École nationale
de la statistique
et de l'administration
économique

School year 2016-2017

Hidden Markov Models & Sequential Monte Carlo Methods
-
Project based on the article "On particle filters applied to electricity load forecasting" by T. Launay, A. Philippe and S. Lamarche

Electricity load forecasting using particle filters

PROFESSOR
Nicolas CHOPIN

STUDENTS
Maxime CALLOIX
Romane PERSCH
Anas ATMANI

January 11, 2017

Table of contents

Chapter 1

Summary

Based on the article "On particle filters applied to electricity load forecasting" written by Tristan Launay, Anne Philoppe and Sophie Lamarche, we chose to apply Sequential Monte Carlo methods and more specifically particle filters to eletricity load forecasting. This approach is in particular justified by the fact that particle filters provide online inference : contrary to traditional MCMC methods (such as Gibbs sampling, see our implementation in Appendix), the addition of a new observation does not require the model to be entirely reestimated. This is particularly valuable in the context of electricity load forecasting. As a new observation is added every 30 minutes and as available data to estimate the model covers a very large period, this leads to very computational expensive reestimations of the model.

1.1 Data

We considered measurements over a 5-year period ranging from 01/01/2011 to 12/31/2015 in metropolitan France (excluding Corsica). The corresponding electricity load data is available on RTE's website (http://clients.rte-france.com/lang/an/visiteurs/vie/vie_stats_conso_inst.jsp) and the corresponding weather data is available on Meteo France's website (https://donneespubliques.meteofrance.fr/?fond=produit&id_produit=90&id_rubrique=32).

Note that on Meteo France's website, we do not have access to half-hourly temperatures. We only have access to temperatures measured every 3 hours. In practice, we would therefore not be able to produce a complete day-ahead forecast using this data. However, for academic purposes, this is not a big issue, since we are considering a different model for each half-hour. Hence, we are still able to estimate a model and produce forecasts for at least one half-hour. For example, we chose here to estimate the model at 3:00 pm[1].

Finally, note that temperatures are available for all French weather stations. In the article, they chose to use an internal EDF model to compute the temperature aggregate T_n^{heat}. In our case, we simply averaged the temperature among all weather stations after carefully excluding the ones located in Corsica or in French overseas departements and territories.

1.2 Model (for a specific half-hour)

Let y_n the observed electricity load at time n. Let $\theta = (\sigma_s, \sigma_g, u^{heat}, \kappa, \sigma)$ a fixed known parameter (which will actually be estimated using PMCMC). We also assume that the temperatures and daytypes are known, including when forecasting. Finally, we chose to neglect the cooling effect, since we did not have access to the corresponding

[1] This choice is motivated by the fact that we wanted to avoid issues due to time change.

data. The authors actually point out that the cooling effect is relatively limited in France, so we should still get reasonable forecast results. Hence, our detailed model is :

$$y_n = x_n + \nu_n \quad \text{where } \nu_n \sim \mathcal{N}(0, \sigma^2)$$

The state x_n is made of 2 parts :

$$x_n = x_n^{season} + x_n^{heat}$$

which are defined by :

$$x_n^{season} = s_n \cdot \kappa_{daytype_n}$$

$$x_n^{heat} = g_n^{heat}(T_n^{heat} - u^{heat})\mathbb{1}_{(u^{heat} > T_n^{heat})}$$

The various components are following the dynamic :

$$s_n = s_{n-1} + \epsilon_n^s \quad \text{where } \epsilon_n^s \sim \mathcal{N}(0, \sigma_{s,n}^2,]-s_{n-1}, +\infty[)$$

$$g_n^{heat} = g_{n-1}^{heat} + \epsilon_n^g \quad \text{where } \epsilon_n^g \sim \mathcal{N}(0, \sigma_{g,n}^2,]-\infty, -g_{n-1}^{heat}[)$$

$$\sigma_{s,n} = \sigma_{s,n-1} + \eta_n^s \quad \text{where } \eta_n^s \sim \mathcal{N}(0, \sigma_s^2,]-\sigma_{s,n-1}, +\infty[)$$

$$\sigma_{g,n} = \sigma_{g,n-1} + \eta_n^g \quad \text{where } \eta_n^g \sim \mathcal{N}(0, \sigma_g^2,]-\sigma_{g,n-1}, +\infty[)$$

NB : $\mathcal{N}(\mu, \Sigma, S)$ denotes the truncated Gaussian distribution with mean μ and variance Σ with support S.

1.3 Code structure

Our Python code is structured as follows :

- A module called "data_extraction_script_module" enabling the user to load the data in a pandas dataframe in a ready-to-use format (temperatures are aggregated, dates are labelled with a daytype...)
- A module called "Data_Generation" enabling to generate signals y with a given set of parameters θ : this module has mainly been used to test and debug the particle filter before estimating the parameters (it will not described in this report)
- A module called "particle_filtering_module" with 2 classes : one enables the user to run a particle filter and the other enables to run a Particle Marginal Metropolis-Hastings algorithm (PMMH) in order to estimate the parameters of the model
- A Python notebook called "Main_Computation" providing our results, after running the different algorithms mentioned above

1.4 Particle Filter

The Particle Filter is primordial and used for two main purposes. Firstly, we use it to compute the likelihood used in the PMCMC algorithms. Secondly, once the parameters have been estimated, we use the Particle Filter in order to forecast the electricity load.
As in the article, the Particle Filter takes the parameters : $\theta = (\sigma^2, \kappa, u^{heat}, \sigma_s^2, \sigma_g^2)$

1.4.1 Particle Filter initialization

The initialization of the Particle Filter is an essential step because the model is very sensitive to the initial particles, in particularly to their variances ($\sigma_{s,n}^2$ and $\sigma_{g,n}^2$). If these variances are too far (relatively to σ_s^2 and σ_g^2) from the real value, the Particle Filter will not be able to compute particles close to the reality even with a large burn-out period. The prior described in the article for the distribution of the variance are very vague (Inverse Gamma Law without moments) and so the simulation can diverge from the real value. Like the authors, we therefore used a Gibbs Sampler to initialize. We also assumed for the Gibbs sampler that $\sigma^2_{s,n}$ and $\sigma^2_{g,n}$ are constant as described in the article and therefore removing one layer of dynamic. All the calculus and convergence results are presented in the appendix.

