Particle MCMC and SMC^2

Ed Herbst

November 17, 2020

Embedding PF Likelihoods into Posterior Samplers

- Likelihood functions for nonlinear DSGE models can be approximated by the PF.

- We will now embed the likelihood approximation into a posterior sampler: PFMH Algorithm (a special case of PMCMC).

Embedding PF Likelihoods into Posterior Samplers

- Distinguish between:
  - $\{p(Y|\theta), p(\theta|Y), p(Y)\}$, which are related according to:

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{p(Y)}, \quad p(Y) = \int p(Y|\theta)p(\theta)d\theta$$

  - $\{\hat{p}(Y|\theta), \hat{p}(\theta|Y), \hat{p}(Y)\}$, which are related according to:

$$\hat{p}(\theta|Y) = \frac{\hat{p}(Y|\theta)p(\theta)}{\hat{p}(Y)}, \quad \hat{p}(Y) = \int \hat{p}(Y|\theta)p(\theta)d\theta.$$

- Surprising result from Andrieu et al. (2010): under certain conditions we can replace $p(Y|\theta)$ by $\hat{p}(Y|\theta)$ and still obtain draws from $p(\theta|Y)$.

PFMH Algorithm

For $i = 1$ to N:

1. Draw ϑ from a density $q(\vartheta|\theta^{i-1})$.
2. Set $\theta^i = \vartheta$ with probability

$$\alpha(\vartheta|\theta^{i-1}) = \min\left\{1,\ \frac{\hat{p}(Y|\vartheta)p(\vartheta)/q(\vartheta|\theta^{i-1})}{\hat{p}(Y|\theta^{i-1})p(\theta^{i-1})/q(\theta^{i-1}|\vartheta)}\right\}$$

   and $\theta^i = \theta^{i-1}$ otherwise. The likelihood approximation $\hat{p}(Y|\vartheta)$ is computed using a particle filter.

Why Does the PFMH Work?

- At each iteration the filter generates draws $\tilde{s}_t^j$ from the proposal distribution $g_t(\cdot|s_{t-1}^j)$.

- Let $\tilde{S}_t = \left(\tilde{s}_t^1, \ldots, \tilde{s}_t^M\right)'$ and denote the entire sequence of draws by $\tilde{S}_{1:T}^{1:M}$.

- Selection step: define a random variable A_t^j that contains this ancestry information. For instance, suppose that during the resampling particle $j = 1$ was assigned the value $\tilde{s}_t^{10}$ then $A_t^1 = 10$. Let $A_t = \left(A_t^1, \ldots, A_t^N\right)$ and use $A_{1:T}$ to denote the sequence of A_t's.

- PFMH operates on an enlarged probability space: θ, $\tilde{S}_{1:T}$ and $A_{1:T}$.

Why Does the PFMH Work?

- Use $U_{1:T}$ to denote random vectors for $\tilde{S}_{1:T}$ and $A_{1:T}$. $U_{1:T}$ is an array of *iid* uniform random numbers.
- The transformation of $U_{1:T}$ into $(\tilde{S}_{1:T}, A_{1:T})$ typically depends on θ and $Y_{1:T}$, because the proposal distribution $g_t(\tilde{s}_t|s^j_{t-1})$ depends both on the current observation y_t as well as the parameter vector θ.
- E.g., implementation of conditionally-optimal PF requires sampling from a $N(\bar{s}^j_{t|t}, P_{t|t})$ distribution for each particle j. Can be done using a prob integral transform of uniform random variables.
- We can express the particle filter approximation of the likelihood function as

$$\hat{p}(Y_{1:T}|\theta) = g(Y_{1:T}|\theta, U_{1:T}).$$

where

$$U_{1:T} \sim p(U_{1:T}) = \prod_{t=1}^{T} p(U_t).$$

Why Does the PFMH Work?

