

STK3405/4405

Mandatory assignment 1 of 1

Submission deadline

Thursday 23rd October 2025, 14:30 in Canvas (canvas.uio.no).

Instructions

You can choose between scanning handwritten notes or typing the solution directly on a computer (for instance with L^AT_EX). The assignment must be submitted as a single PDF file. Scanned pages must be clearly legible. The submission must contain your name, course and assignment number.

It is expected that you give a clear presentation with all necessary explanations. Remember to include all relevant plots and figures. Students who fail the assignment, but have made a genuine effort at solving the exercises, are given a second attempt at revising their answers. All aids, including collaboration, are allowed, but the submission must be written by you and reflect your understanding of the subject. If we doubt that you have understood the content you have handed in, we may request that you give an oral account.

In exercises where you are asked to write a computer program, you need to hand in the code along with the rest of the assignment. It is important that the submitted program contains a trial run, so that it is easy to see the result of the code.

Application for postponed delivery

If you need to apply for a postponement of the submission deadline due to illness or other reasons, you have to contact the lecturer (e-mail: arne@math.uio.no) well before the deadline.

The mandatory assignment in this course must be approved before you are allowed to take the final examination.

Complete guidelines about delivery of mandatory assignments:

uio.no/english/studies/admin/compulsory-activities/mn-math-mandatory.html

GOOD LUCK!

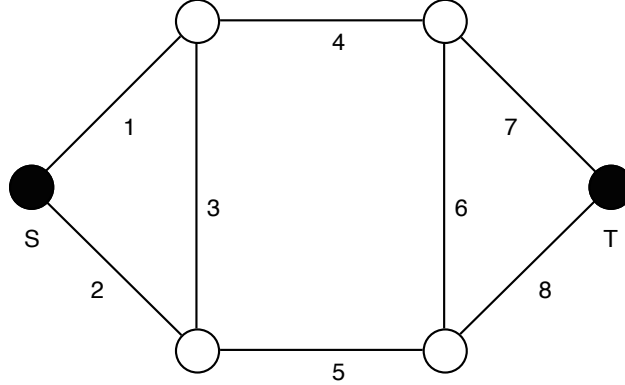


Figure 1: A 2-terminal undirected network system with 8 components

Problem 1. In this problem we consider the 2-terminal undirected network system (C, ϕ) shown in Figur 1. A pdf-file containing this figure can be found [here](#).

The nodes of the system are assumed to be functioning *perfectly* with probability 1, and the terminals of the system are the nodes labeled S and T . The components in the system are the *edges* in the network, labeled $1, \dots, 8$. Thus, the component set C is the set $\{1, \dots, 8\}$. We introduce the component state variables $X_1, \dots, X_8$. We assume that these variables are stochastically independent, and that the component reliabilities are given by:

$$P(X_i = 1) = p_i, \quad i = 1, \dots, 8.$$

We also introduce the component state vector $\mathbf{X} = (X_1, \dots, X_8)$ and the vector of component reliabilities $\mathbf{p} = (p_1, \dots, p_8)$.

In this assignment you will analyze this system both analytically and by using Monte Carlo simulation. For the simulations you will use scripts written in Python. The scripts needed for the assignment will be based on scripts you can download using links provided below. Thus, you do not need to write the Python scripts from scratch. Instead you simply adapt the scripts to the problems considered in the assignment.

IMPORTANT: You may include the pdf-file containing the network shown in Figur 1 as part of your assignment. When solving the points

below, you will create various plots. These plots, as well as the adapted Python scripts you have used, should be *included in the assignment report*.

- a) Find all the minimal path and cut sets of the system (C, ϕ) , and determine:

d = The length of the shortest path

c = The length of the shortest cut

- b) Show that the structure function of the system (C, ϕ) can be written as:

$$\begin{aligned}\phi(\mathbf{X}) = & X_3 X_6 \cdot (X_1 \amalg X_2) \cdot (X_4 \amalg X_5) \cdot (X_7 \amalg X_8) \\ & + X_3(1 - X_6) \cdot (X_1 \amalg X_2) \cdot ((X_4 X_7) \amalg (X_5 X_8)) \\ & + (1 - X_3) X_6 \cdot ((X_1 X_4) \amalg (X_2 X_5)) \cdot (X_7 \amalg X_8) \\ & + (1 - X_3)(1 - X_6) \cdot ((X_1 X_4 X_7) \amalg (X_2 X_5 X_8))\end{aligned}$$

- c) Use the result from (b) to find a similar expression for:

$$h(\mathbf{p}) = P(\phi(\mathbf{X}) = 1) = E[\phi(\mathbf{X})]$$

We now assume more specifically that the vector of component reliabilities is:

$$\mathbf{p} = (0.6, 0.3, 0.5, 0.4, 0.7, 0.5, 0.3, 0.6)$$

and start out by considering how the system reliability can be estimated using *crude Monte Carlo simulations*. Thus, we sample N independent binary vectors $\mathbf{X}_1, \dots, \mathbf{X}_N$, all having the same joint distribution as the component state vector $\mathbf{X}$. For each sample, the resulting value of the structure function is calculated. The system reliability $h = E[\phi]$ is then estimated by:

$$\hat{h}_{MC} = \frac{1}{N} \sum_{r=1}^N \phi(\mathbf{X}_r)$$

In order to monitor the convergence of the simulations, we introduce:

$$\hat{h}_{MC}^{(m)} = \frac{1}{m} \sum_{r=1}^m \phi(\mathbf{X}_r), \quad m = 1, \dots, N.$$

Thus, $\hat{h}_{MC}^{(m)}$ is the crude Monte Carlo estimate based on the m first simulations. By plotting $\hat{h}_{MC}^{(m)}$ as a function of m , we get a *convergence curve* for the crude Monte Carlo estimate.

