Particle Filters

Ed Herbst

November 19, 2019

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.

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

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

Some nonlinear models in macro

Gust et al. (2017): estimates a nonlinear DSGE subject to the zero lower bound.

Bocola (2016): a nonlinear model of sovereign default.

Fernndez-Villaverde et al. (2009): a macroeconomic model with stochastic volatility.

Key question: how to estimate model using likelihood techniques?

Cannot use Kalman filter – instead use a particle filter.

Particle Filters

There are many particle filters...

We will focus on three types:

1. Bootstrap PF
2. A generic PF
3. A conditionally-optimal PF

Filtering - General Idea

State-space representation of nonlinear DSGE model

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

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 $s_t|Y_{1:t}$.

1. Initialization at time $t-1$: $p(s_{t-1}|Y_{1:t-1}, \theta)$
2. Forecasting t given $t-1$:
   - Transition equation:
     $p(s_t|Y_{1:t-1}, \theta) = \int p(s_t|s_{t-1}, Y_{1:t-1}, \theta) p(s_{t-1}|Y_{1:t-1}, \theta) ds_{t-1}$
   - Measurement equation:
     $p(y_t|Y_{1:t-1}, \theta) = \int p(y_t|s_t, Y_{1:t-1}, \theta) p(s_t|Y_{1:t-1}, \theta) ds_t$
3. 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

1. **Initialization.** Draw the initial particles from the distribution $s_0^j \overset{iid}{\sim} p(s_0)$ and set $W_0^j = 1$, $j = 1, \ldots, M$.
2. **Recursion.** For $t = 1, \ldots, T$:
   2.1 **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 $t = 1, \ldots, T$:
   2.1 **Forecasting s_t.**
   2.2 **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 $t = 1, \ldots, T$:
   2.1 **Forecasting s_t.**
   2.2 **Forecasting y_t.** Define the incremental weights

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

   2.3 **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 $t = 1, \ldots, T$:
   2.1 **Forecasting s_t.**
   2.2 **Forecasting y_t.**
   2.3 **Updating.**
   2.4 **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

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

Likelihood Approximation

- The approximation of the log likelihood function is given by

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

- One can show that the approximation of the likelihood function is unbiased.
- This implies that the approximation of the log likelihood function is downward biased.

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.

An empirical introduction to BSPF

Let's check the BSPF on a linear process

$$
\begin{aligned}
s_t &= \rho s_{t-1} + \sigma_e \epsilon_t, \quad \epsilon_t \sim N(0,1) \\
y_t &= 2 s_t + \sigma_u u_t, \quad u_t \sim N(0,1)
\end{aligned}
$$

Let's also assume that $s_0 \sim N(1,1)$.

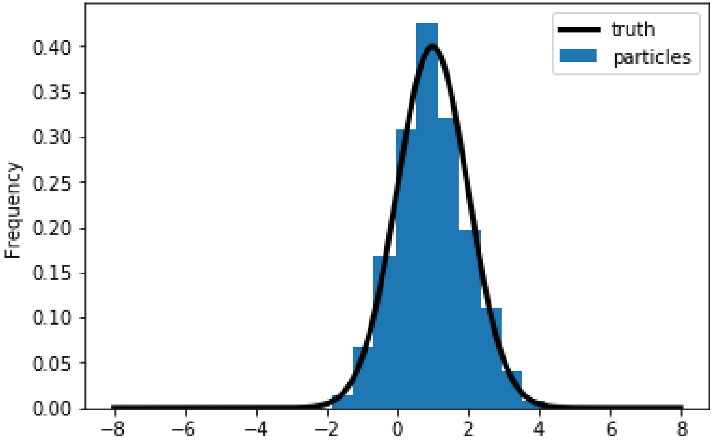
$\rho = 0.8$.

$\sigma_e = 0.1$

We are going to go through one iteration as the particle filter, with $M = 1000$ particles.