Define the joint distribution

$$p_g\big(Y_{1:T}, \theta, U_{1:T}\big) = g(Y_{1:T}|\theta, U_{1:T})p\big(U_{1:T}\big)p(\theta).$$

The PFMH algorithm samples from the joint posterior

$$p_g\big(\theta, U_{1:T}|Y_{1:T}\big) \propto g(Y|\theta, U_{1:T})p\big(U_{1:T}\big)p(\theta)$$

and discards the draws of $\big(U_{1:T}\big)$. For this procedure to be valid, it needs to be the case that PF approximation is unbiased:

$$\mathbb{E}[\hat{p}(Y_{1:T}|\theta)] = \int g(Y_{1:T}|\theta, U_{1:T})p\big(U_{1:T}\big)d\theta = p(Y_{1:T}|\theta).$$

Why Does the PFMH Work?

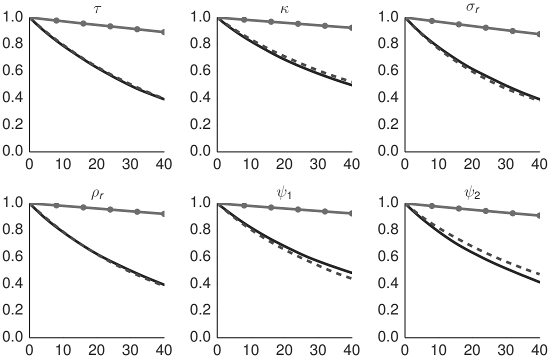
- We can express acceptance probability directly in terms of $\hat{p}(Y_{1:T}|\theta)$.
- Need to generate a proposed draw for both θ and $U_{1:T}$: ϑ and $U^*_{1:T}$.
- The proposal distribution for $(\vartheta, U^*_{1:T})$ in the MH algorithm is given by $q(\vartheta|\theta^{(i-1)})p(U^*_{1:T})$.
- No need to keep track of the draws $(U^*_{1:T})$.
- MH acceptance probability:

$$
\begin{aligned}
\alpha(\vartheta|\theta^{i-1}) &= \min \left\{ 1, \frac{\frac{g(Y|\vartheta, U^*)p(U^*)p(\vartheta)}{q(\vartheta|\theta^{(i-1)})p(U^*)}}{\frac{g(Y|\theta^{(i-1)}, U^{(i-1)})p(U^{(i-1)})p(\theta^{(i-1)})}{q(\theta^{(i-1)}|\theta^*)p(U^{(i-1)})}} \right\} \\
&= \min \left\{ 1, \frac{\hat{p}(Y|\vartheta)p(\vartheta)\big/q(\vartheta|\theta^{(i-1)})}{\hat{p}(Y|\theta^{(i-1)})p(\theta^{(i-1)})\big/q(\theta^{(i-1)}|\vartheta)} \right\}.
\end{aligned}
$$

Small-Scale DSGE: Accuracy of MH Approximations

- Results are based on $N_{run} = 20$ runs of the PF-RWMH-V algorithm.
- Each run of the algorithm generates $N = 100,000$ draws and the first $N_0 = 50,000$ are discarded.
- The likelihood function is computed with the Kalman filter (KF), bootstrap particle filter (BS-PF, $M = 40,000$) or conditionally-optimal particle filter (CO-PF, $M = 400$).
- "Pooled" means that we are pooling the draws from the $N_{run} = 20$ runs to compute posterior statistics.

Autocorrelation of PFMH Draws



Notes: The figure depicts autocorrelation functions computed from the output of the 1 Block RWMH-V algorithm based on the Kalman filter (solid), the conditionally-optimal particle filter (dashed) and the bootstrap particle filter (solid with dots).

