Sampling from VAR using Conditional SMC

January 30, 2025

1 Introduction

1.1 Vector Autoregression (VAR) Model

Consider the following Vector Autoregression (VAR) model with N variables and lag p:

$$y_t = \Pi z_t + v_t, \tag{1}$$
$$v_t = A^{-1}\Lambda_t^{1/2}\epsilon_t, \quad \epsilon_t \sim N(0, I_N), \tag{2}$$
$$\log \lambda_t = \log \lambda_{t-1} + e_t, \quad e_t \sim N(0, \Phi). \tag{3}$$

where:

$z_t = \begin{bmatrix} 1 & y_{t-1}^T & \cdots & y_{t-p}^T \end{bmatrix}^T \in \mathbb{R}^{Np+1}$,
$\Pi \in \mathbb{R}^{N \times (Np+1)}$,
$\Lambda_t = \text{diag}(\lambda_t)$,
$A^{-1} \in \mathbb{R}^{N \times N}$, a lower triangular matrix with ones on the diagonal,
$\Phi \in \mathbb{R}^{N \times N}$, a diagonal matrix.

1.2 Motivation

With the assumption that Φ is diagonal, the variance of the volatility for each variable follows an independent random walk:

$$\log \lambda_t = \log \lambda_{t-1} + e_{j,t}, \quad e_{j,t} \sim N(0, \Phi_{j,j}).$$

This structure enables us to decompose the high-dimensional problem into a series of one-dimensional simulations, simplifying the computational framework. Given

these conditions, the particle filter becomes a natural and heuristic choice. Its ability to handle non-linear and non-Gaussian state-space models aligns well with the random walk dynamics of the volatilities, enabling efficient inference and simulation.

As we will demonstrate, compared to the method proposed in [2], which employs a Metropolis-Hastings algorithm to simulate each $\lambda_{t,j}$ sequentially for $t = 1, \ldots, T$ and $j = 1, \ldots, N$, using a particle filter effectively corresponds to a block update within a Gibbs sampler. Specifically, the particle filter updates $\lambda_{1:T,j}$ in a single step. By leveraging the properties of Gibbs sampling, this block update strategy significantly improves the mixing of the Markov chain, leading to more efficient inference.

Introduce Particle Gibbs The simulator introduced above is a Particle Gibbs sampler, which incorporates a Conditional Sequential Monte Carlo (CSMC) step within a Gibbs sampling framework. The number of particles used in the CSMC directly affects the mixing properties of the sampler. However, as we will demonstrate, even a small number of particles can result in significant improvements compared to the method proposed in [2].

1.3 Prior

In the following section, we construct a Gibbs sampler to sample from the posterior distribution of A, Π, and λ_t for $t = 1, \ldots, T$. We begin by specifying the priors for these parameters and the assumed distributions for the remaining parameters.

The following set up of priors are according to [1].

For each individual entry in the matrix A that is not fixed, we assign it an independent uninformative prior given by $N(0, 10^6)$.

We adopt the Minnesota prior for Π and represent it as $\Pi = [\Pi_0, \Pi_1, \ldots, \Pi_p]$. Except for Π_1, the prior means are set to 0. The prior mean of Π_1 is specified as a diagonal matrix, with diagonal elements set to either 1 or 0 depending on the degree of persistence observed in the data series $y_{1:T,j}$ (i.e., the j-th diagonal element depends on the correlation of $y_{1:T,j}$).

For Π_0, we set the prior variance to 100. Otherwise, the variance structure is defined as follows:

- $\text{Var}(\Pi_{(l)}^{i,i}) = \frac{\lambda_1}{l^{\lambda_3}}$ for $l = 1, \ldots, p$,
- $\text{Var}(\Pi_{(l)}^{i,j}) = \frac{\lambda_1 \lambda_2 \sigma_i^2}{l^{\lambda_3} \sigma_j^2}$ for $i \neq j$, $l = 1, \ldots, p$,

where:

- $\lambda_1, \lambda_2, \lambda_3$ are hyperparameters controlling the prior structure,

- σ_i^2 represents the variance of the residuals for series i from a univariate autoregressive model.

For a medium-sized model, we set the parameters as follows:

$$\lambda_1 = 0.05, \quad \lambda_2 = 0.5, \quad \lambda_3 = 2.$$

To set the scale parameters σ_i^2, we follow common practice and define it as the variance of the residuals from a univariate autoregressive model.