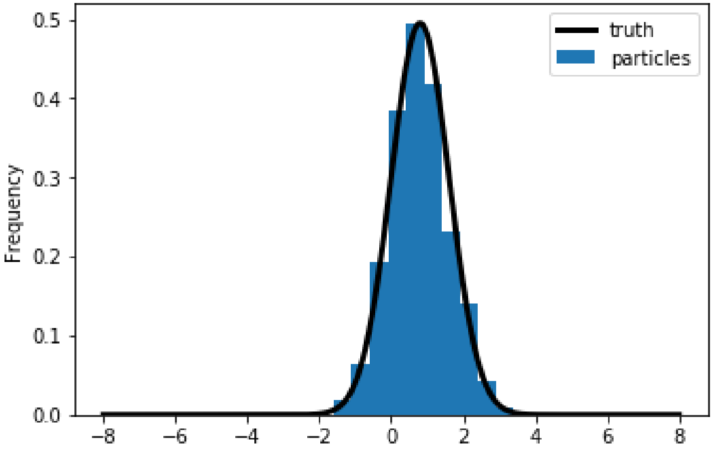
Initialization

To obtain draws from s_0, we draw 1000 particles from a $N(1,1)$.



Forecasting s_1

For each of the 1000 particles, we simulate from $s_1^i = \rho s_0^i + \sigma_e e^i$ with $e^i \sim N(0, 1)$.



Updating s_1

Now it's time to reweight the particles based on the how well they actually predicted y_1.

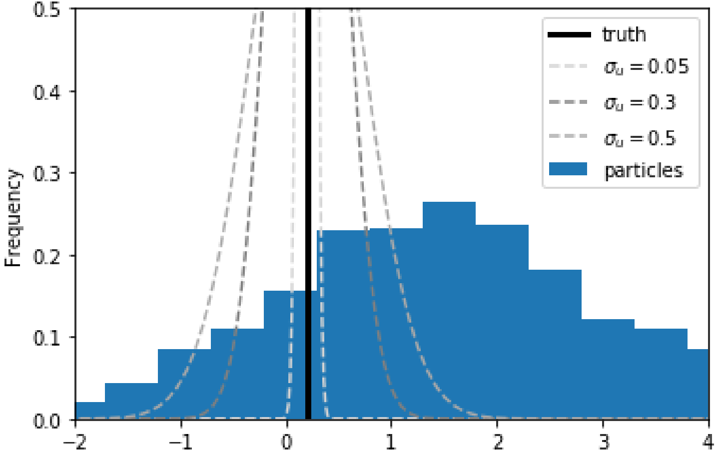
To predict y_1, we simply multiply s_t^i by 2.

How good is this prediction, let's think about in the context of ME.

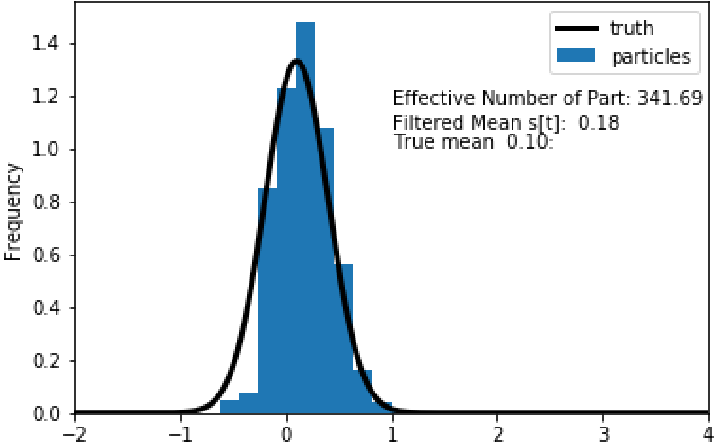
$$y_1 = 0.2, \quad \sigma_u \in \{0.05, 0.3, 0.5\}$$

If the ME is very small, the only particles that make very accurate predictions are worthwhile.

