Particle Filtering

Frank Schorfheide
University of Pennsylvania
Gerzensee Ph.D. Course on Bayesian Macroeconometrics

May 28, 2019

From Linear to Nonlinear DSGE Models

- Linear DSGE model leads to

$$\begin{aligned} y_t &= \Psi_0(\theta) + \Psi_1(\theta)t + \Psi_2(\theta)s_t + u_t, \quad u_t \sim N(0, \Sigma_u), \\ s_t &= \Phi_1(\theta)s_{t-1} + \Phi_\epsilon(\theta)\epsilon_t, \quad \epsilon_t \sim N(0, \Sigma_\epsilon). \end{aligned}$$

- Nonlinear DSGE model leads to

$$\begin{aligned} y_t &= \Psi(s_t, t; \theta) + u_t, \quad u_t \sim F_u(\cdot; \theta) \\ s_t &= \Phi(s_{t-1}, \epsilon_t; \theta), \quad \epsilon_t \sim F_\epsilon(\cdot; \theta). \end{aligned}$$

From Linear to Nonlinear DSGE Models

- While DSGE models are inherently nonlinear, the nonlinearities are often small and decision rules are approximately linear.
- One can add certain features that generate more pronounced nonlinearities:
  - stochastic volatility;
  - markov switching coefficients;
  - asymmetric adjustment costs;
  - occasionally binding constraints.

Particle Filters

- There are many particle filters...
- We will focus on three types:
  - Bootstrap PF
  - A generic PF
  - A conditionally-optimal PF

Filtering - General Idea

- State-space representation of linearized DSGE model

$$
\begin{aligned}
y_t &= \Psi_0(\theta) + \Psi_1(\theta)t + \Psi_2(\theta)s_t(+u_t) \quad \text{measurement} \\
s_t &= \Phi_1(\theta)s_t + \Phi_\epsilon(\theta)\epsilon_t \quad \text{state transition}
\end{aligned}
$$

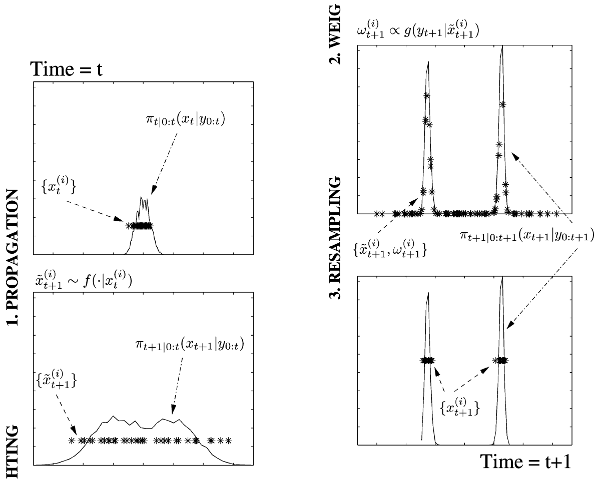
- Likelihood function:

$$
p(Y_{1:T}|\theta) = \prod_{t=1}^{T} p(y_t|Y_{1:t-1},\theta)
$$

- A filter generates a sequence of conditional distributions $s_t|Y_{1:t}$.
- Iterations:
  - Initialization at time $t-1$: $p(s_{t-1}|Y_{1:t-1},\theta)$
  - Forecasting t given $t-1$:
    1. Transition equation: $p(s_t|Y_{1:t-1},\theta) = \int p(s_t|s_{t-1},Y_{1:t-1},\theta)p(s_{t-1}|Y_{1:t-1},\theta)ds_{t-1}$
    2. Measurement equation: $p(y_t|Y_{1:t-1},\theta) = \int p(y_t|s_t,Y_{1:t-1},\theta)p(s_t|Y_{1:t-1},\theta)ds_t$
  - Updating with Bayes theorem. Once y_t becomes available:

$$
p(s_t|Y_{1:t},\theta) = p(s_t|y_t,Y_{1:t-1},\theta) = \frac{p(y_t|s_t,Y_{1:t-1},\theta)p(s_t|Y_{1:t-1},\theta)}{p(y_t|Y_{1:t-1},\theta)}
$$

Bootstrap Particle Filter – Idea



Bootstrap Particle Filter

1. **Initialization.** Draw the initial particles from the distribution $s_0^j \stackrel{iid}{\sim} p(s_0)$ and set $W_0^j = 1$, $j = 1, \dots, M$.
2. **Recursion.** For $t = 1, \dots, T$:
   1. **Forecasting s_t.** Propagate the period $t-1$ particles $\{s_{t-1}^j, W_{t-1}^j\}$ by iterating the state-transition equation forward:

$$\tilde{s}_t^j = \Phi(s_{t-1}^j, \epsilon_t^j; \theta), \quad \epsilon_t^j \sim F_\epsilon(\cdot; \theta). \tag{1}$$

An approximation of $\mathbb{E}[h(s_t)|Y_{1:t-1}, \theta]$ is given by

$$\hat{h}_{t,M} = \frac{1}{M} \sum_{j=1}^{M} h(\tilde{s}_t^j) W_{t-1}^j. \tag{2}$$

Bootstrap Particle Filter

1. **Initialization.**
2. **Recursion.** For $t = 1, \ldots, T$:
   1. **Forecasting s_t.**
   2. **Forecasting y_t.** Define the incremental weights

$$\tilde{w}_t^j = p(y_t|\tilde{s}_t^j, \theta). \tag{3}$$

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

$$\hat{p}(y_t|Y_{1:t-1}, \theta) = \frac{1}{M} \sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j. \tag{4}$$

If the measurement errors are $N(0, \Sigma_u)$ then the incremental weights take the form

$$\tilde{w}_t^j = (2\pi)^{-n/2} |\Sigma_u|^{-1/2} \exp\left\{ -\frac{1}{2} \big(y_t - \Psi(\tilde{s}_t^j, t; \theta)\big)' \Sigma_u^{-1} \big(y_t - \Psi(\tilde{s}_t^j, t; \theta)\big) \right\}, \tag{5}$$

where n here denotes the dimension of y_t.