We set a non-informative prior $N(0, 100)$ for $\lambda_{0,j}$ for $j = 1, \ldots, N$. As in CSMC, we sample in the log space, we assign a prior $N(0, 2.31)$ for $\log \lambda_{0,j}$.

2 Conditional SMC

With the assumption that Φ is a diagonal matrix, we can run a Conditional SMC for each $j = 1, \ldots, N$ independently. Consider the posterior distribution of $\log \lambda_{t,j}$:

$$\begin{aligned} &\pi(\log \lambda_{t,j} | A, \Pi, y) \\ &\propto \exp\left\{ -\frac{(\log \lambda_{t,j} - \log \lambda_{t-1,j})^2}{2\Phi_{j,j}} \right\} \frac{1}{\lambda_{t,j}^{1/2}} \exp\left\{ -\frac{(A^{(j)T}(y_t - \Pi z_t))^2}{2\lambda_{t,j}} \right\}. \end{aligned}$$

Using the bootstrap filter, in each time step, we sample $\log \lambda_{t,j} \sim N(\log \lambda_{t-1,j}, \Phi_{j,j})$ and then weight the particle by $\frac{1}{\lambda_{t,j}^{1/2}} \exp\left\{ -\frac{(A^{(j)T}(y_t - \Pi z_t))^2}{2\lambda_{t,j}} \right\}$.

Remark 1. *There could be better way of sampling that will lead to smaller variance?*

2.1 Backward Sampling

To mitigate the effects of path degeneracy and allow particles to explore the state space effectively, backward sampling is often employed. However, in this specific example, under the given model and prior assumptions, the update rate remains notably high even without backward sampling.

Nevertheless, the following experiments are done with backward sampling.

2.2 Number of Particles

For a fixed T (here $T \approx 650$), the number of particles used in CSMC significantly influences the mixing of $\log \lambda_{1:T,1:N}$. In Figure 2a, we execute the CSMC algorithm 2000 times and calculate the percentage of time steps where the simulated particle

Algorithm 1 Conditional Sequential Monte Carlo Algorithm

procedure FORWARD FILTERING($N, T, Num, \Phi, A, \Pi, y$)
  for $j = 1$ to N **do**
    Initialize fixed particle $\log \lambda_{1:T,j}^{(0)} \sim \pi(\log \lambda_{1:T,j} \mid A, \Pi, y)$
    Initialize $\log \lambda_{0,j}^{1:Num}$ from the prior distribution.
    for $t = 1$ to T **do**
      if ESS $(W_{1:Num}^{t-1}) \leq 0.5 \times Num$ **then**
        Resample particles: $A_t^{1:Num} \sim$ Multinomial$(W_{1:Num}^{t-1})$.
        $W_{1:Num}^{t-1} = 1$
      else
        $A_t^{1:Num} = 1 : Num$
      end if
      for $n = 1$ to Num **do**
        if $n = 1$ **then**
          set $\log \lambda_{t,j}^{(1)} = \log \lambda_{t,j}^{(0)}$
          $W_n^t = W_n^{t-1} \times \frac{1}{\lambda_{t,j}^{01/2}} \exp\left\{-\frac{(A^{(j)T}(y_t - \Pi z_t))^2}{2\lambda_{t,j}^0}\right\}$
        else
          Sample $\log \lambda_{t,j}^{(n)} \sim N(\log \lambda_{t-1,j}^{A_t^n}, \Phi_{j,j})$.
          $W_n^t = W_n^{t-1} \times \frac{1}{\lambda_{t,j}^{n1/2}} \exp\left\{-\frac{(A^{(j)T}(y_t - \Pi z_t))^2}{2\lambda_{t,j}^n}\right\}$
        end if
      end for
    end for
    Sample $B_T \sim Multinomial(W_{1:Num}^T)$.
    Set $\log \lambda_{T,j}^* = \log \lambda_{T,j}^{B_T}$
    for $t = T - 1 to 1$ **do**
      $B_t = A_t^{B_{t+1}}$
      $\log \lambda_{t,j}^* = \log \lambda_{T,j}^{B_t}$
    end for
  end for
  return $\log \lambda_{1:T,j}^*$.
end procedure

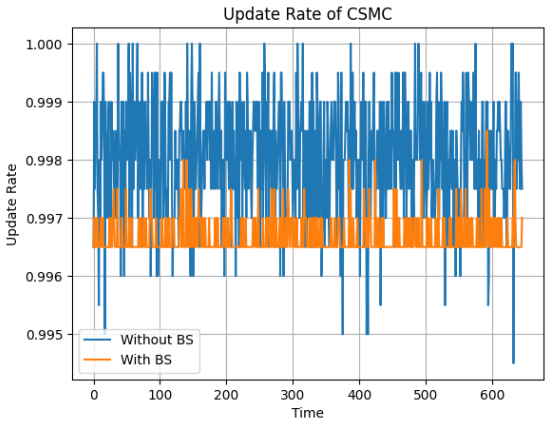


Figure 1: Compare update rate of CSMC with and without backward sampling.