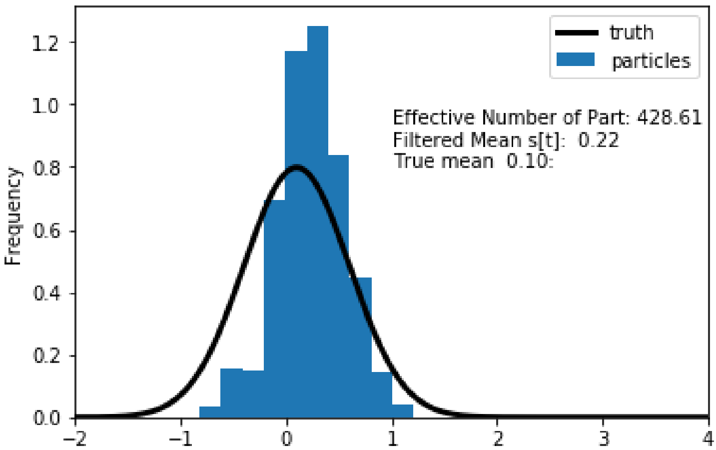
Predicting y_1



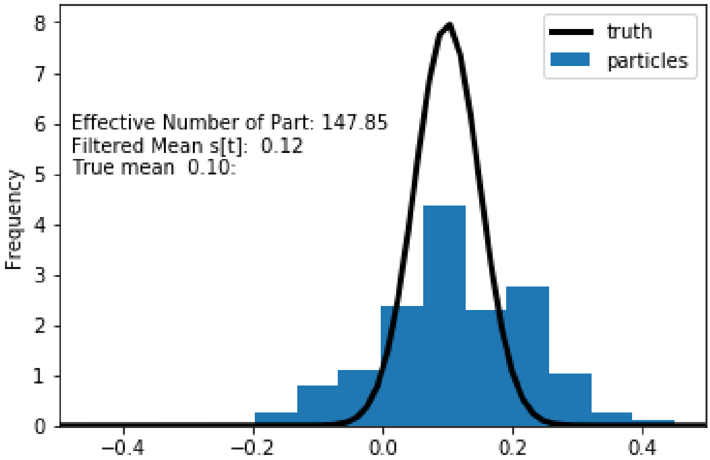
Updated $s_1, \sigma_u = 0.3$



Updated $s_1, \sigma_u = 0.5$



Updated $s_1, \sigma_u = 0.05$



Generic Particle Filter

1. **Initialization.** Same as BS PF
2. **Recursion.** For $t = 1, \ldots, T$:
   2.1 **Forecasting s_t.** Draw $\tilde{s}_t^j$ from density $g_t(\tilde{s}_t | s_{t-1}^j, \theta)$ and define
   $$\omega_t^j = \frac{p(\tilde{s}_t^j | s_{t-1}^j, \theta)}{g_t(\tilde{s}_t^j | s_{t-1}^j, \theta)}. \tag{11}$$
   An approximation of $\mathbb{E}[h(s_t)|Y_{1:t-1}, \theta]$ is given by
   $$\hat{h}_{t,M} = \frac{1}{M} \sum_{j=1}^{M} h(\tilde{s}_t^j) \omega_t^j W_{t-1}^j. \tag{12}$$
   2.2 **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
   $$\hat{p}(y_t | Y_{1:t-1}, \theta) = \frac{1}{M} \sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j. \tag{13}$$
   2.3 **Updating.** Same as BS PF
   2.4 **Selection.** Same as BS PF

Asymptotics

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

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

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

$$
\begin{aligned}
& \hat{h}_{t,M} - \mathbb{E}[h(s_t)|Y_{1:t-1}] \qquad (14) \\
&= \frac{1}{M}\sum_{j=1}^{M}\left(h(\tilde{s}^j_t) - \mathbb{E}_{p(\cdot|s^j_{t-1})}[h]\right)W^j_{t-1} \\
&\quad + \frac{1}{M}\sum_{j=1}^{M}\left(\mathbb{E}_{p(\cdot|s^j_{t-1})}[h]W^j_{t-1} - \mathbb{E}[h(s_t)|Y_{1:t-1}]\right) \\
&= I + II,
\end{aligned}
$$

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

Asymptotics

- Updating step approximates

$$\mathbb{E}[h(s_t)|Y_{1:t}] = \frac{\int h(s_t)p(y_t|s_t)p(s_t|Y_{1:t-1})ds_t}{\int p(y_t|s_t)p(s_t|Y_{1:t-1})ds_t} \approx \frac{\frac{1}{M}\sum_{j=1}^{M} h(\tilde{s}_t^j)\tilde{w}_t^j W_{t-1}^j}{\frac{1}{M}\sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j} \tag{15}$$

- Define the normalized incremental weights as

$$v_t(s_t) = \frac{p(y_t|s_t)}{\int p(y_t|s_t)p(s_t|Y_{1:t-1})ds_t}. \tag{16}$$

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

$$\begin{aligned} &\sqrt{M}\big(\tilde{h}_{t,M} - \mathbb{E}[h(s_t)|Y_{1:t}]\big) \qquad (17)\\ &\quad \Longrightarrow \quad N\big(0, \tilde{\Omega}_t(h)\big), \quad \tilde{\Omega}_t(h) = \hat{\Omega}_t\big(v_t(s_t)(h(s_t) - \mathbb{E}[h(s_t)|Y_{1:t}])\big). \end{aligned}$$