Bootstrap Particle Filter

1. **Initialization.**
2. **Recursion.** For $t = 1, \dots, T$:
   1. **Forecasting s_t.**
   2. **Forecasting y_t.** Define the incremental weights

$$\tilde{w}_t^j = p(y_t|\tilde{s}_t^j, \theta). \tag{6}$$

   3. **Updating.** Define the normalized weights

$$\tilde{W}_t^j = \frac{\tilde{w}_t^j W_{t-1}^j}{\frac{1}{M}\sum_{j=1}^M \tilde{w}_t^j W_{t-1}^j}. \tag{7}$$

   An approximation of $\mathbb{E}[h(s_t)|Y_{1:t}, \theta]$ is given by

$$\tilde{h}_{t,M} = \frac{1}{M}\sum_{j=1}^M h(\tilde{s}_t^j)\tilde{W}_t^j. \tag{8}$$

Bootstrap Particle Filter

1. **Initialization.**
2. **Recursion.** For $t = 1, \ldots, T$:
   1. **Forecasting s_t.**
   2. **Forecasting y_t.**
   3. **Updating.**
   4. **Selection (Optional).** Resample the particles via multinomial resampling. Let $\{s_t^j\}_{j=1}^M$ denote M iid draws from a multinomial distribution characterized by support points and weights $\{\tilde{s}_t^j, \tilde{W}_t^j\}$ and set $W_t^j = 1$ for $j =, 1 \ldots, M$.
      An approximation of $\mathbb{E}[h(s_t)|Y_{1:t}, \theta]$ is given by

$$\bar{h}_{t,M} = \frac{1}{M} \sum_{j=1}^{M} h(s_t^j) W_t^j. \tag{9}$$

3. **Likelihood Approximation.** The approximation of the log likelihood function is given by

$$\ln \hat{p}(Y_{1:T}|\theta) = \sum_{t=1}^{T} \ln \left(\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j \right). \tag{10}$$

Asymptotics

- The convergence results can be established recursively, starting from the assumption

$$
\begin{aligned}
\bar{h}_{t-1,M} &\stackrel{a.s.}{\longrightarrow} \mathbb{E}[h(s_{t-1})|Y_{1:t-1}], \\
\sqrt{M}\big(\bar{h}_{t-1,M} - \mathbb{E}[h(s_{t-1})|Y_{1:t-1}]\big) &\Longrightarrow N\big(0, \Omega_{t-1}(h)\big).
\end{aligned}
$$

- Forward iteration: draw s_t from $g_t(s_t|s^j_{t-1}) = p(s_t|s^j_{t-1})$.
- Decompose

$$
\begin{aligned}
&\hat{h}_{t,M} - \mathbb{E}[h(s_t)|Y_{1:t-1}] \\
&= \frac{1}{M}\sum_{j=1}^{M}\left(h(\tilde{s}^j_t) - \mathbb{E}_{p(\cdot|s^j_{t-1})}[h]\right) W^j_{t-1} \\
&\quad + \frac{1}{M}\sum_{j=1}^{M}\left(\mathbb{E}_{p(\cdot|s^j_{t-1})}[h] W^j_{t-1} - \mathbb{E}[h(s_t)|Y_{1:t-1}]\right) \\
&= I + II,
\end{aligned}
\tag{11}
$$

- Both I and II converge to zero (and potentially satisfy CLT).

Asymptotics

- Updating step approximates

$$\mathbb{E}[h(s_t)|Y_{1:t}] = \frac{\int h(s_t)p(y_t|s_t)p(s_t|Y_{1:t-1})ds_t}{\int p(y_t|s_t)p(s_t|Y_{1:t-1})ds_t} \approx \frac{\frac{1}{M}\sum_{j=1}^{M} h(\tilde{s}_t^j)\tilde{w}_t^j W_{t-1}^j}{\frac{1}{M}\sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j} \tag{12}$$

- Define the normalized incremental weights as

$$v_t(s_t) = \frac{p(y_t|s_t)}{\int p(y_t|s_t)p(s_t|Y_{1:t-1})ds_t}. \tag{13}$$

- Under suitable regularity conditions, the Monte Carlo approximation satisfies a CLT of the form

$$\sqrt{M}\left(\tilde{h}_{t,M} - \mathbb{E}[h(s_t)|Y_{1:t}]\right) \tag{14}$$
$$\Longrightarrow \quad N\big(0, \tilde{\Omega}_t(h)\big), \quad \tilde{\Omega}_t(h) = \hat{\Omega}_t\big(v_t(s_t)(h(s_t) - \mathbb{E}[h(s_t)|Y_{1:t}])\big).$$

- Distribution of particle weights matters for accuracy! $\Longrightarrow$ Resampling!

