Sequential Monte Carlo Methods (Squared!)for DSGE Models[1]

Ed Herbst [1] Frank Schorfheide [2]

[1]Federal Reserve Board

[2]University of Pennsylvania, CEPR, and NBER

April 22, 2015

[1]The views expressed in this paper are those of the authors and do not necessarily reflect the views of the Federal Reserve Board of Governors or the Federal Reserve System.

Introduction

- DSGE model: dynamic model of the macroeconomy, indexed by θ – vector of preference and technology parameters. Used for forecasting, policy experiments, interpreting past events.

- In nearly all applications, DSGE models are estimated using Bayesian methods:

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{p(Y)} \propto p(Y|\theta)p(\theta).$$

  Analyzing the posterior is difficult, because the likelihood is intractable.

- "Standard" approach for linearized models (Schorfheide, 2000; Otrok, 2001): use Kalman filter to evaluate likelihood function and Markov chain Monte Carlo (MCMC) methods to implement posterior inference for θ.

Sequential Monte Carlo (SMC) Methods

SMC can be used to

- approximate the posterior of θ: Chopin (2002) ... Creal (2007), Herbst and Schorfheide (2014)
- approximate the likelihood function (particle filtering): Gordon, Salmond, and Smith (1993) ... Fernandez-Villaverde and Rubio-Ramirez (2007)
- or both: SMC^2: Chopin, Jacob, and Papaspiliopoulos (2012) ... this talk

This Talk

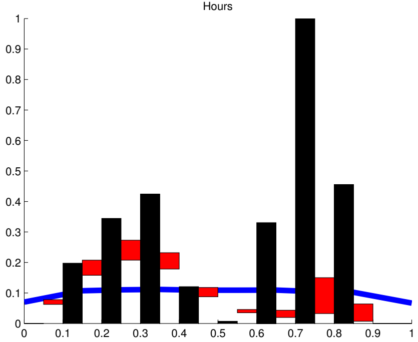
- We will construct an SMC^2 algorithm to estimate a DSGE model:
  - we use SMC for inference about the static parameter θ;
  - we use SMC to obtain a particle filter approximation of the likelihood function.

  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.



The Laboratory

1. A state-space model with a global identification problem:

$$y_t = [1\ 1]s_t, \quad s_t = \left[\begin{array}{cc} \theta_1^2 & 0 \\ (1-\theta_1^2) - \theta_1\theta_2 & (1-\theta_1^2) \end{array} \right] s_{t-1} + \left[\begin{array}{c} 1 \\ 0 \end{array} \right] \epsilon_t.$$

2. Small scale DSGE model (Euler equation, NK Phillips curve, monetary policy rule, three AR(1) exogenous shock processes) estimated on output growth, inflation, and nominal interest rates.

3. Smets-Wouters model estimated based on seven observables.

Some Notation

- Posterior distribution $\pi(\theta) = p(\theta|Y)$ with moments $\mathbb{E}_\pi[\theta]$ and $\mathbb{V}_\pi[\theta]$.

- We often consider functions $h(\theta)$ and write

$$\bar{h}_N = \frac{1}{N} \sum_{i=1}^{N} h(\theta^i)$$

where θ^i's are draws from posterior.

- Note that under *iid* sampling under suitable regularity conditions

$$\sqrt{N}(\bar{h}_N - \mathbb{E}_\pi[h]) \Longrightarrow N\big(0, \mathbb{V}_\pi[h]\big)$$

- Inefficiency factor:

$$\text{InEff}_N = \frac{\mathbb{V}[\bar{h}_N]}{\mathbb{V}_\pi[h]/N}$$

Part I: From Importance Sampling to Sequential Importance Sampling

- Bayes Theorem:

$$\pi(\theta) = \frac{p(Y|\theta)p(\theta)}{p(Y)} = \frac{f(\theta)}{Z}.$$

- Let $g(\theta)$ be a pdf. Write posterior mean of $h(\theta)$ as:

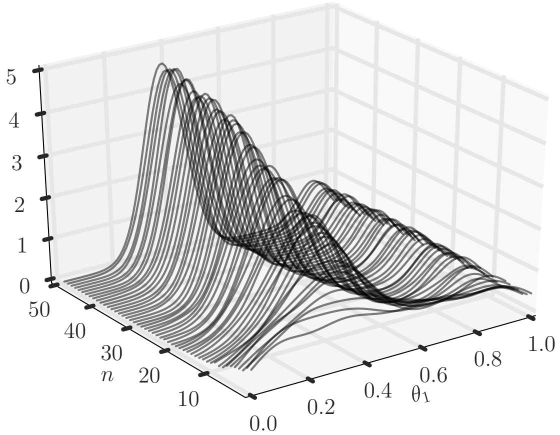
$$\mathbb{E}_\pi[h(\theta)] = \int h(\theta)\frac{f(\theta)}{Z}d\theta = \frac{1}{Z}\int h(\theta)\frac{f(\theta)}{g(\theta)}g(\theta)d\theta$$

- If θ^i's are draws from $g(\cdot)$ then

$$\mathbb{E}_\pi[h] = \frac{\int h(\theta)\frac{f(\theta)}{g(\theta)}g(\theta)d\theta}{\int \frac{f(\theta)}{g(\theta)}g(\theta)d\theta} \approx \frac{\frac{1}{N}\sum_{i=1}^N h(\theta^i)w(\theta^i)}{\frac{1}{N}\sum_{i=1}^N w(\theta^i)}, \quad w(\theta) = \frac{f(\theta)}{g(\theta)}.$$

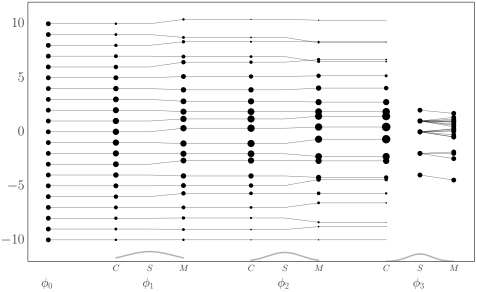
- In general, it's hard to construct a good proposal density $g(\theta)$...

Illustration



$$\pi_n(\theta) = \frac{[p(Y|\theta)]^{\phi_n} p(\theta)}{\int [p(Y|\theta)]^{\phi_n} p(\theta) d\theta} = \frac{f_n(\theta)}{Z_n}, \quad \phi_n = \left(\frac{n}{N_\phi}\right)^\lambda$$

SMC Algorithm for θ: A Graphical Illustration



- $\pi_n(\theta)$ is represented by a swarm of particles $\{\theta_n^i, W_n^i\}_{i=1}^N$:

$$\bar{h}_{n,N} = \frac{1}{N}\sum_{i=1}^{N} W_n^i h(\theta^i) \xrightarrow{a.s.} \mathbb{E}_{\pi_n}[h(\theta_n)].$$

- C is Correction; S is Selection; and M is Mutation.

SMC Algorithm

1. **Initialization.** $(\phi_0 = 0)$. Draw the initial particles from the prior: $\theta_1^i \stackrel{iid}{\sim} p(\theta)$ and $W_1^i = 1$, $i = 1, \dots, N$.
2. **Recursion.** For $n = 1, \dots, N_\phi$,
   1. **Correction.** Reweight the particles from stage $n-1$ by defining the incremental weights

      $$\tilde{w}_n^i = [p(Y|\theta_{n-1}^i)]^{\phi_n - \phi_{n-1}}$$

      and the normalized weights

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

      Then,

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

   2. **Selection.**
   3. **Mutation.**

SMC Algorithm

1. **Initialization.**
2. **Recursion.** For $n = 1, \ldots, N_\phi$,
   1. **Correction.**
   2. **Selection. (Optional Resampling)** Let $\{\hat{\theta}\}_{i=1}^N$ denote N *iid* draws from a multinomial distribution characterized by support points and weights $\{\theta_{n-1}^i, \tilde{W}_n^i\}_{i=1}^N$ and set $W_n^i = 1$. Then,