Small-Scale DSGE: Accuracy of MH Approximations

	Posterior Mean (Pooled)			Inefficiency Factors			Std Dev of Means		
	KF	CO-PF	BS-PF	KF	CO-PF	BS-PF	KF	CO-PF	BS-PF
τ	2.63	2.62	2.64	66.17	126.76	1360.22	0.020	0.028	0.091
κ	0.82	0.81	0.82	128.00	97.11	1887.37	0.007	0.006	0.026
ψ_1	1.88	1.88	1.87	113.46	159.53	749.22	0.011	0.013	0.029
ψ_2	0.64	0.64	0.63	61.28	56.10	681.85	0.011	0.010	0.036
ρ_r	0.75	0.75	0.75	108.46	134.01	1535.34	0.002	0.002	0.007
ρ_g	0.98	0.98	0.98	94.10	88.48	1613.77	0.001	0.001	0.002
ρ_z	0.88	0.88	0.88	124.24	118.74	1518.66	0.001	0.001	0.005
$r^{(A)}$	0.44	0.44	0.44	148.46	151.81	1115.74	0.016	0.016	0.044
$\pi^{(A)}$	3.32	3.33	3.32	152.08	141.62	1057.90	0.017	0.016	0.045
$\gamma^{(Q)}$	0.59	0.59	0.59	106.68	142.37	899.34	0.006	0.007	0.018
σ_r	0.24	0.24	0.24	35.21	179.15	1105.99	0.001	0.002	0.004
σ_g	0.68	0.68	0.67	98.22	64.18	1490.81	0.003	0.002	0.011
σ_z	0.32	0.32	0.32	84.77	61.55	575.90	0.001	0.001	0.003
$\ln \hat{p}(Y)$	-357.14	-357.17	-358.32				0.040	0.038	0.949

SW Model: Accuracy of MH Approximations

- Results are based on $N_{run} = 20$ runs of the PF-RWMH-V algorithm.

- Each run of the algorithm generates $N = 10,000$ draws.

- The likelihood function is computed with the Kalman filter (KF) or conditionally-optimal particle filter (CO-PF).

- "Pooled" means that we are pooling the draws from the $N_{run} = 20$ runs to compute posterior statistics. The CO-PF uses $M = 40,000$ particles to compute the likelihood.

SW Model: Accuracy of MH Approximations

	Post. Mean (Pooled)		Ineff. Factors		Std Dev of Means	
	KF	CO-PF	KF	CO-PF	KF	CO-PF
$(100\beta^{-1} - 1)$	0.14	0.14	172.58	3732.90	0.007	0.034
$\bar{\pi}$	0.73	0.74	185.99	4343.83	0.016	0.079
$\bar{l}$	0.51	0.37	174.39	3133.89	0.130	0.552
α	0.19	0.20	149.77	5244.47	0.003	0.015
σ_c	1.49	1.45	86.27	3557.81	0.013	0.086
Φ	1.47	1.45	134.34	4930.55	0.009	0.056
φ	5.34	5.35	138.54	3210.16	0.131	0.628
h	0.70	0.72	277.64	3058.26	0.008	0.027
ξ_w	0.75	0.75	343.89	2594.43	0.012	0.034
σ_l	2.28	2.31	162.09	4426.89	0.091	0.477
ξ_p	0.72	0.72	182.47	6777.88	0.008	0.051
ι_w	0.54	0.53	241.80	4984.35	0.016	0.073
ι_p	0.48	0.50	205.27	5487.34	0.015	0.078
ψ	0.45	0.44	248.15	3598.14	0.020	0.078
r_π	2.09	2.09	98.32	3302.07	0.020	0.116
ρ	0.80	0.80	241.63	4896.54	0.006	0.025
r_y	0.13	0.13	243.85	4755.65	0.005	0.023
$r_{\Delta y}$	0.21	0.21	101.94	5324.19	0.003	0.022