Algorithm 2 Backward Sampling

procedure BACKWARD SAMPLING($\log \lambda_{1:T,j}^{1:Num}$)

Sample $B_T \sim \text{Multinomial}(W_{1:Num}^T)$,

where $W_n^T = \frac{1}{\lambda_{T,j}^{n1/2}} \exp\left\{-\frac{\left(A^{(j)T}(y_T - \Pi z_T)\right)^2}{2\lambda_{T,j}^n}\right\}$.

Set $\log \lambda_{T,j}^* = \log \lambda_{T,j}^{B_T}$.

for $t = T-1$ to 1 **do**

Compute $W_n^t = \exp\left\{-\frac{\left(A^{(j)T}(y_t - \Pi z_t)\right)^2}{2\lambda_{t,j}^n}\right\} \times N\left(\log \lambda_{t+1,j}^{B_{t+1}} \mid \log \lambda_{t,j}^n, \Phi_{j,j}\right)$.

Sample $B_t \sim \text{Multinomial}(W_{1:Num}^t)$.

Set $\log \lambda_{t,j}^* = \log \lambda_{t,j}^{B_t}$.

end for

return $\log \lambda_{1:T,j}^*$.

end procedure

value differs from the fixed particle value. For a number of particles greater than 10, the update rate at all time steps exceeds 95%.

We also run the entire Gibbs sampler for 2000 iterations with varying numbers of particles. The corresponding mixing behavior is shown in Figure 2b. Adding more particles can smooth the mixing, as seen when using 50 particles, but the overall speed does not vary significantly.

Remark 2. *Any conclusion that can be made?*

3 Gibbs Sampler

3.1 Updating A

For the Gibbs step for updating A, we use the method from [2]. Consider the posterior distribution of A:

$$\begin{aligned}\pi(A \mid y, \Pi, \lambda_{1:T,1:N}) &\propto N(\mu_A, \Sigma_A) \prod_{t=1}^{T} \exp\left\{-\frac{1}{2}(Ay_t - A\Pi z_t)^\top \Lambda_t^{-1}(Ay_t - A\Pi z_t)\right\} \\ &\propto \prod_{j=1}^{N} N(\mu_{A,j}, \Sigma_{A,j}) \prod_{t=1}^{T} \exp\left\{-\frac{1}{2}\frac{(A^{(j)\top}(y_t - \Pi z_t))^2}{\lambda_{t,j}}\right\}.\end{aligned}$$

Let $\hat{y}_t = y_t - \Pi z_t$. Notice that:

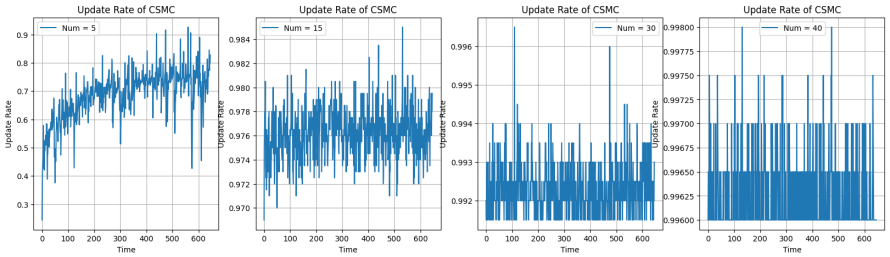
$$A^{(j)}\hat{y}_t = A_{j,1}\hat{y}_{t,1} + A_{j,2}\hat{y}_{t,2} + \cdots + A_{j,j-1}\hat{y}_{t,j-1} + \hat{y}_{t,j},$$

where A is a lower triangular matrix with diagonal entries equal to one. Based on the prior structure imposed on A, we can perform independent updates on each row of A:

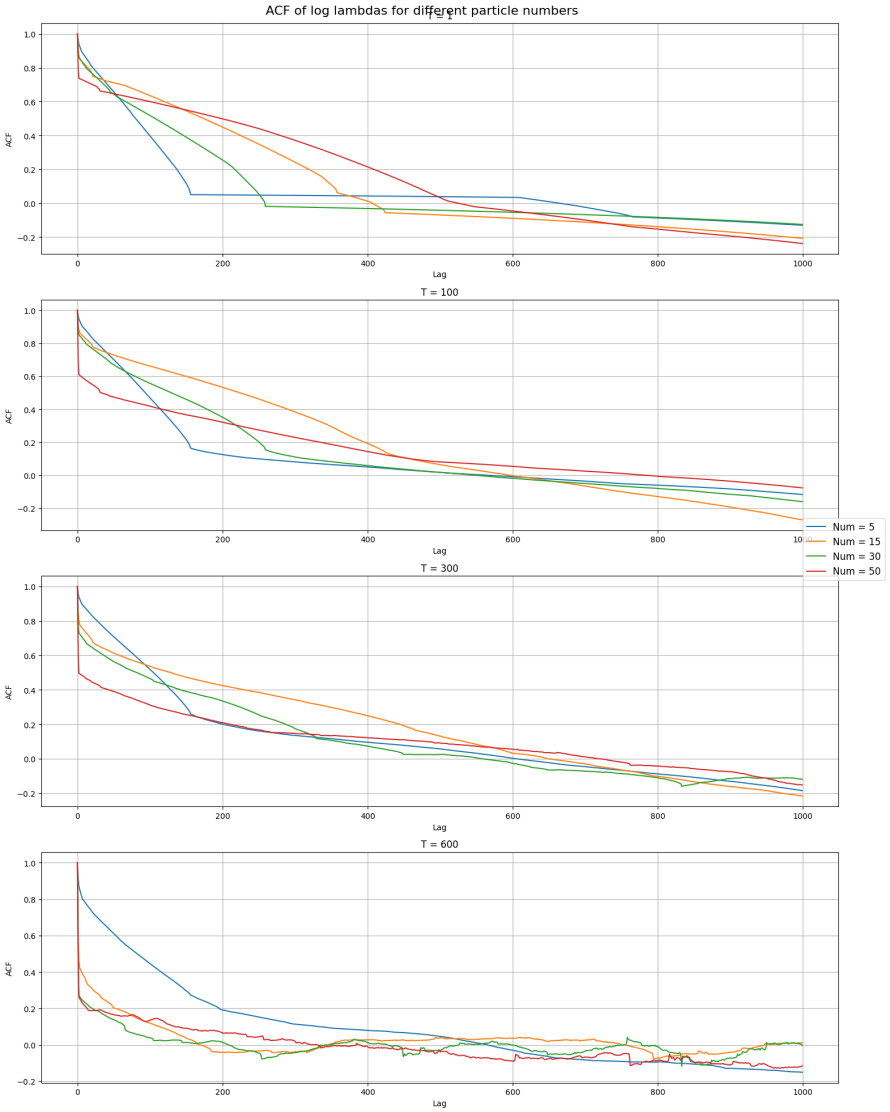
$$A \mid y, \Pi, \lambda_{1:T,1:N} \sim N(\bar{\mu}_{A,j}, \bar{\Sigma}_{A,j}),$$

where:

$$\begin{aligned}\bar{\Sigma}_{A,j} &= \left(\Sigma_{A,j} + \sum_{t=1}^{T} \frac{\hat{y}_{t,1:j-1}\hat{y}_{t,1:j-1}^\top}{\lambda_{t,j}}\right) \in \mathbb{R}^{(j-1)\times(j-1)}, \\ \bar{\mu}_{A,j} &= \bar{\Sigma}_{A,j}\left(\Sigma_{A,j}^{-1}\mu_{A,j} + \sum_{t=1}^{T} \frac{\hat{y}_{t,1:j-1}}{\lambda_{t,j}}\right) \in \mathbb{R}^{j-1}.\end{aligned}$$



(a) Update rate of each time step of CSMC with backward sampling.



(b) ACF of selected time step of CSMC with backward sampling.

Figure 2: Analysis of CSMC performance with backward sampling: (a) Update rate across time steps and (b) ACF for selected time steps.

3.2 Updating Π

For the update of Π, we use the exact same methods as introduced in [1].