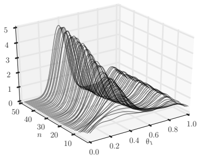
$$\hat{h}_{n,N} = \frac{1}{N} \sum_{i=1}^N W_n^i h(\hat{\theta}_n^i) \approx \mathbb{E}_{\pi_n}[h(\theta)].$$

   3. **Mutation.** Propagate the particles $\{\hat{\theta}_i, W_n^i\}$ via N_{MH} steps of a MH algorithm with transition density $\theta_n^i \sim K_n(\theta_n | \hat{\theta}_n^i; \zeta_n)$ and stationary distribution $\pi_n(\theta)$. Then,

$$\bar{h}_{n,N} = \frac{1}{N} \sum_{i=1}^N h(\theta_n^i) W_n^i \approx \mathbb{E}_{\pi_n}[h(\theta)].$$

Remarks

- Correction Step:
  - reweight particles from iteration $n-1$ to create importance sampling approximation of $\mathbb{E}_{\pi_n}[h(\theta)]$
- Selection Step: the resampling of the particles
  - (good) equalizes the particle weights and thereby increases accuracy of subsequent importance sampling approximations;
  - (not good) adds a bit of noise to the MC approximation.
- Mutation Step:
  - adapts particles to posterior $\pi_n(\theta)$;
  - imagine we don't do it: then we would be using draws from prior $p(\theta)$ to approximate posterior $\pi(\theta)$, which can't be good!



The Transition Kernel in Mutation Step

- Transition kernel $K_n(\theta|\hat{\theta}_n;\zeta_n)$: generated by running M steps of a Metropolis-Hastings algorithm.
- Suppose $M = 1$: Draw ϑ_t^i from

  $$q(\vartheta_t^i|\hat{\theta}_t^i);$$

  define the acceptance ratio can be expressed as

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

  and set $\theta_t^i = \vartheta_t^i$ with prob $\alpha(\cdot)$ and equal to $\hat{\theta}_t^i$ otherwise.
- Example:

  $$\vartheta|(\hat{\theta}_n^i, \Sigma_{n,b}^*) \sim N\left(\hat{\theta}_n^i, c_n^2\Sigma_n^*\right).$$

- Lessons from DSGE model MCMC:
  - blocking of parameters can reduces persistence of Markov chain;
  - mixture proposal density avoids "getting stuck."
- Adaptive choice of tuning parameters, e.g., $\zeta_n = \{c_n, \Sigma_n\}$.

The Selling Point: SMC and Multi-modal Posteriors

- In the small DSGE model, the shocks evolve according to:

$$\begin{aligned} \hat{z}_t &= \rho_z \hat{z}_{t-1} + \epsilon_{z,t}, \quad \epsilon_{z,t} \sim N(0, \sigma_z^2), \\ \hat{g}_t &= \rho_g \hat{g}_{t-1} + \epsilon_{g,t}, \quad \epsilon_{z,t} \sim N(0, \sigma_g^2). \end{aligned}$$

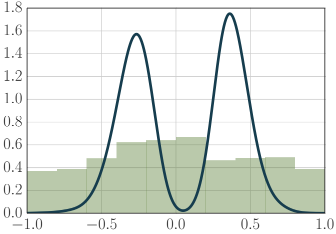
- Now suppose we generalize the shock structure:

$$\begin{aligned} \begin{bmatrix} \hat{z}_t \\ \hat{g}_t \end{bmatrix} &= \begin{bmatrix} \rho_z & \rho_{zg} \\ \rho_{gz} & \rho_g \end{bmatrix} \begin{bmatrix} \hat{z}_{t-1} \\ \hat{g}_{t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{z,t} \\ \epsilon_{g,t} \end{bmatrix}, \\ \begin{bmatrix} \epsilon_{z,t} \\ \epsilon_{g,t} \end{bmatrix} &\sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_z^2 & 0 \\ 0 & \sigma_g^2 \end{bmatrix} \right). \end{aligned}$$

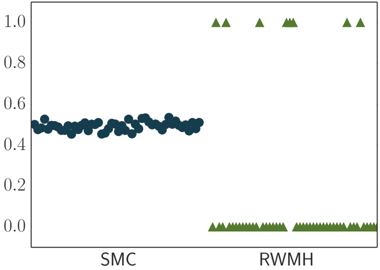
- Let's compare posterior computations based on standard RWMH and SMC.