- d) Consider the Python script found [here](#) which can be used to estimate the reliability of a bridge system using crude Monte Carlo simulations. Modify this script based on the results from (b) and (c) so that it can be used to estimate the reliability of the 2-terminal undirected network system (C, ϕ) shown in Figure 1.

The modified script should calculate the *true reliability* of the system using the expression for the reliability function $h(\mathbf{p})$ derived in (c), and also *estimate this reliability* based on the simulation results. Finally, the script should create a plot of the *convergence curve* described above and save this in a file in a folder called *crudegen*.

IMPORTANT: Before you run the script you must create a folder called *crudegen* in the directory where you store the script.

At the top of the script the number of simulations, N is entered into the variable `num_sims`. Run the script with `num_sims = 1000`, and include the results as well as the plot of the convergence curve in your assignment report.

We then proceed by using *conditional Monte Carlo simulation*, and introduce:

$$S = S(\mathbf{X}) = \sum_{i=1}^8 X_i$$

The set of possible values for S is $\{0, 1, \dots, 8\}$, and we let:

$$\theta_s = E[\phi | S = s], \quad s = 0, 1, \dots, 8.$$

The conditional expectations $\theta_0, \theta_1, \dots, \theta_8$ will be estimated by Monte Carlo sampling from the conditional distributions of $\mathbf{X}$ given $S = s$ for $s = 0, 1, \dots, 8$. The resulting estimates are denoted $\hat{\theta}_0, \hat{\theta}_1, \dots, \hat{\theta}_8$ respectively. Based on these estimates the system reliability will be estimated by:

$$\hat{h}_{CMC} = \sum_{s=0}^8 \hat{\theta}_s \cdot P(S = s)$$

- e) Assuming that $\hat{\theta}_0, \hat{\theta}_1, \dots, \hat{\theta}_8$ are *unbiased* estimates, i.e., $E[\hat{\theta}_s] = \theta_s$, $s = 0, 1, \dots, 8$, show that:

$$E[\hat{h}_{CMC}] = h$$

- f) Explain why $\theta_0 = \theta_1 = \theta_2 = 0$, while $\theta_7 = \theta_8 = 1$. How can this be used to make the conditional simulations more efficient?

- g) Consider the Python script found [here](#) which can be used to estimate the reliability of a bridge system using conditional Monte Carlo simulations. Modify this script based on the results from (a), (b) and (c) so that it can be used to estimate the reliability of the 2-terminal undirected network system (C, ϕ) shown in Figur 1.

The modified script should calculate the distribution of the variable S defined above, and estimate the conditional expectations $\theta_3, \dots, \theta_6$. Moreover, the script should calculate the *true reliability* of the system using the expression for the reliability function $h(\mathbf{p})$ derived in (c), and also *estimate this reliability* based on the simulation results. Finally, the script should create a plot of the *convergence curve* described above and save this in a file in a folder called *cmcgen*.

IMPORTANT: Before you run the script you must create a folder called *cmcgen* in the directory where you store the script.

At the top of the script the number of simulations, N is entered into the variable `num_sims`. Run the script with `num_sims = 1000`, and include all the results as well as the plot of the convergence curve in your assignment report.

Compare the results to the results from (d).

In the remaining problems we assume that all components have the same reliability:

$$p_1 = \dots = p_8 = p$$

The objective is to estimate the system reliability h as a function of this common component reliability, i.e., $h(p)$. In order to verify the simulation results, we will calculate $h(p)$ analytically as well.

- h) By using the result from (c) show that:

$$\begin{aligned} h(p) = & p^2 \cdot (p \amalg p) \cdot (p \amalg p) \cdot (p \amalg p) \\ & + 2p(1-p) \cdot (p \amalg p) \cdot (p^2 \amalg p^2) \\ & + (1-p)^2 \cdot (p^3 \amalg p^3) \end{aligned}$$

- i) Consider the Python script found [here](#) which can be used to estimate the reliability function, $h(p)$, of a bridge system using crude Monte Carlo simulations. Modify this script based on the results from (b) and (h) so that it can be used to estimate the reliability function of the 2-terminal undirected network system (C, ϕ) shown in Figur 1.

The modified script should create a plot containing both the *true reliability function*, as well as *the estimated reliability function* based on the simulation results, and save this in a file in a folder called *crude*.

IMPORTANT: Before you run the script you must create a folder called *crude* in the directory where you store the script.

At the top of the script the number of simulations, N is entered into the variable `num_sims`. Run the script with `num_sims = 1000`, and include the plot of the reliability functions in your assignment report.

- j) Consider the Python script found [here](#) which can be used to estimate the reliability function, $h(p)$, of a bridge system using conditional Monte Carlo simulations. Modify this script based on the results from (b) and (c) so that it can be used to estimate the reliability function of the 2-terminal undirected network system (C, ϕ) shown in [Figur 1](#).

The modified script should create a plot containing both the *true reliability function*, as well as *the estimated reliability function* based on the simulation results, and save this in a file in a folder called *cmc*.

IMPORTANT: Before you run the script you must create a folder called *cmc* in the directory where you store the script.

At the top of the script the number of simulations, N is entered into the variable `num_sims`. Run the script with `num_sims = 1000`, and include the plot of the reliability functions in your assignment report.

Compare the results to the results from (i).