As the impact of the factor g_n^{heat} is not observable when the temperature is below u^{heat}, we need to run the Gibbs sampler in a period with all temperatures below u^{heat} : we chose 14 days in January 2011 for this purpose.

1.4.2 Resampling

When running the Particle Filter, we observed a great degeneracy. This is not surprising. Indeed, the particles can move a lot at each iteration as their four inner variables (s_n, g_n^{heat}, $\sigma_{s,n}$ and $\sigma_{g,n}$) follow a distribution close to the random walk and so not stationnary (the variance increases with the number of iterations).
To fight against this degeneracy, we resample the particle each time the Effective Sample Size (ESS) drops below half of the number of particles (as suggested in the article). We observed a resampling almost all the time (but not always). The Figure 1.1 shows the genealogy of the particles.
However, a degenerate particle can still be kept after the resampling. At each iteration, we only use the value of $f(y_n|x_n)$, where f is the density of $Y|X$, to update the weight (before normalizing it). We have here an identifiabilty issue since $x_n = x^{season} + x^{heat}$ and so a too big x^{season} can be compensated by a too low x^{heat} during some iterations. In addition, we do not observe the impact of g_n^{heat} when the temperature is above u_{heat} and we can hence resample a particle with a degenarate g_n^{heat} during all the "summer" time.

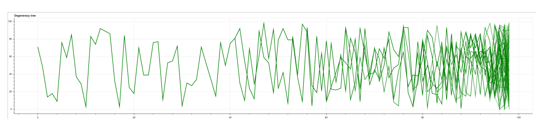


Figure 1.1 – Degeneracy of the particles : genealogy (100 particles, 100 iterations)

1.4.3 Likelihood estimation

The likelihood is computed in Particle Filter. We use it to compute the acceptance ratio in the PMMH algorithm. For computational purpose, we cannot compute the real value of the likelihood because it will be in any case to close to zero for the computer. But we do not need this value because we are only interested by the ratio between the proposal likelihood and the current point likelihood. So we can compute a log-likelihood with big constants (we use a sum instead of a mean because each Particle Filter has the same number of particles):

$$\log L_n = \log L_{n-1} + \log\left(\sum_{j=1}^{M} e^{\frac{(y_n - x_n^j)^2}{2\sigma^2}}\right) - \frac{N}{2}\log(\sigma),$$

where M is the number of particles and N the number of iterations.

1.5 PMCMC

1.5.1 Proposal

We chose a Gaussian random walk where each component of $\theta = (\sigma_s^2, \sigma_g^2, u^{heat}, \kappa, \sigma^2)$ except κ is proposed independently from the other components. This independency simplifies the simulation of the proposal, but is also justified by the fact that we have many components and it would be difficult to tune the correlation hyperparameters in addition to the variances.

However, after a first experiment, we have been obliged to turn the normal proposals for the variances into truncated normals, as some simulations of σ^2 were negative. Therefore, the proposal is not fully symmetric, and we had to take this into account in the calculation of the acceptance ratio (in reality for computational purpose we compute a log-ratio in the same way we do for the likelihood) :

$$r = \frac{p(\theta_*)L_T^N(\theta_*)h(\theta_m|\theta_*)}{p(\theta_m)L_T^N(\theta_m)h(\theta_*|\theta_m)} = \frac{p(\theta_*)L_T^N(\theta_*)\Phi(\frac{(\sigma^2)_m}{std_{h,\sigma^2}})\Phi(\frac{(\sigma_s^2)_m}{std_{h,\sigma_s^2}})\Phi(\frac{(\sigma_g^2)_m}{std_{h,\sigma_g^2}})}{p(\theta_m)L_T^N(\theta_m)\Phi(\frac{(\sigma^2)_*}{std_{h,\sigma^2}})\Phi(\frac{(\sigma_s^2)_*}{std_{h,\sigma_s^2}})\Phi(\frac{(\sigma_g^2)_*}{std_{h,\sigma_g^2}})}$$

Where :
-$L_T^N(\theta')$ is the Particle Filter estimate of the likelihood with parameter set to θ'
-θ_m is the current point
-θ_* is the proposal
-std_{h,σ_s^2}, std_{h,σ_s^2}, std_{h,σ_g^2} are the proposal standard deviation hyperparameters
-Φ is the standard normal CDF

Finally, note that we chose to consider κ as being fixed to $(\frac{1}{8}, ..., \frac{1}{8})$ where 8 is our number of daytypes. This is of course questionable but after a few experiments, we saw that fixing it was significantly improving the acceptance rate of our Metropolis-Hastings algorithm. The alternative we considered was an independent proposal $\kappa/N_{daytype} \sim \mathcal{D}_{N_{daytype}}(1, ..., 1)$, motivated by the fact that using a random walk would not lead to a sum of the eight subparameters equal to one and by the fact that this was the prior suggested in the article. Note moreover that this Dirichlet distribution is uniform, so its PDF simplifies in the calculation of the acceptance ratio.