The Selling Point: SMC and Multimodal Posterior

"True" Posterior ρ_{gz}



$\mathbb{P}_\pi\{\rho_{zg} > 0\}$



Theoretical Properties

- Goal: strong law of large numbers (SLLN) and central limit theorem (CLT) as $N \longrightarrow \infty$ for every iteration $n = 1, \ldots, N_\phi$.

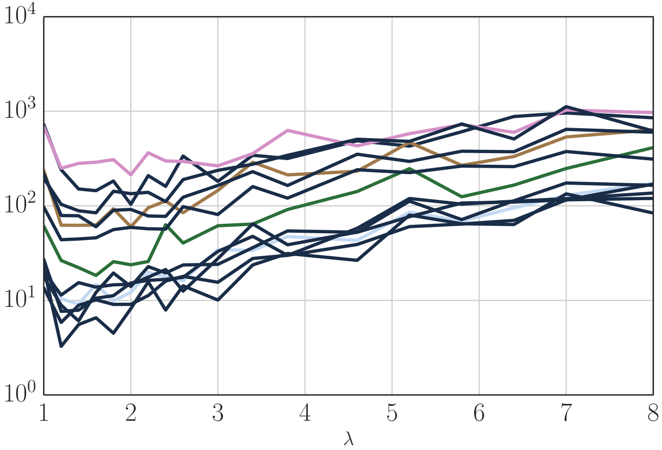
- Regularity conditions:
  - proper prior;
  - bounded likelihood function;
  - $2 + \delta$ posterior moments of $h(\theta)$.

- Idea of proof (Chopin, 2004):
  - SLLN and CLT can be proved recursively.
  - For step n assume that $n-1$ approximation (with normalized weights) yields

$$\sqrt{N}\left(\frac{1}{N}\sum_{i=1}^{N} h(\theta^i_{n-1})W^i_{n-1} - \mathbb{E}_{\pi_{n-1}}[h(\theta)]\right) \Longrightarrow N\big(0, \Omega_{n-1}(h)\big)$$

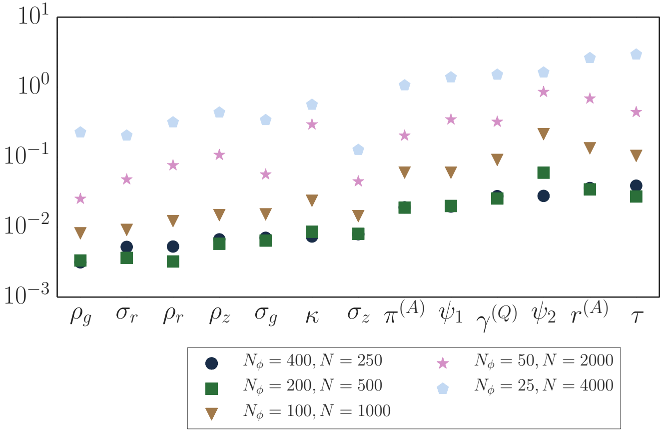
  - Initialization: SLLN and CLT for *iid* random variables because we sample from prior.
  - Then show that convergence also holds for stage n posterior.