3.3 Updating Φ

In [1], they impose an inverse Wishart prior on Φ, with scale I_N and degree of freedom N+2. Since we are considering diagonal Φ, this breaks down to imposing independent inverse gamma prior each diagonal element of Φ i.e.

$$\Phi_{j,j} \sim \text{InvGamma}((N+2)/2, 1/2) \forall j = 1, \ldots, N.$$

The posterior of Φ is then given by:

$$\begin{aligned}\pi(\Phi_{j,j}|y_{1:T}, A, \Pi, \lambda_{1:T,j}) &\propto \pi(\Phi_{j,j}) \prod_{t=1}^{T} exp\left\{-\frac{(\log \lambda_{t,j} - \log \lambda_{t-1,j})^2}{2\Phi_{j,j}}\right\}, \\ &\sim \text{InvGamma}(\frac{N+2+T}{2}, \frac{1+\sum_{t=1}^{T}(\log \lambda_{t,j} - \log \lambda_{t-1,j})^2}{2})\end{aligned}$$

3.4 Updating $\lambda_{1:T,1:N}$

In this subsection, we introduce the method used by [1] to sample $\lambda_{1:T,1:N}$. When Φ is diagonal, [1] adopts the method from [2]. Considering Markovianity, they focus on the distribution:

$$\begin{aligned}\pi(\lambda_{t,j}|\lambda_{-t,j}, A, \Pi, y) \propto &\frac{1}{\lambda_{t,j}^{1/2}} \exp\left\{-\frac{(A^{(j)T}(y_t - \Pi z_t))^2}{2\lambda_{t,j}}\right\} \\ &\times \exp\left\{-\frac{1}{2}\frac{(\log \lambda_{j,t} - \log \lambda_{j,t-1})^2}{\Phi_{j,j}} - \frac{1}{2}\frac{(\log \lambda_{j,t} - \log \lambda_{j,t+1})^2}{\Phi_{j,j}}\right\}\frac{1}{\lambda_{t,j}} \\ \propto &\left(\frac{1}{\lambda_{t,j}}\right)^{1.5} \exp\left\{-\frac{(A^{(j)T}(y_t - \Pi z_t))^2}{2\lambda_{t,j}}\right\} \times \exp\left\{-\frac{(\log \lambda_{j,t} - m_{j,t})^2}{2\sigma_{j,t}^2}\right\},\end{aligned}$$

where:

$$\begin{aligned}m_{j,t} &= \frac{1}{2}(\log \lambda_{t-1,j} + \log \lambda_{t+1,j}), \\ \sigma_{j,t}^2 &= \frac{1}{2}\Phi_{j,j}.\end{aligned}$$

They therefore consider a Metropolis-Hasting sampler with target $\pi(\lambda_{t,j}|\lambda_{-t,j}, A, \Pi, y)$ and proposal

$$q(\lambda_{t,j}) \propto \frac{1}{\lambda_{t,j}^{1/2}} \exp\left\{-\frac{(A^{(j)T}(y_t - \Pi z_t))^2}{2\lambda_{t,j}}\right\}.$$

We sample from q in the log space.

3.5 Updating $\log \lambda_t$ using Kalman Filter.

Consider the state space model for volatility, let $x_{t,j} = \log \lambda_{t,j}$ for convenience. For $j = 1, \ldots, N$,

$$\begin{aligned} log((Av_t)^{(j)2}) &= x_{t,j} + \log \epsilon_{t,j}^2 & \epsilon_{t,j} &\sim \text{iid} N(0,1) \\ x_{j,t} &= x_{j,t-1} + e_{j,t} & e_t &\sim N(0, \Phi). \end{aligned}$$

Let $v_{t,j}^* = log((Av_t)^{(j)2} + c)$, we approximate the state space model by

$$v_{t,j}^* = x_{t,j} + z_{t,j}$$

where $z_{t,j}$ simulated from a mixture of Gaussian, i.e.

$$f(z_{j,t}) = \sum_{k=1}^{K} q_k f_N(z_{j,t} | m_k - 1.2704, \sigma_k^2) \text{iid}.$$

f_N stands for the density function of normal distribution, and the constants m_i, σ_i, q_i are available in [4]. They also chose K = 7, c = 0.001.

Let $s_{t,j}$ denotes the group of Gaussian that $z_{t,j}$ belongs to, i.e.

$$\begin{aligned} &\mathbb{P}(s_{t,j} = k) = q_k \\ &z_{t,j} | s_{t,j} = k \sim N(m_k - 1.2704, \sigma_k^2). \end{aligned}$$

Suppose $v_{t,j}^*$ is observed (which can be deduced from updates of other blocks in the Gibbs sampler), we simulate $x_{1:T,1:N}$ using technique in [5].