1.5.2 Prior

The prior distribution are very vague. We used the prior suggested by the article p22, which are also the same used in the Gibbs samplers for the initialization of the Particle Filter.

1.5.3 Initialization and Choice of parameters

We ran our PMMH choosing $u^{heat} = 13$ and $\sigma_s^2 = \sigma_g^2 = \sigma^2 = 10^7$ as initial values. The choice of 10^7 was motivated by the fact that this was the order of magnitude of the variance of the observations y. We chose $u^{heat} = 13$ according to the prior suggested by the article for u^{heat}, which is a $\mathcal{N}(14, 1)$ and after noticing that on very few iterations, the accepted u^{heat} values were always lower than 14. As we had no idea about the true value of σ_s^2, σ_g^2, σ^2 and as the prior suggested in the article was not very informative, we chose to put rather large standard deviation hyperparameters. Hence, we chose : $std_{h,\sigma_s^2} = std_{h,\sigma_s^2} = std_{h,\sigma_g^2} = 10^6$ and $std_{h,u^{heat}} = 1$. The drawback of this choice is obviously that the acceptance rate will be rather low after a first stage where θ moves to its right order of magnitude. Using a Particle Filter on 1000 days (almost 3 years) with a Gibbs sampling initialization on 30 days and with 500 particles to estimate the likelihood at each iteration, the algorithm took about 16 hours to produce 300 Metropolis iterations. This very high computational time prevented us from tuning the hyperparameters more accurately to increase the acceptance rate. Finally, note that we could not reduce the length of the filtering to less than 2 years due to the high seasonality of electricity consumption.

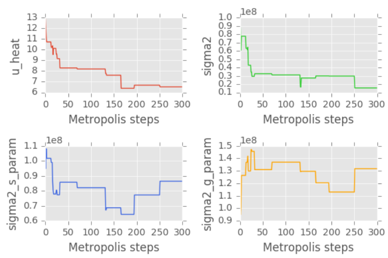
1.5.4 Results of the algorithm

The evolution of the parameters across the PMMH algorithm is displayed in the Figure 1.2.
We used the average of the distribution to estimate the parameters θ that we will use for the forecasting (see Table 1.1). The low acceptance rate is not totally surprising since the model can degenerate even with resampling if it is initialized with parameters too far from the reality. It is difficult to explain the value of the parameters. However, we can notice that the u_{heat} parameter is lower than the value in the article (around 7.3 against 14 in the article). This is lower than expected, because it means that the threshold when people really begin to consume a significant

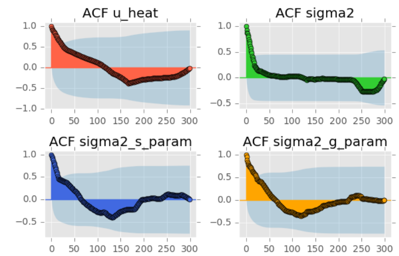
amount of electricity for heating is 7.3 (approximatively the 10% quantile of the temperature distribution). The fact that gaz consumption can also be used for heating may be an explanation of this result.

Table 1.1 – PMMH results : θ estimate (computed using a burnin = 30 iterations)

	Acceptance rate	u^{heat}	σ^2	σ_s^2	σ_g^2
PMMH	5.35 %	7.3	27E+06	79E+06	127E+06



(a) Evolution of the parameters



(b) ACF

Figure 1.2 – PMMH results

1.6 Forecasts

When forecasting, we assume as in the article that the temperature of the several next days is available because Meteo France has models that can predict very accurately the temperature. We compute the forecasts in time horizon from 1 day ahead to 5 days ahead.

The forecasts are not as good as we want to. The MAPE (Mean Average Percentile Error) is quite big (around 20%). Forecasts contain a lot of outliers which explain half of the MAPE : the MAPE without outliers is around 10% (see Figure 1.3). We did not have the time to implement a good monitoring method for these outliers because of the long time needed to run the different algorithms. In the error, we can also observe a week-seasonality : this is because we fixed κ to $(\frac{1}{8}, ..., \frac{1}{8})$ (as explained in the PMMH section), so we are not able to monitore the specific days (week-end, bank holidays...) seasonality as we do not differentiate them with different κ components (see figure 1.5.

However, our forecasting is far from being bad. We still observe an annual seasonality : forecasts of electricity loads during winter are higher than during summer (see Figure 1.4). We also observe that the difference between forecasts and real values have a standard deviation in the same order of magnitude than the σ parameter.

We observe the same result in time horizon greater than one day but with a MAPE increasing with the time horizon.

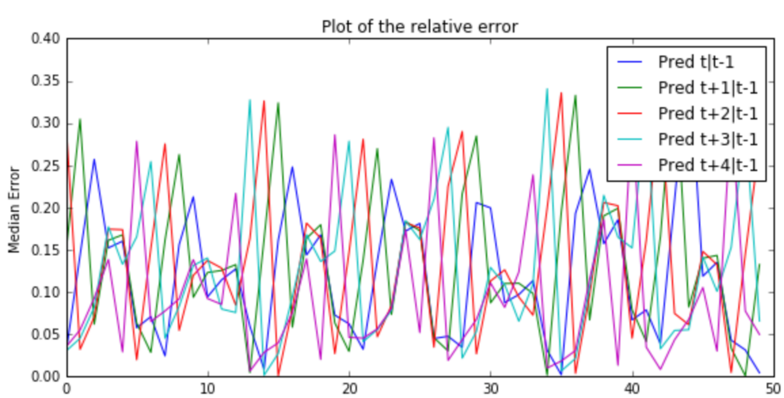


Figure 1.3 – MAPE in the Forecasting with different time horizon (50 first days, without outliers) for each iteration