Effect of λ on Inefficiency Factors $\text{InEff}_N[\bar{\theta}]$ (Small DSGE)



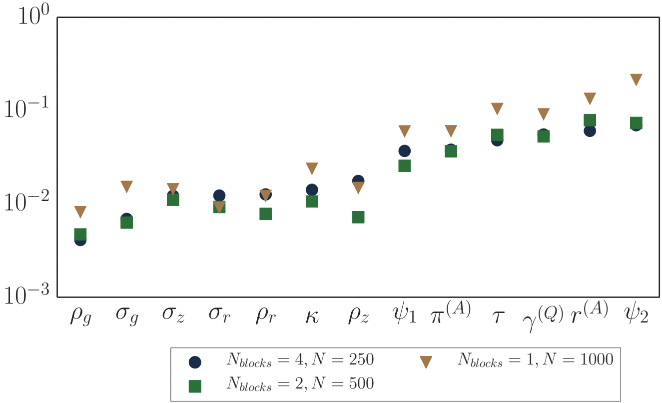
Notes: The figure depicts hairs of $\text{InEff}_N[\bar{\theta}]$ as function of λ. The inefficiency factors are computed based on $N_{run} = 50$ runs of the SMC algorithm. Each hair corresponds to a DSGE model parameter.

Number of Stages N_ϕ vs Number of Particles N (Small DSGE)



Notes: $\mathbb{V}[\bar{\theta}]/\mathbb{V}_\pi[\theta]$ for a specific configuration of the SMC algorithm. The inefficiency factors are computed based on $N_{run} = 50$ runs of the SMC algorithm. $N_{blocks} = 1$, $\lambda = 2$, $N_{MH} = 1$.

Number of blocks N_{blocks} in Mutation Step vs Number of Particles N (Small DSGE)



Notes: $\mathbb{V}[\bar{\theta}]/\mathbb{V}_{\pi}[\theta]$ for a specific configuration of the SMC algorithm. The inefficiency factors are computed based on $N_{run} = 50$ runs of the SMC algorithm. $N_{\phi} = 100$, $\lambda = 2$, $N_{MH} = 1$.

Part II: Approximating the Likelihood with a Particle Filter

- DSGE model takes the form of a state-space model with
  - Measurement equation: $p(y_t|s_t, \theta)$
  - State-transition equation: $p(s_t|s_{t-1}, \theta)$

- Evaluation of the likelihood function $p(Y_{1:T}|\theta)$ requires filter that integrates out the latent s_t's as a by-product:

$$
\begin{aligned}
p(y_t|Y_{1:t-1}) &= \int p(y_t|s_t)p(s_t|s_{t-1})p(s_{t-1}|Y_{1:t-1})ds_{t-1:t} \\
p(s_t|Y_{1:t}) &\propto p(y_t|s_t) \int p(s_t|s_{t-1})p(s_{t-1}|Y_{1:t-1})ds_{t-1}
\end{aligned}
$$

- Sequential Monte Carlo techniques (= particle filters) can be used to solve filtering problem.

- Particles $\{s_t^j, W_t^j\}_{j=1}^M$ approximate

$$
\bar{h}_{t,M} = \frac{1}{M}\sum_{j=1}^{M} h(s_t^j)W_t^j \approx \mathbb{E}[h(s_t)|Y_{1:t}, \theta].
$$

Generic Particle Filter

1. **Initialization.** Draw the initial particles from the distribution $s_0^j \stackrel{iid}{\sim} p(s_0)$ and set $W_0^j = 1$, $j = 1, \ldots, M$.
2. **Recursion.** For $t = 1, \ldots, T$:
   1. **Forecasting s_t.** Draw $\tilde{s}_t^j$ from density $g_t(\tilde{s}_t | s_{t-1}^j, \theta)$ and define

      $$\omega_t^j = \frac{p(\tilde{s}_t^j | s_{t-1}^j, \theta)}{g_t(\tilde{s}_t^j | s_{t-1}^j, \theta)}.$$

      Then,

      $$\hat{h}_{t,M} = \frac{1}{M} \sum_{j=1}^{M} h(\tilde{s}_t^j) \omega_t^j W_{t-1}^j \approx \mathbb{E}[h(s_t) | Y_{1:t-1}, \theta].$$

   2. **Forecasting y_t.** Define the incremental weights

      $$\tilde{w}_t^j = p(y_t | \tilde{s}_t^j, \theta) \omega_t^j.$$

      The predictive density $p(y_t | Y_{1:t-1}, \theta)$ can be approximated by

      $$\hat{p}(y_t | Y_{1:t-1}, \theta) = \frac{1}{M} \sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j.$$