First, we use Kalman Filter to calculate $x_{t|t} = \mathbb{E}[x_{t,j} | v_{1:t}^*]$ and $P_{t|t} = \text{Var}[x_t | v_{1:t}^*]$.

Initialize $x_{1|0}$ and $P1|0$, we recursively update:

prediction step

$$\begin{aligned} &x_{t|t-1} = x_{t-1|t-1} \\ &P_{t|t-1} = P_{t-1|t-1} + \Phi \end{aligned}$$

updating step

$$\begin{aligned} &K_t = P_{t|t-1}(P_{t|t-1} + \text{diag}(\sigma_{s_t}))^{-1} \\ &\mu_t = v_t^* - (m_{s_t} - 1.2704) - x_{t|t-1} \\ &x_{t|t} = x_{t|t-1} + K_t \mu_t \\ &P_{t|t} = (I_N - K_t) Pt|t-1. \end{aligned}$$

To simulate the trajectory $x_{1:T} | v_{1:T}^*$, we also need $\mathbb{E}[x_t | v_{1:t}^*, x_{t+1}]$ and $\text{Var}[x_t | v_{1:t}^*, x_{t+1}]$.

Suppose now Φ is in general nondiagonal, consider its Cholesky decomposition

$$\Phi = L \Delta L^T$$

, where L is a lower triangular matrix with diagonal ones and Δ is a diagonal matrix. Rewrite

$$\tilde{x}_t = L^{-1}x_t$$
$$\tilde{x}_{t,j} = (L^{-1})^{(j)}x_{t-1} + u_{t,j} \qquad u_t \sim N(0, \Delta_j).$$

For $j = 1, \ldots, N$, define

$$x_{t|t,j} = \mathbb{E}[x_t | v^*_{1:t}, x_{t+1,1:j}] \qquad P_{t|t,j} = \mathrm{Var}[x_t | v^*_{1:t}, x_{t+1,1:j}],$$

and initialize $x_{t|t,0} = x_{t|t}$, $P_{t|t,0} = P_{t|t}$. We update recursively:

$$\epsilon_{t,j} = \tilde{x}_{t+1,j} - (L^{-1})^{(j)T} x_{t|t,j-1}$$
$$R_{t,j} = (L^{-1})^{(j)T} P_{t|t,j-1} (L^{-1})^{(j)} + \Delta_j$$
$$x_{t|t,j} = x_{t|t,j-1} + P_{t|t,j-1}(L^{-1})^{(j)} \epsilon_{t,j}/R_{t,j}$$
$$P_{t|t,j} = P_{t|t,j-1} - P_{t|t,j-1}(L^{-1})^{(j)}(L^{-1})^{(j)T} P_{t|t,j-1}/R_{t,j}.$$

We then obtain $x_{t|t,N} = \mathbb{E}[x_t | v^*_{1:t}, x_{t+1}]$, $P_{t|t,N} = \mathrm{Var}[x_t | v^*_{1:t}, x_{t+1}]$.

By

$$p(x_{1:T} | v^*_{1:T}) = p(x_T | v^*_{1:T}) \prod_{t=1}^{T-1} p(x_t | v^*_{1:t}, x_{t+1})$$

we can generate the trajectory.

Given $x_{1:T,1:N}$, to simulate $s_{1:T,1:N}$ we only need to consider independently

$$p(s_{t,j} = k | x_{t,j}, v^*_{t,j}) \propto q_k f_N(v^*_{t,j} | x_{t,j} + m_k - 1.2704, \sigma_k^2).$$

4 Experiment on FRED data

In this section, we use FRED data from [3] to compare the Gibbs sampler proposed in [1] with the Gibbs sampler incorporating CSMC.

We consider a medium-sized model with $N = 18$ and $p = 13$. Figure **??** shows that the mixing of A and Π generated by both samplers is comparable, with no significant differences observed. However, the mixing of $\log \lambda$ is noticeably slower when using the method in [1].

It is worth noting that we adjusted the hyperparameters in both samplers to ensure comparable runtime. Specifically, we choose the number of particles to be 30, number of MH step to be 15.

Also consider the comparison of CPU time with the two methods.