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

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
$$p(m_t, s_t|Y_{1:t}) = p(m_t|Y_{1:t})p(s_t|m_t, Y_{1:t}), \tag{18}$$
where
$$s_t|(m_t, Y_{1:t}) \sim N\big(\bar{s}_{t|t}(m_t), P_{t|t}(m_t)\big). \tag{19}$$
- 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
$$\begin{aligned}
&\int h(m_t, s_t)p(m_t, s_t, Y_{1:t})d(m_t, s_t) \qquad (20)\\
&= \int \left[\int h(m_t, s_t)p(s_t|m_t, Y_{1:t})ds_t\right] p(m_t|Y_{1:t})dm_t\\
&\approx \frac{1}{M}\sum_{j=1}^{M}\left[\int h(m_t^j, s_t^j)p_N\big(s_t|\bar{s}_{t|t}^j, P_{t|t}^j\big)ds_t\right] W_t^j.
\end{aligned}$$

More on Conditionally-Linear Models

- We used Rao-Blackwellization to reduce variance:

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

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

$$\omega_t^j = \frac{p(\tilde{m}_t^j|m_{t-1}^j)}{g_t(\tilde{m}_t^j|m_{t-1}^j)}. \tag{21}$$

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

$$\begin{aligned}\tilde{s}_{t|t-1}^j &= \Phi_0(\tilde{m}_t^j) + \Phi_1(\tilde{m}_t^j)s_{t-1}^j \\ P_{t|t-1}^j &= \Phi_\epsilon(\tilde{m}_t^j)\Sigma_\epsilon(\tilde{m}_t^j)\Phi_\epsilon(\tilde{m}_t^j)' \\ \tilde{y}_{t|t-1}^j &= \Psi_0(\tilde{m}_t^j) + \Psi_1(\tilde{m}_t^j)t + \Psi_2(\tilde{m}_t^j)\tilde{s}_{t|t-1}^j \\ F_{t|t-1}^j &= \Psi_2(\tilde{m}_t^j)P_{t|t-1}^j\Psi_2(\tilde{m}_t^j)' + \Sigma_u.\end{aligned} \tag{22}$$

More on Conditionally-Linear Models

- Then,

$$\int h(m_t, s_t) p(m_t, s_t | Y_{1:t-1}) d(m_t, s_t) \tag{23}$$
$$= \int \left[\int h(m_t, s_t) p(s_t | m_t, Y_{1:t-1}) ds_t \right] p(m_t | Y_{1:t-1}) dm_t$$
$$\approx \frac{1}{M} \sum_{j=1}^{M} \left[\int h(m_t^j, s_t^j) p_N\left(s_t | \tilde{s}^j_{t|t-1}, P^j_{t|t-1}\right) ds_t \right] \omega_t^j W_{t-1}^j$$

- The likelihood approximation is based on the incremental weights

$$\tilde{w}_t^j = p_N\left(y_t | \tilde{y}^j_{t|t-1}, F^j_{t|t-1}\right) \omega_t^j. \tag{24}$$

- 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:

$$\begin{aligned}
\tilde{s}^j_{t|t} &= \tilde{s}^j_{t|t-1} + P^j_{t|t-1} \Psi_2(\tilde{m}_t^j)' \left(F^j_{t|t-1}\right)^{-1} (y_t - \bar{y}^j_{t|t-1}) \\
\tilde{P}^j_{t|t} &= P^j_{t|t-1} - P^j_{t|t-1} \Psi_2(\tilde{m}_t^j)' \left(F^j_{t|t-1}\right)^{-1} \Psi_2(\tilde{m}_t^j) P^j_{t|t-1}.
\end{aligned}$$

Particle Filter For Conditionally Linear Models

1. **Initialization.**
2. **Recursion.** For $t = 1, \ldots, T$:
   2.1 **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 (21), and compute $\tilde{s}_{t|t-1}^j$ and $P_{t|t-1}^j$ according to (22). An approximation of $\mathbb{E}[h(s_t, m_t)|Y_{1:t-1}, \theta]$ is given by (24).
   2.2 **Forecasting y_t.** Compute the incremental weights $\tilde{w}_t^j$ according to (24). Approximate the predictive density $p(y_t|Y_{1:t-1}, \theta)$ by

$$\hat{p}(y_t|Y_{1:t-1}, \theta) = \frac{1}{M} \sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j. \tag{26}$$