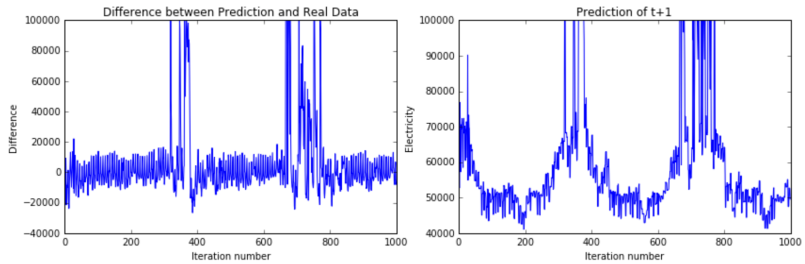


Figure 1.4 – Forecasting of Electricity Load

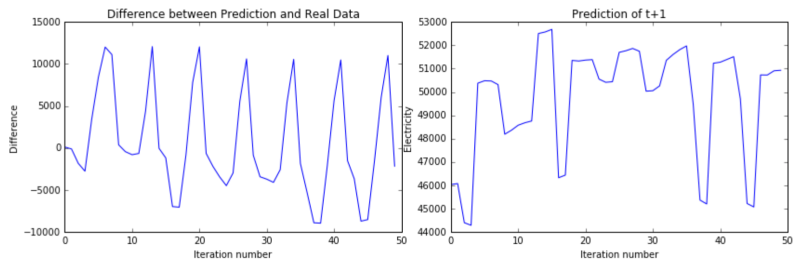


Figure 1.5 – Forecasting of Electricity Load (50 jours)

1.7 Conclusion

To conclude, during this project we have used two main applications of the Hidden Markov Models course : the Particle Filter and the Particule Marginal Metropolis Hasting in order to forecast the electricity loads in France. Of course, the results are not as good as we would like to and we had to make some simplifications from the article (we ran the model only for a single half-hour, 3:00 pm). However, the project still enlightens the qualities of Hidden Markov Models (how we can compute a likelihood using Feynman-Kac models, the resampling step...) and its difficulties (identifiability of the model in our case, bad computational rounding).

Chapter 2

Appendix

2.1 Particle Filter initialization using a Gibbs sampler

In the article, the authors mention page 19 and 20 the need of choosing a "good" initialization for the particle filter in order to avoid degeneracy after only the very first step (only one particle is selected at the first step and its weight is therefore equal to 1). They suggest to use an MCMC software such as BUGS or JAGS to estimate the smoothed distribution up to a certain time $n_0 - 1$, and then to use the resulting simulations corresponding to time $n_0 - 1$ in order to initialize the particle filter. They do not exactly precise the MCMC method they used to achieve this, but as the model is hierarchical and as they suggest to use BUGS (Bayesian inference Using Gibbs Sampling), it seems that they used a Gibb sampler.

2.1.1 Model specification

Apparently, the authors faced some issues when running the MCMC estimation with the initial model. They therefore suggest to use this simplification of the dynamics of the components:

$$s_n = s_{n-1} + \epsilon_n^s \quad \text{where } \epsilon_n^s \sim \mathcal{N}(0, \sigma_{s,n}^2,]-s_{n-1}, +\infty[)$$

$$g_n^{heat} = g_{n-1}^{heat} + \epsilon_n^g \quad \text{where } \epsilon_n^g \sim \mathcal{N}(0, \sigma_{g,n}^2,]-\infty, -g_{n-1}^{heat}[)$$

$$\sigma_{s,n} = \sigma_{s,n-1} = \sigma_{s,*}$$

$$\sigma_{g,n} = \sigma_{g,n-1} = \sigma_{g,*}$$

With initial distribution :

$$s_0 \sim \mathcal{N}(0, 10^8, \mathbb{R}_+)$$

$$g_0^{heat} \sim \mathcal{N}(0, 10^8, \mathbb{R}_-)$$

$$\sigma_{s,*}^2 \sim \mathcal{IG}(10^{-2}, 10^{-2})$$

$$\sigma_{g,*}^2 \sim \mathcal{IG}(10^{-2}, 10^{-2})$$

NB : $\mathcal{N}(\mu, \Sigma, S)$ denotes the truncated Gaussian distribution with mean μ and variance Σ with support S.

The state at time n of the model is defined by the vector $(s_n, g_n^{heat}, \sigma_{s,n}, \sigma_{g,n})$, which is equal in our case to $(s_n, g_n^{heat}, \sigma_{s,*}, \sigma_{g,*})$. Our goal is to simulate particles wrt the smoothed distribution
$\pi(s_{0:n_0-1}, g_{0:n_0-1}^{heat}, \sigma_{s,*}, \sigma_{g,*} | y_{0:n_0-1})$ using a Gibbs sampler, which will consequently provide us simulations from the filtered distribution on a "diminished state" $\pi(s_{n_0-1}, g_{n_0-1}^{heat} | y_{0:n_0-1})$ if we only consider the subprocess $(s_{n_0-1}, g_{n_0-1}^{heat})$ resulting from the Gibbs sampler.

The idea to get a proper initialization distribution for the particle filter is thus to focus here on the results on the components s and g^{heat} and to add an additional prior on σ^2_{g,n_0-1} and σ^2_{g,n_0-1} based on the empirical errors ϵ^s_n and ϵ^g_n obtained from the Gibbs sampler (see page 20 of the article for more details).