Remark 3. *I want to compare the CPU time when the two samplers have comparable mixing. For this, I run a small model with N = 2 and p =3. But I don't think it is a good way to compare mixing because mixing might be affected by the data series. But is there other way to compare?*

References

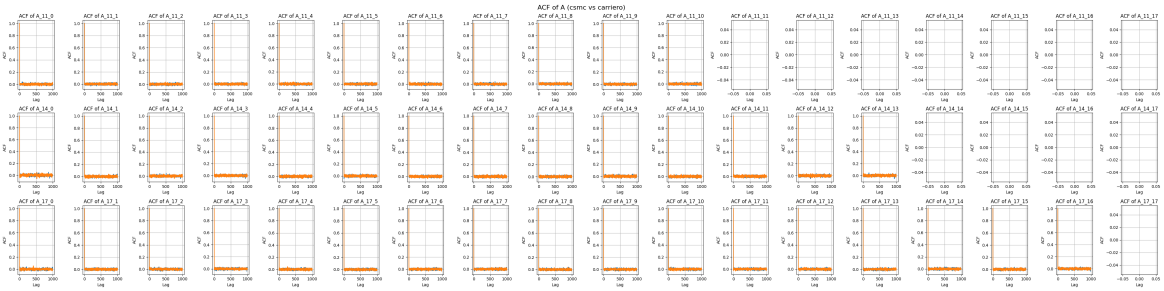
[1] Carriero, A., Clark, T. E., & Marcellino, M. (2019). Large Bayesian vector autoregressions with stochastic volatility and non-conjugate priors. *Journal of Econometrics, 212*(1), 137–154. https://doi.org/10.1016/j.jeconom.2019.04.024

[2] Cogley, T., & Sargent, T. J. (2005). Drifts and volatilities: monetary policies and outcomes in the post WWII US. Review of Economic Dynamics, 8(2), 262–302. https://doi.org/10.1016/j.red.2004.10.009

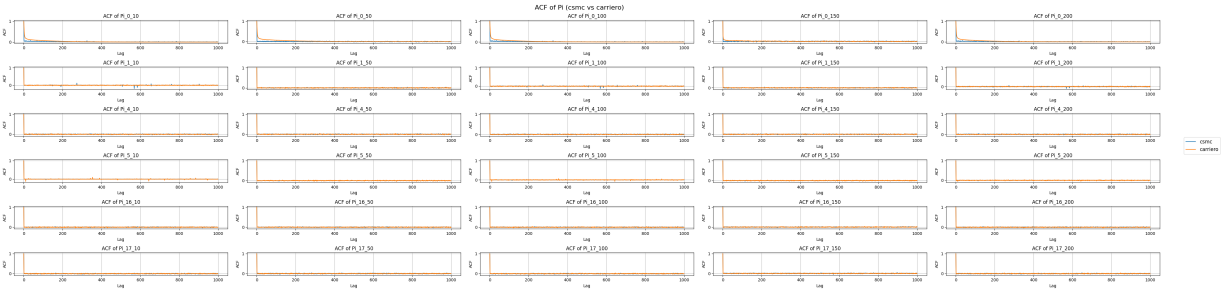
[3] McCracken, M. W., Ng, S. (2015). FRED-MD: a monthly database for macroeconomic research. Journal of Business and Economic Statistics, 34(4), 574–589. https://doi.org/10.1080/07350015.2015.1086655

[4] Kim, S., Shepherd, N., Chib, S. (1998). Stochastic Volatility: Likelihood Inference and Comparison with ARCH Models. The Review of Economic Studies, 65(3), 361–393. https://doi.org/10.1111/1467-937x.00050

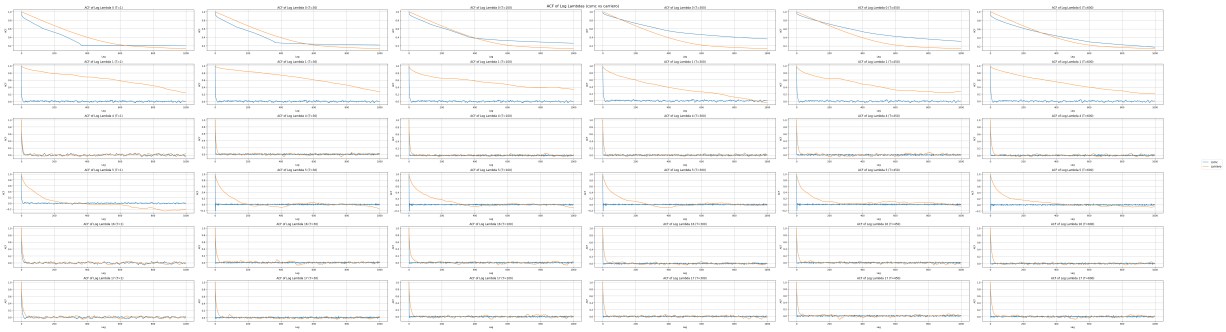
[5] Carter, C. K., Kohn, R. (1994). On Gibbs sampling for state space models. Biometrika, 81(3), 541. https://doi.org/10.2307/2337125