   2.3 **Updating.** Define the normalized weights

$$\tilde{W}_t^j = \frac{\tilde{w}_t^j W_{t-1}^j}{\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j} \tag{27}$$

   and compute $\tilde{s}_{t|t}^j$ and $\tilde{P}_{t|t}^j$ according to (25). An approximation of $\mathbb{E}[h(m_t, s_t)|Y_{1:t}, \theta]$ can be obtained from $\{\tilde{m}_t^j, \tilde{s}_{t|t}^j, \tilde{P}_{t|t}^j, \tilde{W}_t^j\}$

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

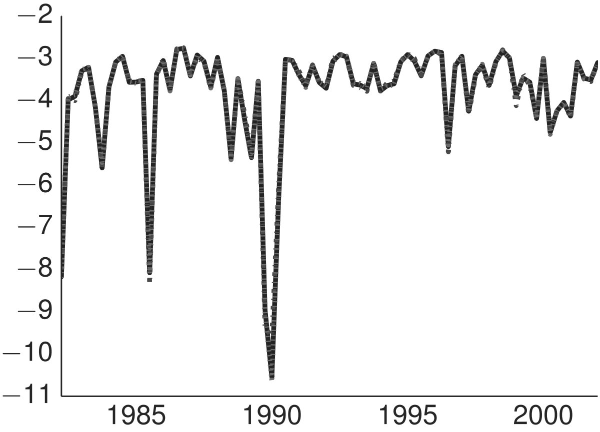
Illustration 1: Small-Scale DSGE Model

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

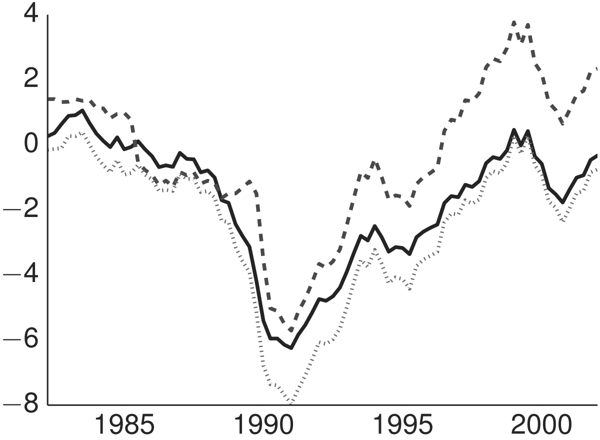
$\ln \hat{p}(y_t|Y_{1:t-1}, \theta^m)$ vs. $\ln p(y_t|Y_{1:t-1}, \theta^m)$



Notes: The results depicted in the figure are based on a single run of the bootstrap PF (dashed, $M = 40,000$), the conditionally-optimal PF (dotted, $M = 400$), and the Kalman filter (solid).

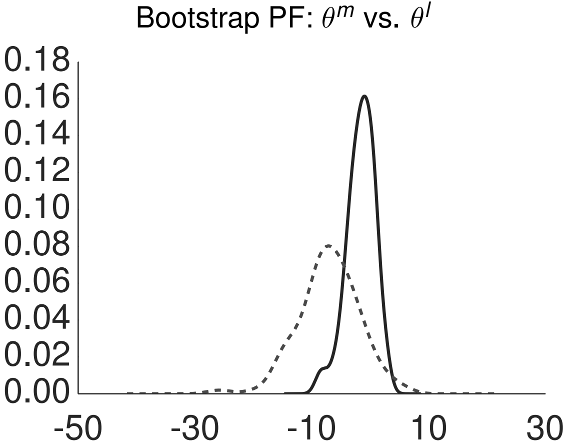
Filtered State

$\widehat{\mathbb{E}}[\hat{g}_t|Y_{1:t},\theta^m]$ vs. $\mathbb{E}[\hat{g}_t|Y_{1:t},\theta^m]$



Notes: The results depicted in the figure are based on a single run of the bootstrap PF (dashed, $M = 40,000$), the conditionally-optimal PF (dotted, $M = 400$), and the Kalman filter (solid).

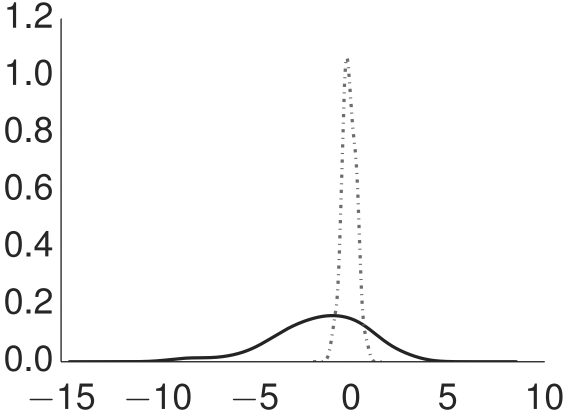
Distribution of Log-Likelihood Approximation Errors}



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}

θ^m: Bootstrap vs. Cond. Opt. PF



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 ($M = 400$).