The Role of Measurement Errors

- Measurement errors may not be intrinsic to DSGE model.
- Bootstrap filter needs non-degenerate $p(y_t|s_t,\theta)$ for incremental weights to be well defined.
- Decreasing the measurement error variance Σ_u, holding everything else fixed, increases the variance of the particle weights, and reduces the accuracy of Monte Carlo approximation.

Generic Particle Filter

1. **Initialization.** Same as BS PF
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

$$\omega_t^j = \frac{p(\tilde{s}_t^j | s_{t-1}^j, \theta)}{g_t(\tilde{s}_t^j | s_{t-1}^j, \theta)}. \tag{15}$$

      An approximation of $\mathbb{E}[h(s_t) | Y_{1:t-1}, \theta]$ is given by

$$\hat{h}_{t,M} = \frac{1}{M} \sum_{j=1}^{M} h(\tilde{s}_t^j) \omega_t^j W_{t-1}^j. \tag{16}$$

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

$$\tilde{w}_t^j = p(y_t | \tilde{s}_t^j, \theta) \omega_t^j. \tag{17}$$

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

$$\hat{p}(y_t | Y_{1:t-1}, \theta) = \frac{1}{M} \sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j. \tag{18}$$

   3. **Updating.** Same as BS PF
   4. **Selection.** Same as BS PF

Adapting the Generic PF

- Conditionally-optimal importance distribution:

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

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

- Approximately conditionally-optimal distributions: from linearize version of DSGE model or approximate nonlinear filters.

- Conditionally-linear models: do Kalman filter updating on a subvector of s_t. Example:

$$\begin{aligned} y_t &= \Psi_0(m_t) + \Psi_1(m_t)t + \Psi_2(m_t)s_t + u_t, \quad u_t \sim N(0, \Sigma_u), \\ s_t &= \Phi_0(m_t) + \Phi_1(m_t)s_{t-1} + \Phi_\epsilon(m_t)\epsilon_t, \quad \epsilon_t \sim N(0, \Sigma_\epsilon), \end{aligned}$$

  where m_t follows a discrete Markov-switching process.

More on Conditionally-Linear Models

- State-space representation is linear conditional on m_t.
- Write

$$p(m_t, s_t | Y_{1:t}) = p(m_t | Y_{1:t}) p(s_t | m_t, Y_{1:t}), \tag{19}$$

where

$$s_t | (m_t, Y_{1:t}) \sim N\big(\bar{s}_{t|t}(m_t), P_{t|t}(m_t)\big). \tag{20}$$

- Vector of means $\bar{s}_{t|t}(m_t)$ and the covariance matrix $P_{t|t}(m)_t$ are sufficient statistics for the conditional distribution of s_t.
- Approximate $(m_t, s_t)|Y_{1:t}$ by $\{m_t^j, \bar{s}_{t|t}^j, P_{t|t}^j, W_t^j\}_{i=1}^N$.
- The swarm of particles approximates

$$\int h(m_t, s_t) p(m_t, s_t, Y_{1:t}) d(m_t, s_t) \tag{21}$$

$$= \int \left[\int h(m_t, s_t) p(s_t | m_t, Y_{1:t}) ds_t \right] p(m_t | Y_{1:t}) dm_t$$

$$\approx \frac{1}{M} \sum_{j=1}^{M} \left[\int h(m_t^j, s_t^j) p_N\big(s_t | \bar{s}_{t|t}^j, P_{t|t}^j\big) ds_t \right] W_t^j.$$

More on Conditionally-Linear Models

- We used Rao-Blackwellization to reduce variance:

$$\begin{aligned} \mathbb{V}[h(s_t, m_t)] &= \mathbb{E}\big[\mathbb{V}[h(s_t, m_t)|m_t]\big] + \mathbb{V}\big[\mathbb{E}[h(s_t, m_t)|m_t]\big] \\ &ge \mathbb{V}\big[\mathbb{E}[h(s_t, m_t)|m_t]\big] \end{aligned}$$

- To forecast the states in period t, generate $\tilde{m}_t^j$ from $g_t(\tilde{m}_t|m_{t-1}^j)$ and define:

$$\omega_t^j = \frac{p(\tilde{m}_t^j|m_{t-1}^j)}{g_t(\tilde{m}_t^j|m_{t-1}^j)}. \quad (22)$$

- The Kalman filter forecasting step can be used to compute:

$$\begin{aligned} \tilde{s}_{t|t-1}^j &= \Phi_0(\tilde{m}_t^j) + \Phi_1(\tilde{m}_t^j)s_{t-1}^j \\ P_{t|t-1}^j &= \Phi_\epsilon(\tilde{m}_t^j)\Sigma_\epsilon(\tilde{m}_t^j)\Phi_\epsilon(\tilde{m}_t^j)' \\ \tilde{y}_{t|t-1}^j &= \Psi_0(\tilde{m}_t^j) + \Psi_1(\tilde{m}_t^j)t + \Psi_2(\tilde{m}_t^j)\tilde{s}_{t|t-1}^j \\ F_{t|t-1}^j &= \Psi_2(\tilde{m}_t^j)P_{t|t-1}^j\Psi_2(\tilde{m}_t^j)' + \Sigma_u. \end{aligned} \quad (23)$$

More on Conditionally-Linear Models

- Then,

$$\int h(m_t, s_t) p(m_t, s_t | Y_{1:t-1}) d(m_t, s_t) \tag{24}$$

$$= \int \left[\int h(m_t, s_t) p(s_t | m_t, Y_{1:t-1}) ds_t \right] p(m_t | Y_{1:t-1}) dm_t$$

$$\approx \frac{1}{M} \sum_{j=1}^{M} \left[\int h(m_t^j, s_t^j) p_N\big(s_t | \tilde{s}_{t|t-1}^j, P_{t|t-1}^j\big) ds_t \right] \omega_t^j W_{t-1}^j$$