SW Model: Accuracy of MH Approximations

	Post. Mean (Pooled)		Ineff. Factors		Std Dev of Means	
	KF	CO-PF	KF	CO-PF	KF	CO-PF
ρ_a	0.96	0.96	153.46	1358.87	0.002	0.005
ρ_b	0.22	0.21	325.98	4468.10	0.018	0.068
ρ_g	0.97	0.97	57.08	2687.56	0.002	0.011
ρ_i	0.71	0.70	219.11	4735.33	0.009	0.044
ρ_r	0.54	0.54	194.73	4184.04	0.020	0.094
ρ_p	0.80	0.81	338.69	2527.79	0.022	0.061
ρ_w	0.94	0.94	135.83	4851.01	0.003	0.019
ρ_{ga}	0.41	0.37	196.38	5621.86	0.025	0.133
μ_p	0.66	0.66	300.29	3552.33	0.025	0.087
μ_w	0.82	0.81	218.43	5074.31	0.011	0.052
σ_a	0.34	0.34	128.00	5096.75	0.005	0.034
σ_b	0.24	0.24	186.13	3494.71	0.004	0.016
σ_g	0.51	0.49	208.14	2945.02	0.006	0.021
σ_i	0.43	0.44	115.42	6093.72	0.006	0.043
σ_r	0.14	0.14	193.37	3408.01	0.004	0.016
σ_p	0.13	0.13	194.22	4587.76	0.003	0.013
σ_w	0.22	0.22	211.80	2256.19	0.004	0.012
$\ln \hat{p}(Y)$	-964	-1018			0.298	9.139

Computational Considerations

- We implement the PFMH algorithm on a single machine, utilizing up to twelve cores.
- For the small-scale DSGE model it takes 30:20:33 [hh:mm:ss] hours to generate 100,000 parameter draws using the bootstrap PF with 40,000 particles. Under the conditionally-optimal filter we only use 400 particles, which reduces the run time to 00:39:20 minutes.
- For the SW model it took 05:14:20:00 [dd:hh:mm:ss] days to generate 10,000 draws using the conditionally-optimal PF with 40,000 particles.

SMC^2

- We will construct an *SMC*^{2} algorithm to estimate a DSGE model:
  - we use SMC for inference about the static parameter θ;
  - we use SMC to obtain a particle filter approximation of the likelihood function.

  and document its accuracy.

- Rather than delving straight into the *SMC*^{2} algorithm we proceed in a step-wise manner:
  - discuss how SMC can be used for inference about θ in models in which the likelihood function can be evaluated with the Kalman filter; conduct simulation experiments to document the accuracy of SMC approximation of posterior moments;
  - review how particle filters can be used to construct a Monte Carlo approximation of the likelihood function and conduct simulation experiments to document the accuracy.

Why???

- Likelihood evaluation for nonlinear DSGE models requires nonlinear filtering $\longrightarrow$ sequential Monte Carlo.

- For inference about the static parameter θ, "standard" MCMC methods can be quite inaccurate. Multimodal posteriors may arise because it is difficult to
  - disentangling internal and external propagation mechanisms;
  - disentangling the relative importance of shocks.

Putting the Pieces Together – *SMC*^{2}

- Start from SMC algorithm ... replace actual likelihood by particle filter approximation in the correction and mutation steps of SMC algorithm.
- Data tempering instead of likelihood tempering: $\pi_n^D(\theta) = p(\theta|Y_{1:t_n})$.
- Key Idea: let

$$\hat{p}(Y_{1:t_n}|\theta_n) = g(Y_{1:t_n}|\theta_n, U_{1:t_n}).$$

  where $U_{1:t_n} \sim p(U_{1:t_n})$ is an array of *iid* uniform random variables generated by the particle filter.
- Important Result: Particle filter delivers an unbiased estimate of the incremental weight $p(Y_{t_{n-1}+1:t_n}|\theta)$:

$$\int g(Y_{1:t_n}|\theta_n, U_{1:t_n}) p(U_{1:t_n}) dU_{1:t_n} = p(Y_{1:t_n}|\theta_n).$$