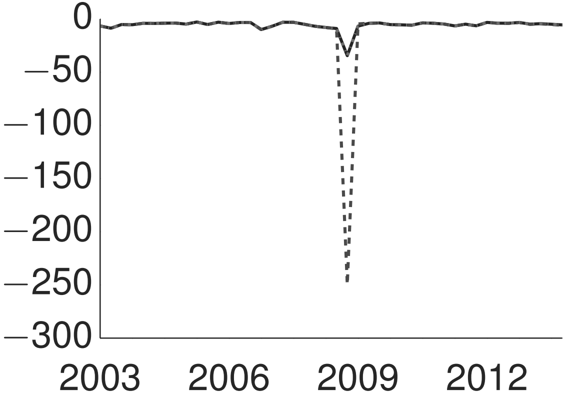
Summary Statistics for Particle Filters

	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

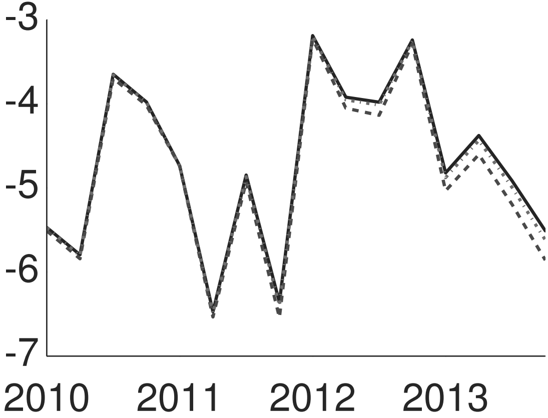
Mean of Log-likelihood Increments $\ln \hat{p}(y_t | Y_{1:t-1}, \theta^m)$



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.

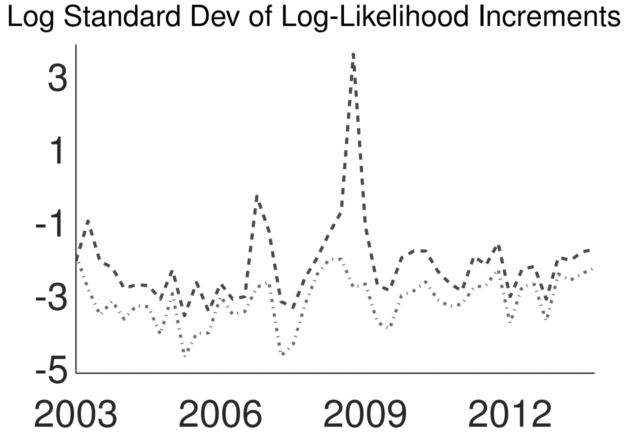
Great Recession and Beyond

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.

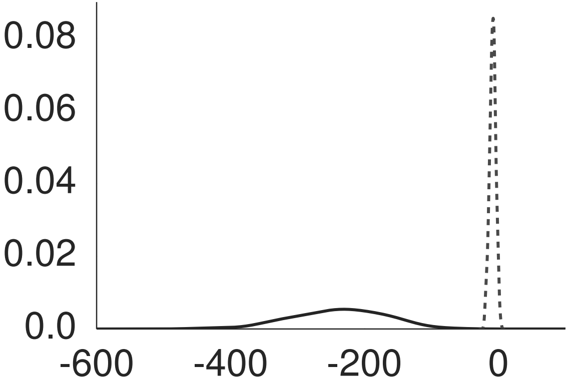
Great Recession and Beyond



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

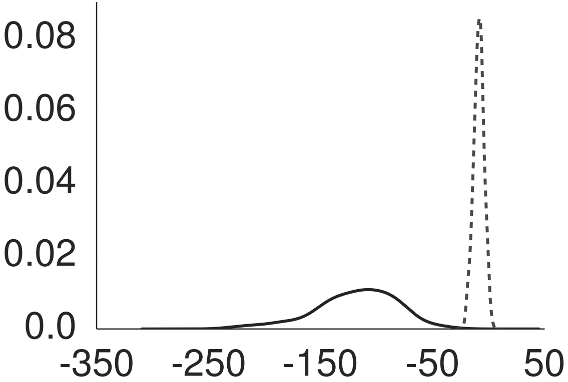
BS ($M = 40,000$) versus CO ($M = 4,000$)



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

BS ($M = 400,000$) versus CO ($M = 4,000$)



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

Tempered Particle Filter

- Use sequence of distributions between the forecast and updated state distributions.
- Candidates? Well, *the PF will work arbitrarily well when $\Sigma_u \to \infty$.*
- Reduce measurement error variance from an inflated initial level $\Sigma_u(\theta)/\phi_1$ to the nominal level $\Sigma_u(\theta)$.