- The likelihood approximation is based on the incremental weights

$$\tilde{w}_t^j = p_N\big(y_t | \tilde{y}_{t|t-1}^j, F_{t|t-1}^j\big) \omega_t^j. \tag{25}$$

- Conditional on $\tilde{m}_t^j$ we can use the Kalman filter once more to update the information about s_t in view of the current observation y_t:

$$\begin{aligned} \tilde{s}_{t|t}^j &= \tilde{s}_{t|t-1}^j + P_{t|t-1}^j \Psi_2(\tilde{m}_t^j)' \big(F_{t|t-1}^j\big)^{-1} (y_t - \bar{y}_{t|t-1}^j) \\ \tilde{P}_{t|t}^j &= P_{t|t-1}^j - P_{t|t-1}^j \Psi_2(\tilde{m}_t^j)' \big(F_{t|t-1}^j\big)^{-1} \Psi_2(\tilde{m}_t^j) P_{t|t-1}^j. \end{aligned} \tag{26}$$

Particle Filter For Conditionally Linear Models

1. **Initialization.**
2. **Recursion.** For $t = 1, \ldots, T$:
   1. **Forecasting s_t.** Draw $\tilde{m}_t^j$ from density $g_t(\tilde{m}_t | m_{t-1}^j, \theta)$, calculate the importance weights ω_t^j in (22), and compute $\tilde{s}_{t|t-1}^j$ and $P_{t|t-1}^j$ according to (23). An approximation of $\mathbb{E}[h(s_t, m_t)|Y_{1:t-1}, \theta]$ is given by (25).
   2. **Forecasting y_t.** Compute the incremental weights $\tilde{w}_t^j$ according to (25). Approximate the predictive density $p(y_t|Y_{1:t-1}, \theta)$ by

$$\hat{p}(y_t|Y_{1:t-1}, \theta) = \frac{1}{M} \sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j. \tag{27}$$

   3. **Updating.** Define the normalized weights

$$\tilde{W}_t^j = \frac{\tilde{w}_t^j W_{t-1}^j}{\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j} \tag{28}$$

   and compute $\tilde{s}_{t|t}^j$ and $\tilde{P}_{t|t}^j$ according to (26). An approximation of $\mathbb{E}[h(m_t, s_t)|Y_{1:t}, \theta]$ can be obtained from $\{\tilde{m}_t^j, \tilde{s}_{t|t}^j, \tilde{P}_{t|t}^j, \tilde{W}_t^j\}$.
   4. **Selection.**
3. **Likelihood Approximation.**

Nonlinear and Partially Deterministic State Transitions

- Example:

$$s_{1,t} = \Phi_1(s_{t-1}, \epsilon_t), \quad s_{2,t} = \Phi_2(s_{t-1}), \quad \epsilon_t \sim N(0,1).$$

- Generic filter requires evaluation of $p(s_t|s_{t-1})$.
- Define $\varsigma_t = [s_t', \epsilon_t']'$ and add identity $\epsilon_t = \epsilon_t$ to state transition.
- Factorize the density $p(\varsigma_t|\varsigma_{t-1})$ as

$$p(\varsigma_t|\varsigma_{t-1}) = p^\epsilon(\epsilon_t)p(s_{1,t}|s_{t-1}, \epsilon_t)p(s_{2,t}|s_{t-1}).$$

  where $p(s_{1,t}|s_{t-1}, \epsilon_t)$ and $p(s_{2,t}|s_{t-1})$ are pointmasses.
- Sample innovation ϵ_t from $g_t^\epsilon(\epsilon_t|s_{t-1})$.
- Then

$$\omega_t^j = \frac{p(\tilde{\varsigma}_t^j|\varsigma_{t-1}^j)}{g_t(\tilde{\varsigma}_t^j|\varsigma_{t-1}^j)} = \frac{p^\epsilon(\tilde{\epsilon}_t^j)p(\tilde{s}_{1,t}^j|s_{t-1}^j, \tilde{\epsilon}_t^j)p(\tilde{s}_{2,t}^j|s_{t-1}^j)}{g_t^\epsilon(\tilde{\epsilon}_t^j|s_{t-1}^j)p(\tilde{s}_{1,t}^j|s_{t-1}^j, \tilde{\epsilon}_t^j)p(\tilde{s}_{2,t}^j|s_{t-1}^j)} = \frac{p^\epsilon(\tilde{\epsilon}_t^j)}{g_t^\epsilon(\tilde{\epsilon}_t^j|s_{t-1}^j)}.$$

Degenerate Measurement Error Distributions

- Our discussion of the conditionally-optimal importance distribution suggests that in the absence of measurement errors, one has to solve the system of equations

$$y_t = \Psi\big(\Phi(s^j_{t-1}, \tilde{\epsilon}^j_t)\big),$$

to determine $\tilde{\epsilon}^j_t$ as a function of s^j_{t-1} and the current observation y_t.

- Then define

$$\omega^j_t = p^\epsilon(\tilde{\epsilon}^j_t) \quad \text{and} \quad \tilde{s}^j_t = \Phi(s^j_{t-1}, \tilde{\epsilon}^j_t).$$

- Difficulty: one has to find all solutions to a nonlinear system of equations.

