Problem 2, project 2, TMA4285 2015
The Kalman filter

In this problem you shall estimate the unknown position of an object moving in two dimensional space from noisy observation of the location given a probabilistic model for the unknown movement path. The observations (a $n \times 2$ matrix `z`, $n = 100$) can loaded into R with the command

```
load(url("http://www.math.ntnu.no/~jarlet/tidsrekker/gps.RData"))
```

We will assume that the velocity vector remains constant within every interval between discrete time points $t = 1, 2, \ldots, n$ at which observations are made such that each component of the position vector changes according to

$$\begin{aligned} x_{t+1,1} &= x_{t,1} + v_{t,1} \\ x_{t+1,2} &= x_{t,2} + v_{t,2} \end{aligned} \tag{1}$$

In addition, each component of the velocity vector follows an AR(1) model with a common autoregressive coefficient ϕ such that

$$\begin{aligned} v_{t+1,1} &= \phi v_{t,1} + a_{t+1,1}, \\ v_{t+1,2} &= \phi v_{t,2} + a_{t+1,2}, \end{aligned} \tag{2}$$

where $a_t = (a_{t,1}, a_{t,2})^T$ is bivariate $N(0, \sigma_a^2 I_2)$ white noise. According to this model, the velocity thus changes instantaneously at each time point $t = 1, 2, \ldots, 100$.

Given the position $x_{t,1}, x_{t,2}$ the observed position is given by

$$\begin{aligned} z_{t,1} &= x_{t,1} + b_{t,1} \\ z_{t,2} &= x_{t,2} + b_{t,2} \end{aligned} \tag{3}$$

where $b_t = (b_{t,1}, b_{t,2})^T$ is bivariate $N(0, \sigma_b^2 I_2)$ white noise independent of $\{a_t\}$.

We will assume that the parameter values are known to be $\phi = 0.9$, $\sigma_a^2 = 1$ and $\sigma_b^2 = 9$.

Explain how this represents a state space model of the form

$$\begin{aligned} Y_{t+1} &= AY_t + Ga_{t+1} \\ Z_t &= HY_t + b_t \end{aligned} \tag{4}$$

where $a_t \sim N(0, \Sigma)$ and $b_t \sim N(0, \Omega)$.

Write your own code implementing the Kalman filter in R for this state space model (see eq. (14) to (16) in the Lecture summary). Choose sensible initial values for $\hat{Y}_{1|0}$ and $V_{1|0}$ reflecting the behaviour of the process. If some component of the state vector is non-stationary, a vague prior distribution with sufficiently large variance may be specified. To store the various quantities you need to compute, two and three dimensional arrays in R will be useful (e.g. `array(NA,dim=c(100,4,4))`).

Visualize the estimated movement path in a suitable way together with the observed data. Around each estimated position, draw ellipses enclosing, say 50% of the posterior probability distribution of each position at each time point, that is, the distribution of $(x_{t,1}, x_{t,2})$ conditional on $Z_1, Z_2, \ldots, Z_{100}$. You may use the function enclosed below for this. Comment what you see.

The next objective is to estimate the length of the moment path

$$L(Y_1, Y_2, \ldots, Y_n) = \sum_{t=1}^{n-1} \sqrt{v_{t,1}^2 + v_{t,2}^2}, \tag{5}$$

given our model and the observed data.

First compute $L^* = L(\hat{Y}_{1|n}, \hat{Y}_{2|n}, \ldots, \hat{Y}_{n|n})$. However, since L is a function of the unknown states $Y_1, Y_2, \ldots, Y_n$, the length is a random variable with its own conditional posterior distribution given the data. To find the overall distribution, you need to simulate a sufficient number of realisations from $Y_1, Y_2, \ldots, Y_n | Z_1, Z_2, \ldots, Z_n$ (see the Lecture summary). You are allowed to use the function `mvrnorm` from the `MASS`-package but note that this is only needed when simulating each $Y_t | Y_{t+1}, Y_{t+2}, \ldots, Y_n$. [1] For illustration, add a few movement paths simulated from the conditional distribution to the plot. Finally, make a histogram of all simulated L values and compare this to L^*. Briefly discuss why there is a difference.

[1] The `rmvnorm` from the `mvtnorm`-package by default does not handle positive semi-definite covariance matrices well if some eigenvalues are slightly negative because of numerical roundoff error, see this posting on stackoverflow.com.

```
# Draw an ellipse enclosing a probability mass alpha of a binormally distributed
# variable in an exisiting plot
#
# Arguments
#   mu: The mean vector
#   sigma: The covariance matrix
#   alpha: The probability to enclose
#   ...: Further arguments to lines
mvnellipse <- function(mu,sigma,alpha=.5,...) {
  tmp <- eigen(sigma)
  theta <- seq(0,2*pi,len=40)
  r <- sqrt(qchisq(alpha,df=2))
  lines(t(mu + tmp$vec %*% diag(sqrt(tmp$val)) %*% (r*rbind(cos(theta),sin(theta)))),...)
}
```