For computational reasons, we chose to simplify their approach, since we could not afford to use $n_0 = 365$ and $Q \times M$ Gibbs iterations (where M is the number of particles and Q is an arbitrary number such that the Gibbs sampler converges and therefore such that we can initialize the M particles of our particle filter with the last M iterations) as the authors did in the article, especially when running the PMCMC. We actually did not really need the Gibbs sampler to converge : we only needed a better initialization than setting every particle to 0 or to some arbitrary value. Hence, we chose to run the MCMC estimation only up to $n_0 = 30$ with $1 \times M$ Gibbs iterations. Then, since we did not run many Gibbs iterations, we chose to initialize every particle with the same value : the last Gibbs simulation at date $n_0 - 1$. We also simplified the initialization of the variances : our experiments proved that it worked better to use the last Gibbs simulation of $\sigma_{s,*}$ and $\sigma_{g,*}$ than to estimate them using the empirical errors ϵ^s_n and ϵ^g_n. This was actually enough to run a PMCMC with a reasonable acceptance rate in a reasonable time.

Finally, note that we see here that Gibbs sampling actually offers a filtering solution to our Hidden Markov Model just like Particle Filters. Nevertheless, the main difference is that it requires simulations from the whole smoothed distribution and hence it is not efficient compared to Particle Filtering which provides online inference (the next state can be simulated by only using the simulations of the previous state). Online estimation of electricity loads and forecasting are therefore more manageable using Particle Filtering.

2.1.2 Detailed calculation

In order to implement a Gibbs sampler, we need to compute the full conditionals of the the smoothed distribution $\pi(s_{0:n_0-1}, g^{heat}_{0:n_0-1}, \sigma_{s,*}, \sigma_{g,*} | y_{0:n_0-1})$.

The algorithm will then be (for θ and $\sigma^2_{g,*}, \sigma^2_{s,*}$ fixed) :
Initialize $(s_0^{(0)}, ..., s_{n_0-1}^{(0)}, g_0^{heat(0)}, ..., g_{n_0-1}^{heat(0)}, , \sigma_{s,*}^{(0)}, \sigma_{g,*}^{(0)})$
For $j = 1, 2, ...$:
Step 1 : $S_0^{(j+1)} \sim \pi(s_0 | y_{0:n_0-1}, g^{heat(j)}_{0:n_0-1}, s^{(j)}_{-0}, \sigma^{(j)}_{s,*}, \sigma^{(j)}_{g,*})$ where $s^{(j)}_{-0}$ denotes $(s^{(j)}_k)_{k\neq 0}$
Step 2: $S_1^{(j+1)} \sim \pi(s_1 | y_{0:n_0-1}, g^{heat(j)}_{0:n_0-1}, s^{(j+1)}_0, s^{(j)}_2, ..., s^{(j)}_{n_0-1}, \sigma^{(j)}_{s,*}, \sigma^{(j)}_{g,*})$
...
Step n_0 : $S_{n_0-1}^{(j+1)} \sim \pi(s_{n_0-1} | y_{0:n_0-1}, g^{heat(j)}_{0:n_0-1}, s^{(j+1)}_{-(n_0-1)}, \sigma^{(j)}_{s,*}, \sigma^{(j)}_{g,*})$ where $s^{(j+1)}_{-(n_0-1)}$ denotes $(s^{(j+1)}_k)_{k\neq n_0-1}$
Step $n_0 + 1$: $G_0^{heat(j+1)} \sim \pi(g_0^{heat} | y_{0:n_0-1}, s^{(j+1)}_{0:n_0-1}, g^{heat(j)}_{-0}, \sigma^{(j)}_{s,*}, \sigma^{(j)}_{g,*})$ where $g^{heat(j)}_{-0}$ denotes $(g^{heat(j)}_k)_{k\neq 0}$
...
Step $2n_0$: $G_{n_0-1}^{heat(j+1)} \sim \pi(g_{n_0-1}^{heat} | y_{0:n_0-1}, s^{(j+1)}_{0:n_0-1}, g^{heat(j+1)}_{-(n_0-1)}, \sigma^{(j)}_{s,*}, \sigma^{(j)}_{g,*})$ where $g^{heat(j+1)}_{-(n_0-1)}$ denotes $(g^{heat(j+1)}_k)_{k\neq n_0-1}$
Step $2n_0 + 1$: $\sigma^{(j+1)}_{s,*} \sim \pi(\sigma_{s,*} | y_{0:n_0-1}, s^{(j+1)}_{0:n_0-1}, g^{heat(j+1)}_{0:n_0-1}, \sigma^{(j)}_{g,*})$
Step $2n_0 + 2$: $\sigma^{(j+1)}_{g,*} \sim \pi(\sigma_{g,*} | y_{0:n_0-1}, s^{(j+1)}_{0:n_0-1}, g^{heat(j+1)}_{0:n_0-1}, \sigma^{(j+1)}_{s,*})$

Full conditional of s_i when $i \neq 0$ We will use denote s_{-i} the vector $(s_k)_{k\neq i}$ in order to simplify the notations in the calculus.

$$\begin{aligned}
\pi(s_i | s_{-i}, g^{heat}_{0:n_0-1}, \sigma_{s,*}, \sigma_{g,*}, y_{0:n_0-1}) &\propto \pi(s_{0:n_0-1}, y_{0:n_0-1} | g^{heat}_{0:n_0-1}, \sigma_{s,*}, \sigma_{g,*}) \quad \text{(Bayes)} \\
&\propto \pi(y_{0:n_0-1} | s_{0:n_0-1}, g^{heat}_{0:n_0-1}, \sigma_{s,*}, \sigma_{g,*}) \pi(s_{0:n_0-1} | g^{heat}_{0:n_0-1}, \sigma_{s,*}, \sigma_{g,*})
\end{aligned}$$