- While resampling duplicates particles, the duplicated particles do not mutate, which can lead to a degeneracy.

Next Steps

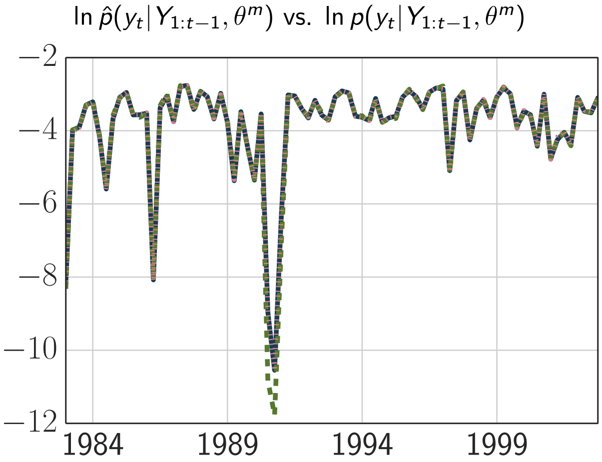
- We will now apply PFs to linearized DSGE models.
- This allows us to compare the Monte Carlo approximation to the "truth."
- Small-scale New Keynesian DSGE model
- Smets-Wouters model

Illustration 1: Small-Scale DSGE Model

Parameter Values For Likelihood Evaluation

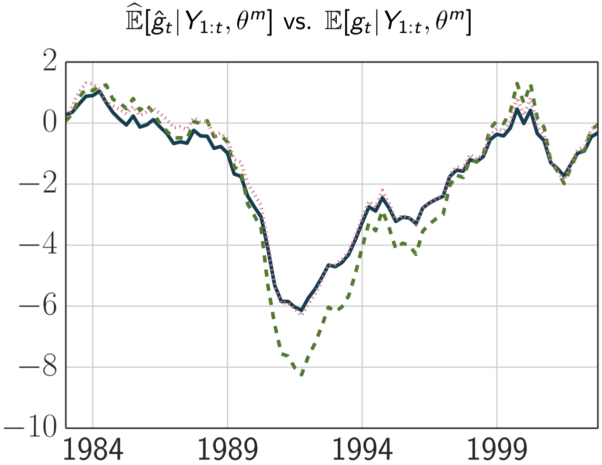
Parameter	θ^m	θ^l	Parameter	θ^m	θ^l
τ	2.09	3.26	κ	0.98	0.89
ψ_1	2.25	1.88	ψ_2	0.65	0.53
ρ_r	0.81	0.76	ρ_g	0.98	0.98
ρ_z	0.93	0.89	$r^{(A)}$	0.34	0.19
$\pi^{(A)}$	3.16	3.29	$\gamma^{(Q)}$	0.51	0.73
σ_r	0.19	0.20	σ_g	0.65	0.58
σ_z	0.24	0.29	$\ln p(Y\|\theta)$	-306.5	-313.4

Likelihood Approximation



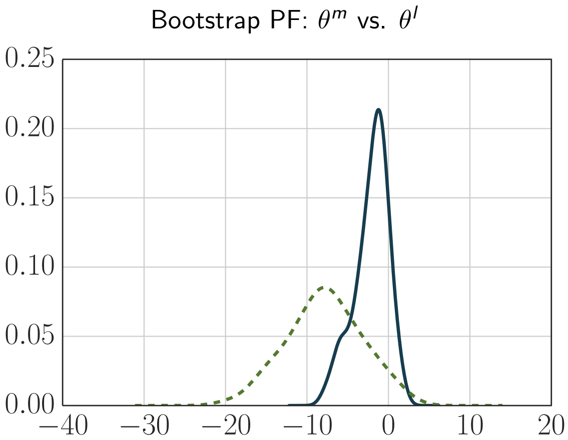
Notes: The results depicted in the figure are based on a single run of the bootstrap PF (dashed), the conditionally-optimal PF (dotted), and the Kalman filter (solid).

Filtered State



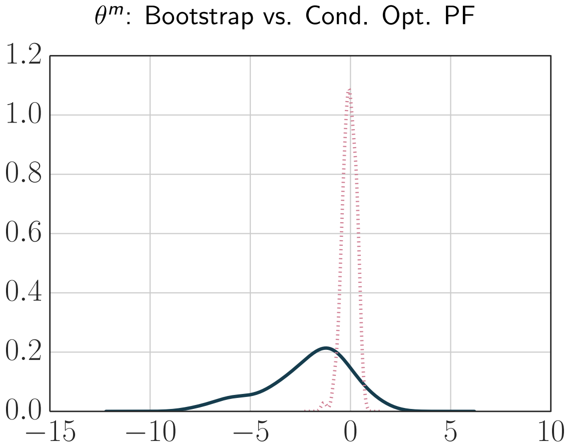
Notes: The results depicted in the figure are based on a single run of the bootstrap PF (dashed), the conditionally-optimal PF (dotted), and the Kalman filter (solid).

Distribution of Log-Likelihood Approximation Errors



Notes: Density estimate of $\hat{\Delta}_1 = \ln \hat{p}(Y_{1:T}|\theta) - \ln p(Y_{1:T}|\theta)$ based on $N_{run} = 100$ runs of the PF. Solid line is $\theta = \theta^m$; dashed line is $\theta = \theta^l$ ($M = 40,000$).