Particle System for SMC^2 Sampler After Stage n

Parameter	State			
(θ_n^1, W_n^1)	$(s_{t_n}^{1,1}, \mathcal{W}_{t_n}^{1,1})$	$(s_{t_n}^{1,2}, \mathcal{W}_{t_n}^{1,2})$	$\cdots$	$(s_{t_n}^{1,M}, \mathcal{W}_{t_n}^{1,M})$
(θ_n^2, W_n^2)	$(s_{t_n}^{2,1}, \mathcal{W}_{t_n}^{2,1})$	$(s_{t_n}^{2,2}, \mathcal{W}_{t_n}^{2,2})$	$\cdots$	$(s_{t_n}^{2,M}, \mathcal{W}_{t_n}^{2,M})$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
(θ_n^N, W_n^N)	$(s_{t_n}^{N,1}, \mathcal{W}_{t_n}^{N,1})$	$(s_{t_n}^{N,2}, \mathcal{W}_{t_n}^{N,2})$	$\cdots$	$(s_{t_n}^{N,M}, \mathcal{W}_{t_n}^{N,M})$

To simplify notation, we add one observation at a time, $n = t$, and write θ_t and $\pi_t(\cdot)$.

SMC^2

1. **Initialization.** Draw the initial particles from the prior: $\theta_0^i \overset{iid}{\sim} p(\theta)$ and $W_0^i = 1$, $i = 1, \ldots, N$.
2. **Recursion.** For $t = 1, \ldots, T$,

   2.1 **Correction.** Reweight the particles from stage $t-1$ by defining the incremental weights

   $$\tilde{w}_t^i = \hat{p}(y_t | Y_{1:t-1}, \theta_{t-1}^i) = g(y_t | Y_{1:t-1}, \theta_{t-1}^i, U_{1:t}^i)$$

   and the normalized weights

   $$\tilde{W}_t^i = \frac{\tilde{w}_n^i W_{t-1}^i}{\frac{1}{N}\sum_{i=1}^N \tilde{w}_t^i W_{t-1}^i}, \quad i = 1, \ldots, N.$$

   Then,

   $$\tilde{h}_{t,N} = \frac{1}{N}\sum_{i=1}^N \tilde{W}_t^i h(\theta_{t-1}^i) \approx \mathbb{E}_{\pi_t}[h(\theta)].$$

   2.2 **Selection.** (unchanged)

   2.3 **Mutation.**

SMC^2

1. **Initialization.**
2. **Recursion.** For $t = 1, \ldots, T$,
   - 2.1 **Correction.**
   - 2.2 **Selection.**
   - 2.3 **Mutation.** Propagate the particles $\{\hat{\theta}_t^i, W_t^i\}$ via 1 step of an MH algorithm. The proposal distribution is given by

$$q(\vartheta_t^i|\hat{\theta}_t^i)p(U_{1:t}^{*i})$$

and the acceptance ratio can be expressed as

$$\alpha(\vartheta_t^i|\hat{\theta}_t^i) = \min \left\{ 1, \frac{g(Y_{1:t}|\vartheta_t^i, U_{1:t}^{*i})p(\vartheta_t^i)p(U_{1:t}^{*i})/q(\vartheta_t^i|\hat{\theta}_t^i)p(U_{1:t}^{*i})}{g(Y_{1:t}|\hat{\theta}_t^i, U_{1:t}^{i})p(\hat{\theta}_t^i)p(U_{1:t}^{i})/q(\hat{\theta}_t^i|\vartheta_t^i)p(U_{1:t}^{i})} \right\}.$$

Then,

$$\bar{h}_{t,N} = \frac{1}{N}\sum_{i=1}^{N} h(\theta_t^i)W_t^i \approx \mathbb{E}_{\pi_t}[h(\theta)].$$

Why Does *SMC*^{2} Work?

- Work on enlarged probability space that includes sequence of random vectors $U^i_{1:t-1}$ that underlies the simulation approximation of the particle filter.
- At the end of iteration $t-1$:
  - Particles $\{\theta^i_{t-1}, U^i_{1:t-1}, W^i_{t-1}\}_{i=1}^N$.
  - For each parameter value θ^i_{t-1} there is PF approx of the likelihood: $\hat{p}(Y_{1:t-1}|\theta^i_{t-1}) = g(Y_{1:t-1}|\theta^i_{t-1}, U^i_{1:t-1})$.
  - Swarm of particles $\{s^{i,j}_{t-1}, \mathcal{W}^{i,j}_{t-1}\}_{j=1}^M$ that represents the distribution $p(s_{t-1}|Y_{1:t-1}, \theta^i_{t-1})$.
- The triplets $\{\theta^i_{t-1}, U^i_{1:t-1}, W^i_{t-1}\}_{i=1}^N$ approximate:

$$\begin{aligned}
&\int\int h(\theta, U_{1:t-1}) p(U_{1:t-1}) p(\theta|Y_{1:t-1}) dU_{1:t-1} d\theta \\
&\approx \quad \frac{1}{N}\sum_{i=1}^N h(\theta^i_{t-1}, U^i_{1:t-1}) W^i_{t-1}.
\end{aligned}$$

Correction Step

- Write the particle filter approximation of the likelihood increment as
$$\tilde{w}_t^i = \hat{p}(y_t|Y_{1:t-1}, \theta_{t-1}^i) = g(y_t|Y_{1:t-1}, U_{1:t}^i, \theta_{t-1}^i).$$
- By induction, we can deduce that $\frac{1}{N}\sum_{i=1}^N h(\theta_{t-1}^i)\tilde{w}_t^i W_{t-1}^i$ approximates the following integral
$$\begin{aligned}&\int\int h(\theta)g(y_t|Y_{1:t-1}, U_{1:t}, \theta)p(U_{1:t})p(\theta|Y_{1:t-1})dU_{1:t}d\theta \\ &= \int h(\theta)\left[\int g(y_t|Y_{1:t-1}, U_{1:t}, \theta)p(U_{1:t})dU_{1:t}\right]p(\theta|Y_{1:t-1})d\theta.\end{aligned}$$
- Provided that the particle filter approximation of the likelihood increment is unbiased, that is,
$$\int g(y_t|Y_{1:t-1}, U_{1:t}, \theta)p(U_{1:t})dU_{1:t} = p(y_t|Y_{1:t-1}, \theta)$$
for each θ, we deduce that $\tilde{h}_{t,N}$ is a consistent estimator of $\mathbb{E}_{\pi_t}[h(\theta)]$.

Selection Step

- Similar to regular SMC.
- We resample in every period for expositional purposes.
- We are keeping track of the ancestry information in the vector $\mathcal{A}_t$. This is important, because for each resampled particle i we not only need to know its value $\hat{\theta}_t^i$ but we also want to track the corresponding value of the likelihood function $\hat{p}(Y_{1:t}|\hat{\theta}_t^i)$ as well as the particle approximation of the state, given by $\{s_t^{i,j}, \mathcal{W}_t^{i,j}\}$, and the set of random numbers $U_{1:t}^i$.
- In the implementation, the likelihood values are needed for the mutation step.
- The $U_{1:t}^i$'s are not required for

Mutation Step

- For each particle i we have:
  - A proposed value ϑ_t^i;
  - A sequence of random vectors $U_{1:t}^*$ drawn from the distribution $p(U_{1:t})$;
  - An associated particle filter approximation of the likelihood:

$$\hat{p}(Y_{1:t}|\vartheta_t^i) = g(Y_{1:t}|\vartheta_t^i, U_{1:t}^*).$$

- The densities $p(U_{1:t}^i)$ and $p(U_{1:t}^*)$ cancel from the formula for the acceptance probability $\alpha(\vartheta_t^i|\hat{\theta}_t^i)$:

$$\begin{aligned}
\alpha(\vartheta|\theta^{i-1}) &= \min\left\{1, \frac{\frac{g(Y|\vartheta,U^*)p(U^*)p(\vartheta)}{q(\vartheta|\theta^{(i-1)})p(U^*)}}{\frac{g(Y|\theta^{(i-1)},U^{(i-1)})p(U^{(i-1)})p(\theta^{(i-1)})}{q(\theta^{(i-1)}|\theta^*)p(U^{(i-1)})}}\right\} \\
&= \min\left\{1, \frac{\hat{p}(Y|\vartheta)p(\vartheta)/q(\vartheta|\theta^{(i-1)})}{\hat{p}(Y|\theta^{(i-1)})p(\theta^{(i-1)})/q(\theta^{(i-1)}|\vartheta)}\right\}.
\end{aligned}$$

Application to Small-Scale DSGE Model

- Results are based on $N_{run} = 20$ runs of the SMC^2 algorithm with $N = 4,000$ particles.
- D is data tempering and L is likelihood tempering.
- KF is Kalman filter, CO-PF is conditionally-optimal PF with $M = 400$, BS-PF is bootstrap PF with $M = 40,000$. CO-PF and BS-PF use data tempering.