(a) Mixing of A

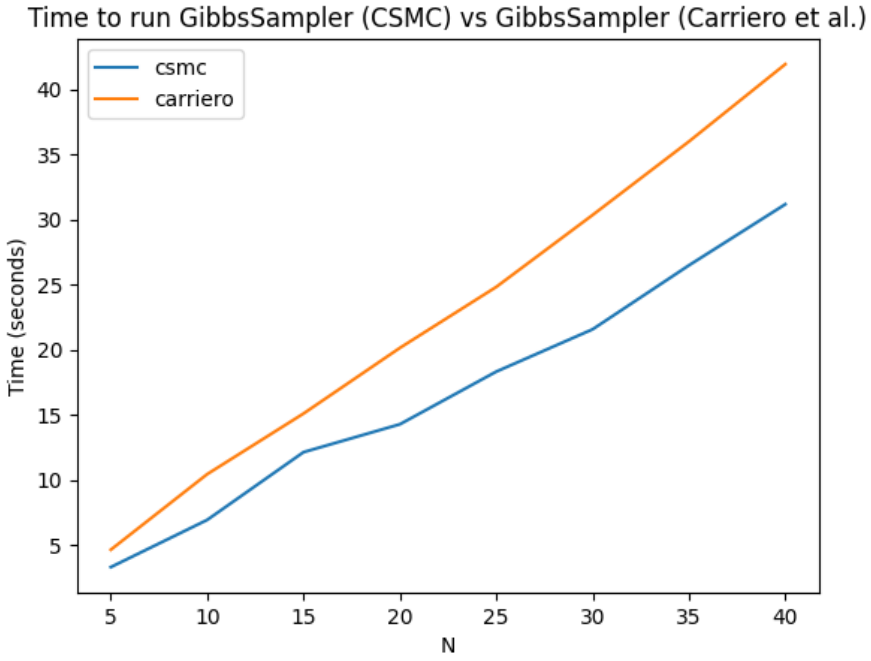


(b) Mixing of Π

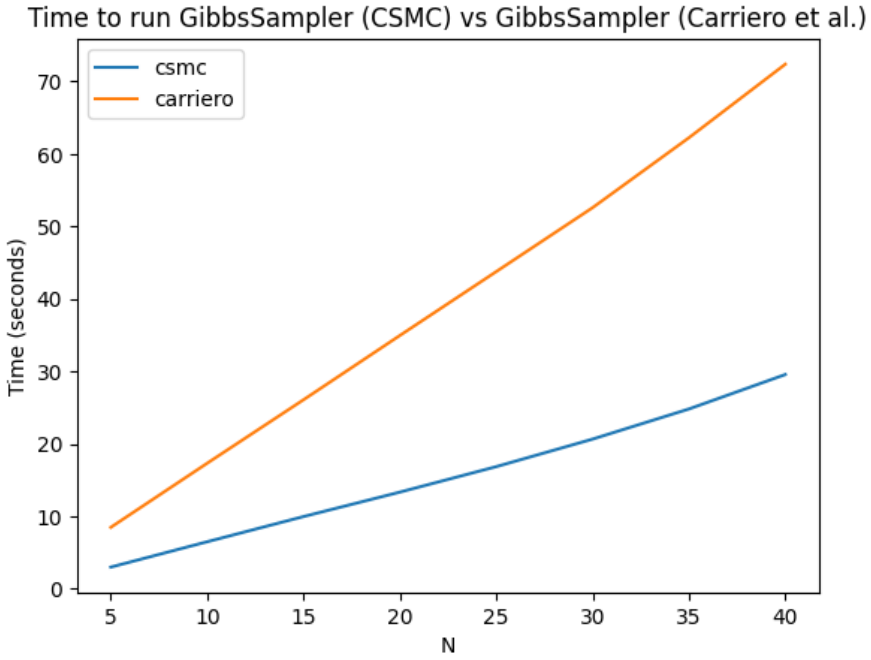


(c) Mixing of λ

Figure 3: Mixing plots for A, Π, and λ.



(a) CPU time of drawing 10 samples. 30 particles vs 20 MH steps.



(b) CPU time of drawing 10 samples. 30 particles vs 40 MH steps.

Figure 4: Comparison of CPU times for different particle-MH step combinations.