Distribution of Log-Likelihood Approximation Errors



Notes: Density estimate of $\hat{\Delta}_1 = \ln \hat{p}(Y_{1:T}|\theta) - \ln p(Y_{1:T}|\theta)$ based on $N_{run} = 100$ runs of the PF. Solid line is bootstrap particle filter ($M = 40,000$); dotted line is conditionally optimal particle filter ($M = 400$).

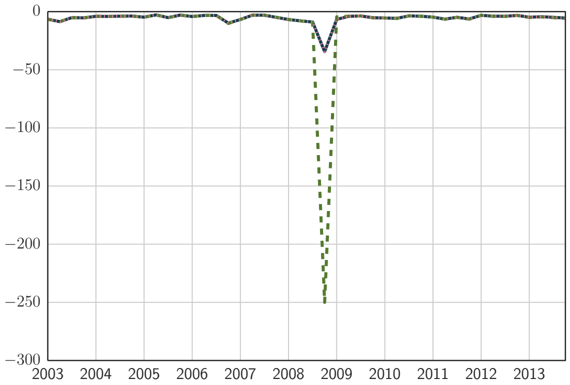
Summary Statistics for Particle Filters

	Bootstrap	Cond. Opt.	Auxiliary
Number of Particles M	40,000	400	40,000
Number of Repetitions	100	100	100
High Posterior Density: $\theta = \theta^m$			
Bias $\hat{\Delta}_1$	-1.39	-0.10	-2.83
StdD $\hat{\Delta}_1$	2.03	0.37	1.87
Bias $\hat{\Delta}_2$	0.32	-0.03	-0.74
Low Posterior Density: $\theta = \theta^l$			
Bias $\hat{\Delta}_1$	-7.01	-0.11	-6.44
StdD $\hat{\Delta}_1$	4.68	0.44	4.19
Bias $\hat{\Delta}_2$	-0.70	-0.02	-0.50

Notes: $\hat{\Delta}_1 = \ln \hat{p}(Y_{1:T}|\theta) - \ln p(Y_{1:T}|\theta)$ and $\hat{\Delta}_2 = \exp[\ln \hat{p}(Y_{1:T}|\theta) - \ln p(Y_{1:T}|\theta)] - 1$. Results are based on $N_{run} = 100$ runs of the particle filters.

Great Recession and Beyond

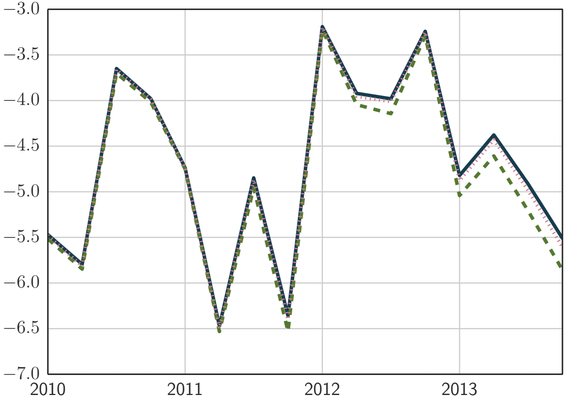
Mean of Log-likelihood Increments $\ln \hat{p}(y_t | Y_{1:t-1}, \theta^m)$



Notes: Solid lines represent results from Kalman filter. Dashed lines correspond to bootstrap particle filter ($M = 40,000$) and dotted lines correspond to conditionally-optimal particle filter ($M = 400$). Results are based on $N_{run} = 100$ runs of the filters.

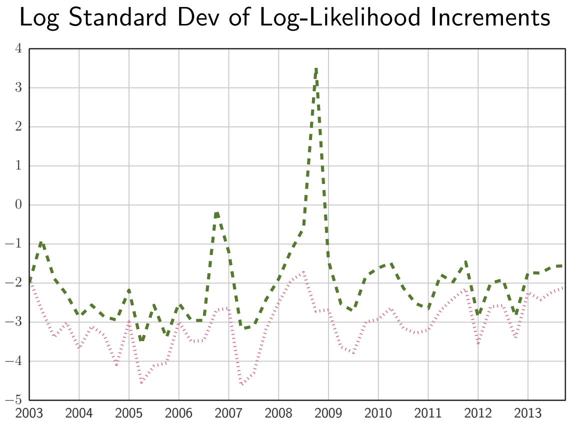
Great Recession and Beyond

Mean of Log-likelihood Increments $\ln \hat{p}(y_t | Y_{1:t-1}, \theta^m)$



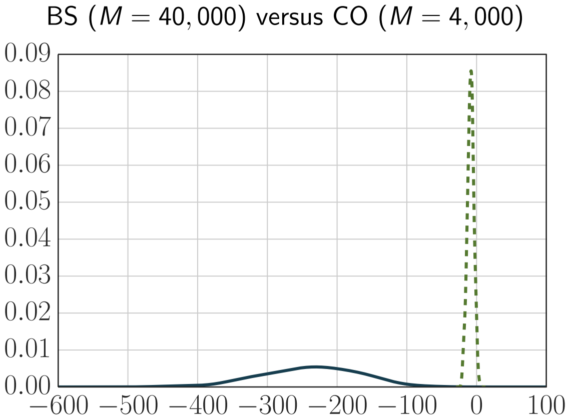
Notes: Solid lines represent results from Kalman filter. Dashed lines correspond to bootstrap particle filter ($M = 40,000$) and dotted lines correspond to conditionally-optimal particle filter ($M = 400$). Results are based on $N_{run} = 100$ runs of the filters.