$s_{0:n_0-1}$ does not depend on $g_{0:n_0-1}^{heat}$ and $\sigma_{g,*}$ due to the hierarchical structure of the model, hence :

$$\begin{aligned}
&\propto \pi(y_{0:n_0-1}|s_{0:n_0-1}, g_{0:n_0-1}^{heat}, \sigma_{s,*}, \sigma_{g,*})\pi(s_{0:n_0-1}|\sigma_{s,*}) \\
&\propto \Big(\prod_{k=0}^{n_0-1} \pi(y_k|s_k, g_k^{heat})\Big)\pi(s_0|\sigma_{s,*})\pi(s_1|s_0,\sigma_{s,*})...\pi(s_{n_0-1}|s_{n_0-2},\sigma_{s,*})
\end{aligned}$$

We can now remove all components which do not depend on s_i :

$$\begin{aligned}
&\propto \pi(y_i|s_i, g_i^{heat})\pi(s_i|s_{i-1},\sigma_{s,*})\pi(s_{i+1}|s_i,\sigma_{s,*}) \\
&\propto \exp(-\frac{1}{2\sigma^2}(y_i - s_i \cdot \kappa_{daytype_i} - g_i^{heat}(T_i^{heat} - u^{heat})\mathbb{1}_{(u^{heat} > T_i^{heat})})^2) \\
&\times \exp(-\frac{1}{2\sigma_{s,*}^2}(s_i - s_{i-1})^2)\exp(-\frac{1}{2\sigma_{s,*}^2}(s_{i+1} - s_i)^2)\mathbb{1}_{s_i \geq 0}\mathbb{1}_{s_{i+1} \geq 0} \\
&\propto \exp(-\frac{1}{2\sigma^2}[-2s_i\kappa_{daytype_i}(y_i - g_i^{heat}(T_i^{heat} - u^{heat})\mathbb{1}_{(u^{heat} > T_i^{heat})})]) \\
&\times \exp(-\frac{\kappa_{daytype_i}^2}{2\sigma^2}[s_i^2])\exp(-\frac{1}{2\frac{\sigma_{s,*}^2}{2}}[s_i^2 - 2s_i(\frac{s_{i-1} + s_{i+1}}{2})])\mathbb{1}_{s_i \geq 0} \\
&\propto \exp(-\frac{1}{2\sigma^2}[-2s_i\kappa_{daytype_i}(y_i - g_i^{heat}(T_i^{heat} - u^{heat})\mathbb{1}_{(u^{heat} > T_i^{heat})})]) \\
&\times \exp(-(\frac{\kappa_{daytype_i}^2}{2\sigma^2} + \frac{1}{2\frac{\sigma_{s,*}^2}{2}})[s_i^2])\exp(-\frac{1}{2\frac{\sigma_{s,*}^2}{2}}[-2s_i(\frac{s_{i-1} + s_{i+1}}{2})])\mathbb{1}_{s_i \geq 0}
\end{aligned}$$

Since $\frac{\kappa_{daytype_i}^2}{2\sigma^2} + \frac{1}{2\frac{\sigma_{s,*}^2}{2}} = \frac{2\sigma^2 + \sigma_{s,*}^2 \cdot \kappa_{daytype_i}^2}{2\sigma_{s,*}^2\sigma^2}$, we get :

$$\begin{aligned}
&\propto \exp(-\frac{2\sigma^2 + \sigma_{s,*}^2 \cdot \kappa_{daytype_i}^2}{2\sigma_{s,*}^2\sigma^2}[-2s_i\frac{\sigma_{s,*}^2\kappa_{daytype_i}(y_i - g_i^{heat}(T_i^{heat} - u^{heat})\mathbb{1}_{(u^{heat} > T_i^{heat})})}{2\sigma^2 + \sigma_{s,*}^2 \cdot \kappa_{daytype_i}^2}]) \\
&\times \exp(-\frac{2\sigma^2 + \sigma_{s,*}^2 \cdot \kappa_{daytype_i}^2}{2\sigma_{s,*}^2\sigma^2}[s_i^2]) \\
&\times \exp(-\frac{2\sigma^2 + \sigma_{s,*}^2 \cdot \kappa_{daytype_i}^2}{2\sigma_{s,*}^2\sigma^2}[-2s_i\frac{(s_{i-1} + s_{i+1})\sigma^2}{2\sigma^2 + \sigma_{s,*}^2 \cdot \kappa_{daytype_i}^2}])\mathbb{1}_{s_i \geq 0} \\
&\propto \exp(-\frac{2\sigma^2 + \sigma_{s,*}^2 \cdot \kappa_{daytype_i}^2}{2\sigma_{s,*}^2\sigma^2}[s_i - M]^2)\mathbb{1}_{s_i \geq 0}
\end{aligned}$$

where $M = \frac{(s_{i-1}+s_{i+1})\sigma^2}{2\sigma^2+\sigma_{s,*}^2 \cdot \kappa_{daytype_i}^2} + \frac{\sigma_{s,*}^2\kappa_{daytype_i}(y_i - g_i^{heat}(T_i^{heat} - u^{heat})\mathbb{1}_{(u^{heat} > T_i^{heat})})}{2\sigma^2+\sigma_{s,*}^2 \cdot \kappa_{daytype_i}^2}$

Hence, we see that : $\boxed{s_i|s_{-i}, g_{0:n_0-1}^{heat}, \sigma_{s,*}, \sigma_{g,*}, y_{0:n_0-1} \sim \mathcal{N}(M, \frac{\sigma_{s,*}^2\sigma^2}{2\sigma^2+\sigma_{s,*}^2 \cdot \kappa_{daytype_i}^2}, \mathbb{R}^+)}$

