Problem Set 7 – The Particle Filter

The goal of this problem set is to use the particle filter in combination with Metropolis-Hastings to estimate parameters from an observed time series.

The variable X_t evolves according to an ARMA$(2,2)$ process:

$$X_t = \rho_1 X_{t-1} + \rho_2 X_{t-2} + \phi_1 \varepsilon_{t-1} + \phi \varepsilon_{t-2} + \varepsilon_t.$$

Unfortunately we cannot observe X_t directly. Instead, every period we observe only data (A_t, B_t) which is related to X_t in the following way:

$$\begin{aligned} A_t &= \exp\left(X_t + \nu_t^A\right) \\ B_t &= \beta X_t^2 + \nu_t^B \end{aligned}$$

The repository contains the observed data series $\{(A_t, B_t)\}_{t=1}^T$ for a $T = 400$.

1. Write the problem in terms of notation used in class, that is define the transition function g, the observation equation h, the state S_t, observables Y_t and the shocks W_t and V_t to state variables and observables, respectively. Which parameters that we need to estimate does the parameter vector θ contain?

2. Use the particle filter to approximate the likelihood function $p\left(Y_t|\theta\right)$ for a given set of parameters θ.

3. Incorporate the routine from 2. into a Metropolis-Hastings algorithm to find posterior distributions for all parameters.

Hints/Suggestions:

- Experiment with the number of particles N, the length M of the MCMC sample and the chain's burn-in period.

- Feel free to experiment with priors. Here is a starting point: the ARMA coefficients are less than one in absolute value, and most of them are less than 0.5 in absolute value. The variance of the measurement error is positive, so you could use a log-normal prior. For β we don't really have any information, so perhaps start with a flat prior.

- I found it a bit difficult to get decent convergence rates. I ended up using normally distributed proposal steps with a sd of 0.05 for all parameters, but convergence rates were still too low.

- The main objective is to get the particle filter to work, so that it gives you an (approximated) likelihood function for the data for a given set of parameters – if the MCMC algorithm doesn't give you great posteriors for parameters don't worry about it too much.