Great Recession and Beyond



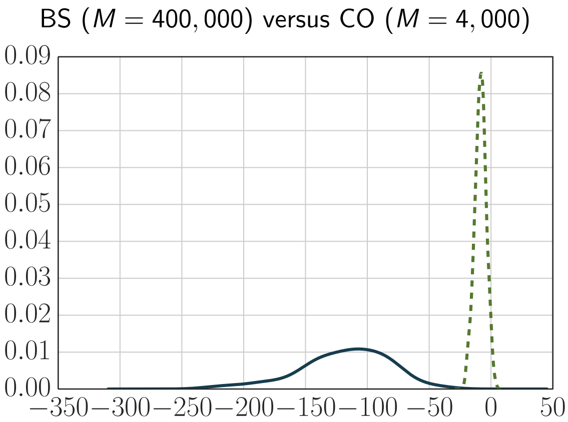
Notes: Solid lines represent results from Kalman filter. Dashed lines correspond to bootstrap particle filter ($M = 40,000$) and dotted lines correspond to conditionally-optimal particle filter ($M = 400$). Results are based on $N_{run} = 100$ runs of the filters.

SW Model: Distr. of Log-Likelihood Approximation Errors



Notes: Density estimates of $\hat{\Delta}_1 = \ln \hat{p}(Y|\theta) - \ln p(Y|\theta)$ based on $N_{run} = 100$. Solid densities summarize results for the bootstrap (BS) particle filter; dashed densities summarize results for the conditionally-optimal (CO) particle filter.

SW Model: Distr. of Log-Likelihood Approximation Errors



Notes: Density estimates of $\hat{\Delta}_1 = \ln \hat{p}(Y|\theta) - \ln p(Y|\theta)$ based on $N_{run} = 100$. Solid densities summarize results for the bootstrap (BS) particle filter; dashed densities summarize results for the conditionally-optimal (CO) particle filter.

SW Model: Summary Statistics for Particle Filters

	Bootstrap		Cond. Opt.	
Number of Particles M	40,000	400,000	4,000	40,000
Number of Repetitions	100	100	100	100
High Posterior Density: $\theta = \theta^m$				
Bias $\hat{\Delta}_1$	-238.49	-118.20	-8.55	-2.88
StdD $\hat{\Delta}_1$	68.28	35.69	4.43	2.49
Bias $\hat{\Delta}_2$	-1.00	-1.00	-0.87	-0.41
Low Posterior Density: $\theta = \theta^l$				
Bias $\hat{\Delta}_1$	-253.89	-128.13	-11.48	-4.91
StdD $\hat{\Delta}_1$	65.57	41.25	4.98	2.75
Bias $\hat{\Delta}_2$	-1.00	-1.00	-0.97	-0.64

Notes: Results are based on $N_{run} = 100$.

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)
  - SMC^2

- 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 (Andrieu, Docet, and Holenstein, 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

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

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

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

- The PFMH algorithm samples from the joint posterior

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

  and discards the draws of $\left(U_{1:T}\right)$.

- For this procedure to be valid, it needs to be the case that PF approximation is unbiased:

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

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:

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

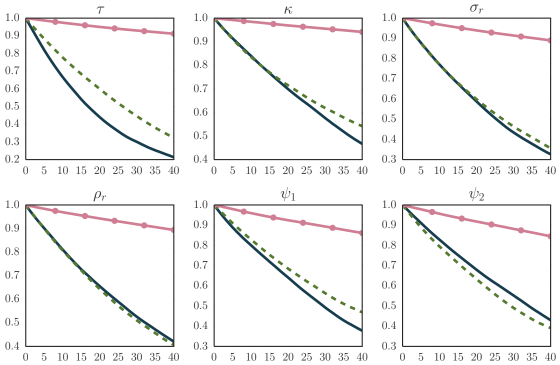
Small-Scale DSGE: Accuracy of MH Approximations

- 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.

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

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).

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
$\ln \hat{p}(Y)$	-964.44	-1017.94			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

- Start from SMC algorithm...
- Data tempering instead of likelihood tempering: $\pi_n^D(\theta) = p(\theta|Y_{1:t_n})$,
- Particle filter can deliver an unbiased estimate of the incremental weight $p(Y_{t_{n-1}+1:t_n}|\theta)$.
- Evaluate PF approximation of likelihood instead of true likelihood in the correction and mutation steps of SMC algorithm.
- Write:

$$\begin{aligned} \hat{p}(y_{t_{n-1}+1:t_n}|Y_{1:t_{n-1}},\theta) &= g(y_{t_{n-1}+1:t_n}|Y_{1:t_{n-1}},\theta,U_{1:t_n}) \\ \hat{p}(Y_{1:t_n}|\theta_n) &= g(Y_{1:t_n}|\theta_n,U_{1:t_n}). \end{aligned}$$

- $U_{1:t_n}$ is an array of *iid* uniform random variables generated by the particle filter with density $p(U_{1:t_n})$. Likelihood increments depend on entire $U_{1:t_n}$. Factorization:

$$p(U_{1:t_n}) = p(U_{1:t_1})p(U_{t_1+1:t_2})\cdots p(U_{t_{n-1}+1:t_n}).$$

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)	$(s_{t_n}^{N,1}, \mathcal{W}_{t_n}^{N,1})$	$(s_{t_n}^{N,2}, \mathcal{W}_{t_n}^{N,2})$	$\cdots$	$(s_{t_n}^{N,M}, \mathcal{W}_{t_n}^{N,M})$