   3. **Updating.**
   4. **Selection.**

Generic Particle Filter

1. **Initialization.**
2. **Recursion.** For $t = 1, \ldots, T$: ...
   - **Updating.** Define the normalized weights

     $$\tilde{W}_t^j = \frac{\tilde{w}_t^j W_{t-1}^j}{\frac{1}{M}\sum_{j=1}^M \tilde{w}_t^j W_{t-1}^j}.$$

     Then,

     $$\tilde{h}_{t,M} = \frac{1}{M}\sum_{j=1}^M h(\tilde{s}_t^j)\tilde{W}_t^j \approx \mathbb{E}[h(s_t)|Y_{1:t},\theta].$$

   - **Selection (Optional).** Resample the particles via multinomial resampling. Leads to $\{s_t^j, W_t^j = 1\}_{j=1}^M$. Then,

     $$\bar{h}_{t,M} = \frac{1}{M}\sum_{j=1}^M h(s_t^j)W_t^j \approx \mathbb{E}[h(s_t)|Y_{1:t},\theta].$$

3. **Likelihood Approximation.**

$$\ln \hat{p}(Y_{1:T}|\theta) = \sum_{t=1}^T \ln\left(\frac{1}{M}\sum_{j=1}^M \tilde{w}_t^j W_{t-1}^j\right).$$

Adapting the Generic PF

- Bootstrap particle filter:

$$g_t(\tilde{s}_t|s^j_{t-1}) = p(\tilde{s}_t|s^j_{t-1}).$$

Easy to implement, but often very inaccurate.

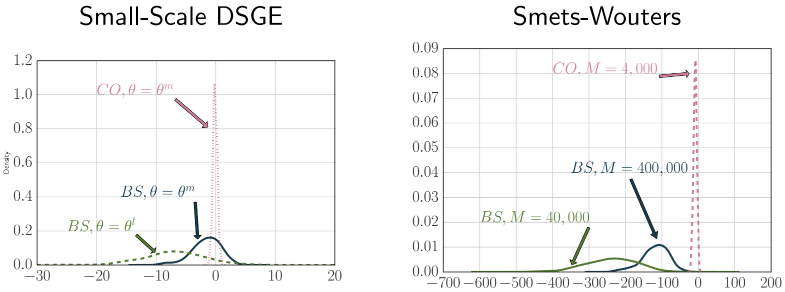
- Conditionally-optimal particle filter:

$$g_t(\tilde{s}_t|s^j_{t-1}) = p(\tilde{s}_t|y_t, s^j_{t-1}).$$

This is the posterior of s_t given s^j_{t-1}. Typically infeasible, but a good benchmark.

- Approximately conditionally-optimal filter: e.g., from linearized version of DSGE model or approximate nonlinear filters.

Distribution of Log-Likelihood Approximation Errors: $\ln \hat{p}(Y|\theta) - \ln p(Y|\theta)$



Notes: $N_{run} = 100$ for the bootstrap (BS) particle filter and conditionally-optimal (CO) particle filter.

Part III: 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: $\theta_0^i \stackrel{iid}{\sim} p(\theta)$ and $W_0^i = 1$, $i = 1, \ldots, N$.
2. **Recursion.** For $t = 1, \ldots, T$,
    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. **Selection.** (unchanged)
    3. **Mutation.**

SMC^2

1. **Initialization.**
2. **Recursion.** For $t = 1, \ldots, T$,
   1. **Correction.**
   2. **Selection.**
   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:

$$\int\int h(\theta, U_{1:t-1}) p(U_{1:t-1}) p(\theta|Y_{1:t-1}) dU_{1:t-1} d\theta$$

$$\approx \quad \frac{1}{N}\sum_{i=1}^N h(\theta^i_{t-1}, U^i_{1:t-1}) W^i_{t-1}.$$

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.

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.
- Our sequential (data-tempering) implementation of the SMC^2 algorithm suffers from particle degeneracy in the intial stages, i.e., for small sample sizes.

Conclusion

- We explored 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.