The Key Idea

- Define

$$p_n(y_t|s_t,\theta) \;\propto\; \phi_n^{d/2}|\Sigma_u(\theta)|^{-1/2}\exp\bigg\{-\frac{1}{2}(y_t-\Psi(s_t,t;\theta))' \times\phi_n\Sigma_u^{-1}(\theta)(y_t-\Psi(s_t,t;\theta))\bigg\},$$

  where:

$$\phi_1<\phi_2<\ldots<\phi_{N_\phi}=1.$$

- Bridge posteriors given s_{t-1}:

$$p_n(s_t|y_t,s_{t-1},\theta)\propto p_n(y_t|s_t,\theta)p(s_t|s_{t-1},\theta).$$

- bridge posteriors given $Y_{1:t-1}$:

$$p_n(s_t|Y_{1:t})=\int p_n(s_t|y_t,s_{t-1},\theta)p(s_{t-1}|Y_{1:t-1})ds_{t-1}.$$

Algorithm Overview

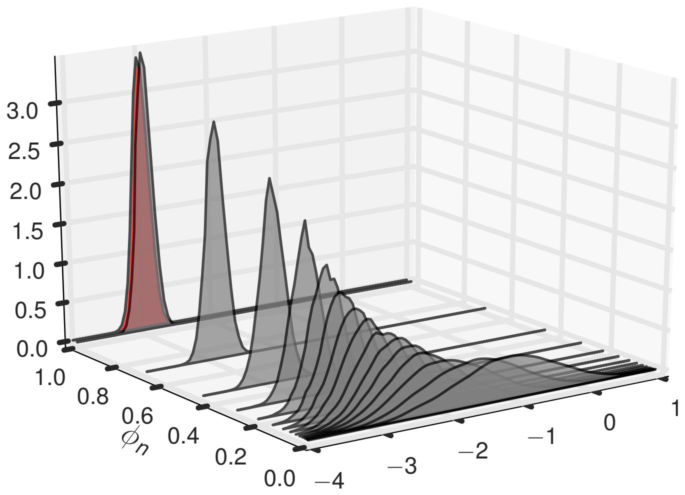
- For each t we start with the BS-PF iteration by simulating the state-transition equation forward.
- Incremental weights are obtained based on inflated measurement error variance Σ_u/ϕ_1.
- *Then we start the tempering iterations*...
- After the tempering iterations are completed we proceed to $t+1$...

Overview}

- If $N_\phi = 1$, this collapses to the Bootstrap particle filter.

- For each time period t, we embed a "static" SMC sampler used for parameter estimation Iterate over $n = 1, \ldots, N_\phi$:
  - Correction step: change particle weights (importance sampling)
  - Selection step: equalize particle weights (resampling of particles)
  - Mutation step: change particle values (based on Markov transition kernel generated with Metropolis-Hastings algorithm)
  - Each step approximates the same $\int h(s_t)p_n(s_t|Y_{1:t},\theta)ds_t$.

An Illustration: $p_n(s_t|Y_{1:t})$, $n = 1, \ldots, N_\phi$.



Choice of ϕ_n

- Based on Geweke and Frischknecht (2014).

- Express post-correction inefficiency ratio as

$$\mathrm{InEff}(\phi_n) = \frac{\frac{1}{M}\sum_{j=1}^{M} \exp[-2(\phi_n - \phi_{n-1})e_{j,t}]}{\left(\frac{1}{M}\sum_{j=1}^{M} \exp[-(\phi_n - \phi_{n-1})e_{j,t}]\right)^2}$$

  where

$$e_{j,t} = \frac{1}{2}(y_t - \Psi(s_t^{j,n-1}, t; \theta))' \Sigma_u^{-1} (y_t - \Psi(s_t^{j,n-1}, t; \theta)).$$

- Pick target ratio r^* and solve equation $\mathrm{InEff}(\phi_n^*) = r^*$ for ϕ_n^*.