Accuracy of *SMC*^{2} Approximations

	Posterior Mean (Pooled)				Inefficiency Factors				Std Dev of Means			
	KF(L)	KF(D)	CO-PF	BS-PF	KF(L)	KF(D)	CO-PF	BS-PF	KF(L)	KF(D)	CO-PF	BS-PF
τ	2.65	2.67	2.68	2.53	1.51	10.41	47.60	6570	0.01	0.03	0.07	0.76
κ	0.81	0.81	0.81	0.70	1.40	8.36	40.60	7223	0.00	0.01	0.01	0.18
ψ_1	1.87	1.88	1.87	1.89	3.29	18.27	22.56	4785	0.01	0.02	0.02	0.27
ψ_2	0.66	0.66	0.67	0.65	2.72	10.02	43.30	4197	0.01	0.02	0.03	0.34
ρ_r	0.75	0.75	0.75	0.72	1.31	11.39	60.18	14979	0.00	0.00	0.01	0.08
ρ_g	0.98	0.98	0.98	0.95	1.32	4.28	250.34	21736	0.00	0.00	0.00	0.04
ρ_z	0.88	0.88	0.88	0.84	3.16	15.06	35.35	10802	0.00	0.00	0.00	0.05
$r^{(A)}$	0.45	0.46	0.44	0.46	1.09	26.58	73.78	7971	0.00	0.02	0.04	0.42
$\pi^{(A)}$	3.32	3.31	3.31	3.56	2.15	40.45	158.64	6529	0.01	0.03	0.06	0.40
$\gamma^{(Q)}$	0.59	0.59	0.59	0.64	2.35	32.35	133.25	5296	0.00	0.01	0.03	0.16
σ_r	0.24	0.24	0.24	0.26	0.75	7.29	43.96	16084	0.00	0.00	0.00	0.06
σ_g	0.68	0.68	0.68	0.73	1.30	1.48	20.20	5098	0.00	0.00	0.00	0.08
σ_z	0.32	0.32	0.32	0.42	2.32	3.63	26.98	41284	0.00	0.00	0.00	0.11
$\ln p(Y)$	-358.75	-357.34	-356.33	-340.47					0.120	1.191	4.374	14.49

Computational Considerations

- The SMC^2 results are obtained by utilizing 40 processors.

- We parallelized the likelihood evaluations $\hat{p}(Y_{1:t}|\theta_t^i)$ for the θ_t^i particles rather than the particle filter computations for the swarms $\{s_t^{i,j}, \mathcal{W}_t^{i,j}\}_{j=1}^M$.

- The run time for the SMC^2 with conditionally-optimal PF ($N = 4,000$, $M = 400$) is 23:24 [mm:ss] minutes, where as the algorithm with bootstrap PF ($N = 4,000$ and $M = 40,000$) runs for 08:05:35 [hh:mm:ss] hours.

- Due to memory constraints we re-computed the entire likelihood for $Y_{1:t}$ in each iteration.

Conclusion

- We explored PMCMC and SMC^2 methods for DSGE models.
- These methods are promising, because they can handle multi-modal posterior surfaces and they can be parallelized.
- However, careful tuning is required and the particle filter approximation of the likelihood function needs to be sufficiently accurate.
- The method worked well for a small-scale DSGE model, but not for the Smets-Wouters model, because there was too much noise in the likelihood approximation.

References

Andrieu, C., A. Doucet, and R. Holenstein (2010): "Particle Markov chain Monte Carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72, 269 342.