Full conditional of s_0 Similarly as in the above calculus, we get :

$$\pi(s_0|s_{-0}, g_{0:n_0-1}^{heat}, \sigma_{s,*}, \sigma_{g,*}, y_{0:n_0-1}) \propto \Big(\prod_{k=0}^{n_0-1} \pi(y_k|s_k, g_k^{heat})\Big)\pi(s_0|\sigma_{s,*})\pi(s_1|s_0,\sigma_{s,*})...\pi(s_{n_0-1}|s_{n_0-2},\sigma_{s,*})$$

The initial distribution of s_0 does not depend on $\sigma_{s,*}$, therefore :

$$\begin{aligned}
&\propto \pi(y_0|s_0, g_0^{heat})\pi(s_0|\sigma_{s,*})\pi(s_1|s_0, \sigma_{s,*}) \\
&\propto \pi(y_0|s_0, g_0^{heat})\pi(s_0)\pi(s_1|s_0, \sigma_{s,*}) \\
&\propto \exp(-\frac{1}{2\sigma^2}(y_0 - s_0 \cdot \kappa_{daytype_0} - g_0^{heat}(T_0^{heat} - u^{heat})\mathbb{1}_{(u^{heat} > T_0^{heat})})^2) \\
&\times \exp(-\frac{s_0^2}{2 \cdot 10^8}) \exp(-\frac{1}{2\sigma_{s,*}^2}(s_1 - s_0)^2)\mathbb{1}_{s_0 \geq 0}
\end{aligned}$$

Hence, by doing a similar calculus as above we get :

$$\propto \exp(-\frac{10^8\sigma_{s,*}^2\kappa_{daytype_0}^2 + \sigma^2\sigma_{s,*}^2 + 10^8\sigma^2}{2 \cdot 10^8 \cdot \sigma^2\sigma_{s,*}^2}[s_0 - M']^2)\mathbb{1}_{s_0 \geq 0}$$

Where $M' = \frac{10^8\sigma^2 s_1 + 10^8\sigma_{s,*}^2\kappa_{daytype_0}(y_0 - g_0^{heat}(T_0^{heat} - u^{heat})\mathbb{1}_{(u^{heat} > T_0^{heat})})}{10^8\sigma_{s,*}^2\kappa_{daytype_0}^2 + \sigma^2\sigma_{s,*}^2 + 10^8\sigma^2}$

Therefore : $\boxed{s_0|s_{-0}, g_{0:n_0-1}^{heat}, \sigma_{s,*}, \sigma_{g,*}, y_{0:n_0-1} \sim \mathcal{N}(M', \frac{10^8 \cdot \sigma^2\sigma_{s,*}^2}{10^8\sigma_{s,*}^2\kappa_{daytype_0}^2 + \sigma^2\sigma_{s,*}^2 + 10^8\sigma^2}, \mathbb{R}^+)}$

Full conditional of g_i^{heat} when $i \neq 0$ Very similarly as for s_i, since both components follow a similar dynamic, we can write that :

$$\pi(g_i^{heat}|g_{-i}^{heat}, s_{0:n_0-1}, \sigma_{s,*}, \sigma_{g,*}, y_{0:n_0-1}) \propto \Big(\prod_{k=0}^{n_0-1}\pi(y_k|s_k, g_k^{heat})\Big)\pi(g_0^{heat}|\sigma_{g,*})\pi(g_1^{heat}|g_0^{heat}, \sigma_{g,*})...\pi(g_{n_0-1}^{heat}|g_{n_0-2}, \sigma_{g,*})$$

We can now remove all components which do not depend on g_i^{heat} :

$$\propto \pi(y_i|s_i, g_i^{heat})\pi(g_i^{heat}|g_{i-1}^{heat}, \sigma_{g,*})\pi(g_{i+1}^{heat}|g_i^{heat}, \sigma_{g,*})$$

After a similar calculus, we recognize that :

$$\boxed{g_i^{heat}|g_{-i}^{heat}, s_{0:n_0-1}, \sigma_{s,*}, \sigma_{g,*}, y_{0:n_0-1} \sim \mathcal{N}(\tilde{M}, \frac{\sigma^2\sigma_{g,*}^2}{2\sigma^2 + \sigma_{g,*}^2(T_i^{heat} - u^{heat})^2\mathbb{1}_{(u^{heat} > T_i^{heat})}}, \mathbb{R}^-)}$$

where $\tilde{M} = \frac{(g_{i-1}^{heat} + g_{i+1}^{heat})\sigma^2}{2\sigma^2 + \sigma_{g,*}^2(T_i^{heat} - u^{heat})^2\mathbb{1}_{(u^{heat} > T_i^{heat})}} + \frac{\sigma_{g,*}^2(T_i^{heat} - u^{heat})\mathbb{1}_{(u^{heat} > T_i^{heat})}(y_i - s_i^{heat}\kappa_{daytype_i})}{2\sigma^2 + \sigma_{g,*}^2(T_i^{heat} - u^{heat})^2\mathbb{1}_{(u^{heat} > T_i^{heat})}}$

Full conditional of g_0^{heat} Using the same arguments as to compute the full conditional of s_0, we get : $\boxed{g_0^{heat}|g_{-0}^{heat}, s_{0:n_0-1}, \sigma_{g,*}, \sigma_g}$

where : $\tilde{M}' = \frac{10^8\sigma^2 g_1^{heat} + 10^8\sigma_{g,*}^2(T_0^{heat} - u^{heat})\mathbb{1}_{(u^{heat} > T_0^{heat})}(y_0 - s_0\kappa_{daytype_0})}{10^8\sigma_{g,*}^2\kappa_{daytype_0}^2 + \sigma^2\sigma_{g,*}^2 + 10^8\sigma^2}$