Small-Scale Model: PF Summary Statistics

	BSPF	TPF			
Number of Particles M	40k	4k	4k	40k	40k
Target Ineff. Ratio r^*		2	3	2	3
High Posterior Density: $\theta = \theta^m$					
Bias	-1.4	-0.9	-1.5	-0.3	-.05
StdD	1.9	1.4	1.7	0.4	0.6
$T^{-1}\sum_{t=1}^{T} N_{\phi,t}$	1.0	4.3	3.2	4.3	3.2
Average Run Time (s)	0.8	0.4	0.3	4.0	3.3
Low Posterior Density: $\theta = \theta^l$					
Bias	-6.5	-2.1	-3.1	-0.3	-0.6
StdD	5.3	2.1	2.6	0.8	1.0
$T^{-1}\sum_{t=1}^{T} N_{\phi,t}$	1.0	4.4	3.3	4.4	3.3
Average Run Time (s)	1.6	0.4	0.3	3.7	2.9

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).
- The book also discusses SMC^2.

Embedding PF Likelihoods into Posterior Samplers$\}$

- $\{p(Y|\theta), p(\theta|Y), p(Y)\}$, 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

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

   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 $\tilde{S}_t = (\tilde{s}_t^1, \ldots, \tilde{s}_t^M)'$ 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 $A_t = (A_t^1, \ldots, A_t^N)$ 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

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

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

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?

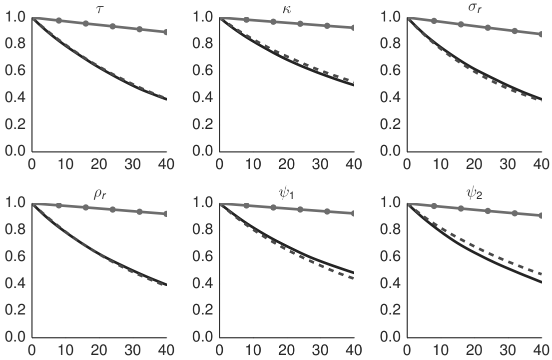
- We can express acceptance probability directly in terms of $\hat{p}(Y_{1:T}|\theta)$.
- Need to generate a proposed draw for both θ and $U_{1:T}$: ϑ and $U^*_{1:T}$.
- The proposal distribution for $(\vartheta, U^*_{1:T})$ in the MH algorithm is given by $q(\vartheta|\theta^{(i-1)})p(U^*_{1:T})$.
-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

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

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.

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-
τ	2.63	2.62	2.64	66.17	126.76	1360.22	0.020	0.028	0.0
κ	0.82	0.81	0.82	128.00	97.11	1887.37	0.007	0.006	0.0
ψ_1	1.88	1.88	1.87	113.46	159.53	749.22	0.011	0.013	0.0
ψ_2	0.64	0.64	0.63	61.28	56.10	681.85	0.011	0.010	0.0
ρ_r	0.75	0.75	0.75	108.46	134.01	1535.34	0.002	0.002	0.0
ρ_g	0.98	0.98	0.98	94.10	88.48	1613.77	0.001	0.001	0.0
ρ_z	0.88	0.88	0.88	124.24	118.74	1518.66	0.001	0.001	0.0
$r^{(A)}$	0.44	0.44	0.44	148.46	151.81	1115.74	0.016	0.016	0.0
$\pi^{(A)}$	3.32	3.33	3.32	152.08	141.62	1057.90	0.017	0.016	0.0
$\gamma^{(Q)}$	0.59	0.59	0.59	106.68	142.37	899.34	0.006	0.007	0.0
σ_r	0.24	0.24	0.24	35.21	179.15	1105.99	0.001	0.002	0.0
σ_g	0.68	0.68	0.67	98.22	64.18	1490.81	0.003	0.002	0.0
σ_z	0.32	0.32	0.32	84.77	61.55	575.90	0.001	0.001	0.0
$\ln \hat{p}(Y)$	-357.14	-357.17	-358.32				0.040	0.038	0.9

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.

References I

Bocola, L. (2016). The pass-through of sovereign risk. *Journal of Political Economy*, 124(4):879926.

Fernndez-Villaverde, J., Guerrn-Quintana, P., Rubio-Ramrez, J., and Uribe, M. (2009). Risk matters: The real effects of volatility shocks.

Gust, C., Herbst, E., Lpez-Salido, D., and Smith, M. E. (2017). The empirical implications of the interest-rate lower bound. *American Economic Review*, 107(7):19712006.