

# Preparatory problems

## PhD course on Sequential Monte Carlo methods 2019

Department of Information Technology, Uppsala University

August 8, 2019

These problems are meant as a preparation to the homework, in particular if you are not very familiar with Monte Carlo methods. You do *not* have to hand in your solutions to these problems.

### P.1 Random number generation basics.

In this problem, you will familiarize yourself with random number generation in the language you have chosen to use.

- (a) Sample 1 000 random numbers from  $\mathcal{N}(0, 2)$  using some built-in function, and plot a histogram of it. Overlay the histogram with the density function for  $\mathcal{N}(x | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$  with  $\mu = 0, \sigma^2 = 2$ .

*Some useful commands:*

- Matlab: `randn()` or `mvnrnd()`
- Julia: `randn()` or `Normal()` in the Distributions package
- R: `rnorm()`
- Python: `numpy.random.normal()` or `scipy.stats.norm.rvs()`

**Hint 1:** Be careful whether the commands are parameterized in variance ( $\sigma^2$ ) or standard deviations ( $\sigma$ )!

**Hint 2:** Normalize the histogram, in order to make the overlaid true density plot easier to compare against.

- (b) Sample 1 000 random numbers from  $\mathcal{N}(0, 1)$  by first drawing uniforms on the interval  $[0, 1]$  and then transform them using the inverse cumulative distribution function for the Gaussian distribution. Plot a histogram of your samples, and overlay it with the density function for  $\mathcal{N}(0, 1)$  to confirm that your results are reasonable.

*Some useful commands:*

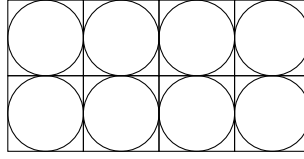
- Matlab: `norminv()`
- Julia: `norminvcdf()` in the StatsFuns package
- R: `qnorm()`
- Python: `scipy.stats.norm.ppf()`

*This idea of transforming uniform  $[0, 1]$  random variables into samples from a certain distribution is called ‘inverse transform sampling’ and is commonly used. We do, however, recommend you to let the language designers take care of these transforms, and encourage you to instead use built-in high-level functions (such as `randn()` etc.) when available, in order to save your time.*

- (c) Sample 1 000 numbers from  $\mathcal{N}(2, 10)$  by first drawing from  $\mathcal{N}(0, 1)$  and then make the appropriate linear transformation. Estimate the variance of your samples to confirm that you have done the right transformation.
- (d) Find out how to set your random seed. (When debugging Monte Carlo algorithms, this tends to be useful.)

## P.2 Compute $\pi$ using Monte Carlo

Consider a huge<sup>1</sup> wall with a wallpaper pattern with circles inside squares next to each other, as sketched below. Now, imagine that you, uniformly at random, throw a dart at the wall.



- (a) What is the probability that you hit inside a circle?  
 (b) Based on the calculation in (a) suggest a Monte Carlo method to estimate  $\pi$ . Implement your suggested method and provide a plot of your estimate of  $\pi$  as a function of the number of Monte Carlo samples  $N$ .

**Hint:** Since the pattern repeats itself infinitely, you could consider throwing dart uniformly on just a single square (instead of an infinite number of identical squares next to each other).

## P.3 Analytical manipulation of Gaussian densities

Consider the joint probability density function for the two scalar random variables  $X_a$  and  $X_b$

$$p(x_a, x_b) = \mathcal{N}(x \mid \mu, \Sigma) \quad (1a)$$

where

$$x = \begin{bmatrix} x_a \\ x_b \end{bmatrix}, \quad \mu = \begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \sigma_{aa} & \sigma_{ab} \\ \sigma_{ab} & \sigma_{bb} \end{bmatrix} \quad (1b)$$

It can be shown that the marginal distribution for  $x_a$  is.

$$p(x_a) = \mathcal{N}(x_a \mid \mu_a, \sigma_{aa}). \quad (2)$$

Show that the conditional distribution for  $X_a \mid X_b = x_b$  is

$$p(x_a \mid x_b) = \mathcal{N}(x_a \mid \mu_{a|b}, \sigma_{a|b}) \quad (3a)$$

where

$$\mu_{a|b} = \mu_a + \frac{\sigma_{ab}}{\sigma_{bb}}(x_b - \mu_b), \quad \sigma_{a|b} = \sigma_{aa} - \frac{\sigma_{ab}^2}{\sigma_{bb}}, \quad (3b)$$

by using the definition of conditioning

$$p(x_a \mid x_b) = \frac{p(x_a, x_b)}{p(x_b)} \quad (4)$$

and the definition of the multivariate Gaussian density

$$\mathcal{N}(x \mid \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d \det \Sigma}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right) \quad (5)$$

where  $d$  is the dimension of the random vector  $x$ .

**Hint 1:** A probability density always integrates to 1. Constant factors can therefore be neglected in your calculations.

**Hint 2:** For densities belonging to the exponential family, such as the Gaussian, it is often convenient to manipulate its logarithm.

---

<sup>1</sup>Let's pretend the wall is infinite.