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

$$\tilde{w}_t^i = \hat{p}(y_t | Y_{1:t-1}, \theta_{t-1}^i) = g(y_t | Y_{1:t-1}, \theta_{t-1}^i, U_{1:t}^i) \tag{29}$$

and the normalized weights

$$\tilde{W}_t^i = \frac{\tilde{w}_n^i W_{t-1}^i}{\frac{1}{N}\sum_{i=1}^N \tilde{w}_t^i W_{t-1}^i}, \quad i = 1, \ldots, N. \tag{30}$$

An approximation of $\mathbb{E}_{\pi_t}[h(\theta)]$ is given by

$$\tilde{h}_{t,N} = \frac{1}{N}\sum_{i=1}^N \tilde{W}_t^i h(\theta_{t-1}^i). \tag{31}$$

SMC^2

1. **Initialization.**
2. **Recursion.** For $t = 1, \ldots, T$,
   1. **Correction.**
   2. **Selection.** Resample the particles via multinomial resampling. Let $\{\hat{\theta}_t^i\}_{i=1}^M$ denote M *iid* draws from a multinomial distribution characterized by support points and weights $\{\theta_{t-1}^i, \tilde{W}_t^i\}_{j=1}^M$ and set $W_t^i = 1$. Define the vector of ancestors $\mathcal{A}_t$ with elements $\mathcal{A}_t^i$ by setting $\mathcal{A}_t^i = k$ if the ancestor of resampled particle i is particle k, that is, $\hat{\theta}_t^i = \theta_{t-1}^k$.
   An approximation of $\mathbb{E}_{\pi_t}[h(\theta)]$ is given by

$$\hat{h}_{t,N} = \frac{1}{N} \sum_{j=1}^{N} W_t^i h(\hat{\theta}_t^i). \tag{32}$$

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

$$q(\vartheta_t^i|\hat{\theta}_t^i)p(U_{1:t}^{*i}) \tag{33}$$

and the acceptance ratio can be expressed as

$$\alpha(\vartheta_t^i|\hat{\theta}_t^i) = \min \left\{ 1, \frac{\hat{p}(Y_{1:t}|\vartheta_t^i)p(\vartheta_t^i)/q(\vartheta_t^i|\hat{\theta}_t^i)}{\hat{p}(Y_{1:t}|\hat{\theta}_t^i)p(\hat{\theta}_t^i)/q(\hat{\theta}_t^i|\vartheta_t^i)} \right\}. \tag{34}$$

An approximation of $\mathbb{E}_{\pi_t}[h(\theta)]$ is given by

$$\bar{h}_{t,N} = \frac{1}{N}\sum_{i=1}^{N} h(\theta_t^i)W_t^i. \tag{35}$$

3. Approximation of $\mathbb{E}_{\pi}[h(\theta)]$ is given by $\bar{h}_{T,N} = \sum_{i=1}^{N} h(\theta_T^i)W_T^i$.

Why Does SMC^2 Work?

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

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

Correction Step

- The particle filter approximation of the likelihood increment can be written as

$$\hat{p}(y_t|Y_{1:t-1}, \theta^i_{t-1}) = g(y_t|Y_{1:t-1}, U^i_{1:t}, \theta^i_{t-1}).$$

- The value of the likelihood function for $Y_{1:t}$ can be tracked recursively as follows:

$$\begin{aligned} \hat{p}(Y_{1:t}|\theta^i_{t-1}) &= \hat{p}(y_t|Y_{1:t-1}, \theta^i_{t-1})\hat{p}(Y_{1:t-1}|\theta^i_{t-1}) \\ &= g(y_t|Y_{1:t}, U^i_{1:t}, \theta^i_{t-1})g(Y_{1:t-1}|U^i_{1:t-1}, \theta^i_{t-1}) \\ &= g(Y_{1:t}|U^i_{1:t}, \theta^i_{t-1}). \end{aligned} \quad (36)$$

  The last equality follows because conditioning $g(Y_{1:t-1}|U^i_{1:t-1}, \theta^i_{t-1})$ also on U_t does not change the particle filter approximation of the likelihood function for $Y_{1:t-1}$.

Correction Step

- 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

$$\int\int h(\theta)g(y_t|Y_{1:t-1},U_{1:t},\theta)p(U_{1:t})p(\theta|Y_{1:t-1})dU_{1:t}d\theta$$
$$= \int h(\theta)\left[\int g(y_t|Y_{1:t-1},U_{1:t},\theta)p(U_{1:t})dU_{1:t}\right]p(\theta|Y_{1:t-1})d\theta.$$

- 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 $\{s_t^{i,j}, W_t^{i,j}\}$, and the set of random numbers $U_{1:t}^i$.
- In the implementation, the likelihood values are needed for the mutation step and the state particles are useful for a quick evaluation of the incremental likelihood in the subsequent correction step.
- The $U_{1:t}^i$'s are not required for the actual implementation of the algorithm but are useful to provide a heuristic explanation for the validity of the algorithm.

Mutation Step

- Essentially one iteration of PFMH algorithm.
- For each particle i:
  - a proposed value ϑ_t^i,
  - an associated particle filter approximation $\hat{p}(Y_{1:t}|\vartheta_t^i)$ of the likelihood,
  - and a sequence of random vectors $U_{1:t}^*$ drawn from the distribution $p(U_{1:t})$.
- The densities $p(U_{1:t}^i)$ and $p(U_{1:t}^*)$ cancel from the formula for the acceptance probability $\alpha(\vartheta_t^i|\hat{\theta}_t^i)$.

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.