used in the algorithms provided in parts (a) and (b) if we would like to use common random numbers.

(d) (12 points) Explain how you would construct a 95% confidence interval on the difference between the fraction of time these two processes spend in state \( k \) during \([0, T]\) using common random numbers. Define necessary notation to be specific about the upper and lower bounds of the confidence interval.

Problem 3. (30 points) Consider a random variable \( X \) that has a geometric distribution with parameter \( p \in (0, 1) \), i.e., \( \Pr\{X = i\} = p(1 - p)^i \) for \( i = 0, 1, \ldots \).

(a) (11 points) Prove that the following is an exact algorithm that generates geometric random variates:

Step 1. Generate a random number \( U \).

Step 2. Return \( X = \lfloor \ln(U) / \ln(1 - p) \rfloor \), where \( \lfloor x \rfloor \) is the floor function that returns the largest integer smaller than or equal to \( x \).

(b) (11 points) Provide another exact algorithm for generating a geometric random variate.

(c) (8 points) Compare the algorithms provided in parts (a) and (b) in terms of their computational efficiency.