Full conditional of $\sigma_{s,*}$ We will actually compute the full conditional on $\sigma_{s,*}^2$ instead of directly $\sigma_{s,*}$, since we have a prior on $\sigma_{s,*}^2$ and not on $\sigma_{s,*}$.

$$\begin{aligned}
\pi(\sigma_{s,*}^2|y_{0:n_0-1}, s_{0:n_0-1}, g_{0:n_0-1}^{heat}, \sigma_{g,*}) &\propto \pi(\sigma_{s,*}^2, y_{0:n_0-1}|s_{0:n_0-1}, g_{0:n_0-1}^{heat}, \sigma_{g,*}) \quad \text{(Bayes)} \\
&\propto \pi(y_{0:n_0-1}|s_{0:n_0-1}, g_{0:n_0-1}^{heat}, \sigma_{g,*}, \sigma_{s,*}^2)\pi(\sigma_{s,*}^2|s_{0:n_0-1}, g_{0:n_0-1}^{heat}, \sigma_{g,*}) \\
&\propto \Big(\prod_{k=0}^{n_0-1}\pi(y_k|s_k, g_k^{heat})\Big)\pi(\sigma_{s,*}^2, s_{0:n_0-1}|g_{0:n_0-1}^{heat}, \sigma_{g,*}) \quad \text{(Bayes)}
\end{aligned}$$

$\pi(y_k|s_k, g_k^{heat})$ does not depend on $\sigma_{s,*}^2$, hence :

$$\begin{aligned}
&\propto \pi(\sigma_{s,*}^2, s_{0:n_0-1}|g_{0:n_0-1}^{heat}, \sigma_{g,*}) \\
&\propto \pi(s_{0:n_0-1}|g_{0:n_0-1}^{heat}, \sigma_{g,*}, \sigma_{s,*}^2)\pi(\sigma_{s,*}^2|g_{0:n_0-1}^{heat}, \sigma_{g,*})
\end{aligned}$$

$s_{0:n_0-1}$ and $\sigma_{s,*}^2$ are independent from $g_{0:n_0-1}^{heat}$ and $\sigma_{g,*}$, hence :

$$\begin{aligned}
&\propto \pi(s_{0:n_0-1}|\sigma_{s,*}^2)\pi(\sigma_{s,*}^2) \\
&\propto \pi(s_0|\sigma_{s,*}^2)\pi(s_1|s_0, \sigma_{s,*}^2)...\pi(s_{n_0-1}|s_{n_0-2}, \sigma_{s,*}^2)\pi(\sigma_{s,*}^2)
\end{aligned}$$

s_0 is independent from $\sigma_{s,*}^2$, hence $\pi(s_0|\sigma_{s,*}^2) = \pi(s_0)$. We can thus remove it and we get :

$$\begin{aligned}
&\propto \big(\prod_{k=1}^{n_0-1} \pi(s_k|s_{k-1}, \sigma_{s,*}^2)\big)\pi(\sigma_{s,*}^2) \\
&\propto \big(\prod_{k=1}^{n_0-1} \frac{1}{\sqrt{\sigma_{s,*}^2}} \exp(-\frac{1}{2\sigma_{s,*}^2}(s_k - s_{k-1})^2)\big)\pi(\sigma_{s,*}^2)
\end{aligned}$$

Since $\sigma_{s,*}^2 \sim \mathcal{IG}(\alpha = 10^{-2}, \beta = 10^{-2})$, we have :

$$\begin{aligned}
&\propto \big(\prod_{k=1}^{n_0-1} \frac{1}{\sqrt{\sigma_{s,*}^2}} \exp(-\frac{1}{2\sigma_{s,*}^2}(s_k - s_{k-1})^2)\big)(\sigma_{s,*}^2)^{-\alpha-1} \exp(-\frac{\beta}{\sigma_{s,*}^2}) \\
&\propto \Big(\frac{1}{\sigma_{s,*}^2}\Big)^{\frac{n_0-1}{2}} \exp(-\frac{1}{2\sigma_{s,*}^2}\sum_{k=1}^{n_0-1}(s_k - s_{k-1})^2)(\sigma_{s,*}^2)^{-\alpha-1} \exp(-\frac{\beta}{\sigma_{s,*}^2}) \\
&\propto (\sigma_{s,*}^2)^{-\alpha-\frac{n_0-1}{2}-1} \exp\big(-\frac{1}{\sigma_{s,*}^2}[\beta + \frac{1}{2}\sum_{k=1}^{n_0-1}(s_k - s_{k-1})^2]\big)
\end{aligned}$$

Therefore : $\boxed{\sigma_{s,*}^2|y_{0:n_0-1}, s_{0:n_0-1}, g_{0:n_0-1}^{heat}, \sigma_{g,*} \sim \mathcal{IG}(\alpha + \frac{n_0-1}{2}, \beta + \frac{1}{2}\sum_{k=1}^{n_0-1}(s_k - s_{k-1})^2) \quad \text{where } \alpha = \beta = 10^{-2}}$

Full conditional of $\sigma_{g,*}$ Similarly, we have :

$\boxed{\sigma_{g,*}^2|y_{0:n_0-1}, s_{0:n_0-1}, g_{0:n_0-1}^{heat}, \sigma_{s,*} \sim \mathcal{IG}(\alpha + \frac{n_0-1}{2}, \beta + \frac{1}{2}\sum_{k=1}^{n_0-1}(g_k^{heat} - g_{k-1}^{heat})^2) \quad \text{where } \alpha = \beta = 10^{-2}}$