TMA4285 Time series models
Project 1
Autumn 2016

Consider the ARMA(1,2) model

$$(1 - \phi_1 B)Z_t = (1 - \theta_1 B - \theta_2 B^2)a_t,$$

where $\{a_t\}_{t=-\infty}^{\infty}$ is Gaussian distributed white noise with zero mean and $\operatorname{Var} a_t = \sigma_a^2$. We will assume that $\phi_1 = -0.8$, $\theta_1 = 1.88$ and $\theta_2 = -0.98$, and $\sigma_a^2 = 1$.

1. For the above model, find analytical formulas for the variance $\operatorname{Var} Z_t$ and the autocorrelation function $\rho_k, k = 1, 2, \ldots$. Also write an R function that takes $\rho_1, \rho_2, \ldots, \rho_k$ as input and computes the partial autocorrelation function at lags up to k using the Durbin-Levinson algorithm.

2. Using the R-function `arima.sim`, simulate $\{z_t\}_{t=1}^{1000}$ from the above model. Note that arima.sim is based on a parameterization of the MA-part of the model with opposites signs from that of Wei. Plot a suitable closeup of the realisation.

   Use the simulated realisation to estimate the acf and the pacf of the model for lags up to $k = 20$ using R-functions `acf` and `pacf`. To study the bias and variance of the sample acf and pacf at different lags k, repeat the process of simulation and estimation several times. Visualise your results and use them to check your theoretical acf and pacf computed using result in point 1.

3. You should now make your own function which can simulate from the above model. Your function does not need to be general, it is sufficient that it can simulate from the above model with parameter values as specified above. You should only use the function rnorm to generate random numbers. Thus, you are not allowed to use the `arima.sim` function or any other random generating functions, except `rnorm`. In

addition to including the code in your solution, you should also explain your simulation strategy in normal text and include calculations that are necessary to understand your simulation strategy. In particular, carefully consider how to choose initial values. What is the joint marginal distribution of Z_1, a_1, a_0? Run repeated simulations to check that your implementation produces a process that is stationary in the variance, that is, $\operatorname{Var} Z_t$ should be the same for all $t = 1, 2, \ldots, 1000$.

Run further checks of your implementation by repeating the simulation and estimation parts in point 2 using your own function in instead of `arima.sim`.

4. Based on the duality between MA(p) and AR(q) processes, briefly discuss the *qualitative* behaviour of the theoretical pacf of the above model in terms of the roots of the moving average polynomial $\theta(B)$.

5. Using the above ARMA(1,2) model, suppose that we want to forecast Z_{21} based on $Z_1, Z_2, \ldots, Z_{20}$ contained in the R-vector `z` which you can load into R with the command

```
load(url("https://www.math.ntnu.no/~jarlet/tidsrekker/zforecast.Rdata"))
```

First compute the forecast via the AR(∞) representations of the model. Since $Z_0, Z_{-1}, \ldots$ etc. in reality are not available, you can only compute an approximate forecast based on the known values by dropping the terms containing $Z_0, Z_{-1}, \ldots$. Also find the forecast error variance using this approach. Make a plot of the forecast weights $\pi_1, \pi_2, \ldots, \pi_{20}$.

6. Also, write your own code computing and plotting the l-step ahead forecast and 95% forecast limits for $l = 1, 2, \ldots, 10$.

7. The optimal 1-step ahead forecast of Z_{21} given only $Z_1, Z_2, \ldots, Z_{20}$ can alternatively be computed by writing Z_{21} as a regression on $Z_1, Z_2, \ldots, Z_{20}$,

$$Z_{21} = \phi_{20,1} Z_{20} + \phi_{20,2} Z_{19} \cdots + \phi_{20,20} Z_1 + e_{20}. \tag{1}$$

Explain why this regression exist for a Gaussian process and how the forecast and the forecast error variance can be computed via this regression.

Modify the function from point 1 such that it instead optionally returns the coefficients in this regression aswell as $\mathrm{Var}(e_{20})$, again computed via the Durbin-Levinson algorithm, see eqs. 3a-c in the Lecture summary. Compute the actual forecast. Plot the forecast weights obtained using this method and compare them to the forecast weights computed via the $\mathrm{AR}(\infty)$ representation of the model. Also compare the to forecast error variances computed via the two methods.

8. Which forecast method is preferable in practice in terms of accuracy and computational cost, respectively? Why is there a considerable difference between the two methods in the present example? Explain how the magnitude of this difference relates to the roots of the MA-polynomial of the model. Would any problem arise if the two methods were used for a non-invertible, stationary ARMA-model?