Sequential Monte Carlo Methods for DSGE Models

Ed Herbst*  Frank Schorfheide+

*Federal Reserve Board
+University of Pennsylvania, PIER, CEPR, and NBER

June 23, 2017

---

1The views expressed in this presentation are those of the presenters and do not necessarily reflect the views of the Federal Reserve Board of Governors or the Federal Reserve System.
These lectures use material from our joint work:

SMC can help to

**Lecture 1**

- **approximate the posterior of** \( \theta \): Chopin (2002) ... Durham and Geweke (2013) ... Creal (2007), Herbst and Schorfheide (2014)

**Lecture 2**

- **or both:** \( SMC^2 \): Chopin, Jacob, and Papaspiliopoulos (2012) ... Herbst and Schorfheide (2015)
Lecture 2
• DSGE models are inherently nonlinear.

• Sometimes linear approximations are sufficiently accurate...
• but in other applications nonlinearities may be important:
  • asset pricing;
  • borrowing constraints;
  • zero lower bound on nominal interest rates;
  • ...

• Nonlinear state-space representation requires nonlinear filter:

\[
\begin{align*}
y_t &= \Psi(s_t, t; \theta) + u_t, \quad u_t \sim F_u(\cdot; \theta) \\
s_t &= \Phi(s_{t-1}, \epsilon_t; \theta), \quad \epsilon_t \sim F_\epsilon(\cdot; \theta).
\end{align*}
\]
• 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 nonlinear DSGE model
  - Measurement Eq.: \( y_t = \Psi(s_t, t; \theta) + u_t, \quad u_t \sim F_u(\cdot; \theta) \)
  - State Transition: \( s_t = \Phi(s_{t-1}, \epsilon_t; \theta), \quad \epsilon_t \sim F_\epsilon(\cdot; \theta). \)
- 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

1 **Initialization.** Draw the initial particles from the distribution $s_0^j \sim p(s_0)$ and set $W_0^j = 1$, $j = 1, \ldots, M$.

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

   1 **Forecasting $s_t$.** Propagate the period $t-1$ particles $\{s_{t-1}^j, W_{t-1}^j\}$ by iterating the state-transition equation forward:

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

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

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

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

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

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

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

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

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

   where $n$ here denotes the dimension of $y_t$. 

E. Herbst and F. Schorfheide

SMC for DSGE Models
Bootstrap Particle Filter

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

   \[ \tilde{w}_t^i = p(y_t|\tilde{s}_t^i, \theta). \]  
   (6)

3 Updating. Define the normalized weights

   \[ \tilde{W}_t^i = \tilde{w}_t^i W_{t-1}^i. \]  
   (7)

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

\[ \tilde{h}_{t,M} = \frac{1}{M} \sum_{j=1}^{M} h(\tilde{s}_t^j) \tilde{W}_t^j. \]  
(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 $\{s^j_t\}_{j=1}^M$ denote $M$ iid draws from a multinomial distribution characterized by support points and weights $\{\tilde{s}^j_t, \tilde{W}^j_t\}$ and set $W^j_t = 1$ for $j = 1, \ldots, M$. An approximation of $E[h(s_t) | Y_{1:t}, \theta]$ is given by

\[
\bar{h}_{t,M} = \frac{1}{M} \sum_{j=1}^{M} h(s^j_t) W^j_t.
\]  

(9)

E. Herbst and F. Schorfheide

SMC for DSGE Models
The approximation of the log likelihood function is given by

$$\ln \hat{p}(Y_{1:T} | \theta) = \sum_{t=1}^{T} \ln \left( \frac{1}{M} \sum_{j=1}^{M} \tilde{w}_t^i W_t^j \right).$$  \hspace{1cm} (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 $\Sigma_u$, holding everything else fixed, increases the variance of the particle weights, and reduces the accuracy of Monte Carlo approximation.
1 **Forecasting** $s_t$. Draw $\tilde{s}_t^j$ from density $g_t(\tilde{s}_t^j|s_{t-1}^j, \theta)$ and define

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

(11)

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

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

(12)

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

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

(13)

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

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

(14)

3 **Updating / Selection.** Same as BS PF
• Conditionally-optimal importance distribution:

\[ g_t(\tilde{s}_t | s_{t-1}^j) = p(\tilde{s}_t | y_t, s_{t-1}^j). \]

This is the posterior of \( s_t \) given \( s_{t-1}^j \). 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:

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

where \( m_t \) follows a discrete Markov-switching process.
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$\theta^m$</th>
<th>$\theta^l$</th>
<th>Parameter</th>
<th>$\theta^m$</th>
<th>$\theta^l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau$</td>
<td>2.09</td>
<td>3.26</td>
<td>$\kappa$</td>
<td>0.98</td>
<td>0.89</td>
</tr>
<tr>
<td>$\psi_1$</td>
<td>2.25</td>
<td>1.88</td>
<td>$\psi_2$</td>
<td>0.65</td>
<td>0.53</td>
</tr>
<tr>
<td>$\rho_r$</td>
<td>0.81</td>
<td>0.76</td>
<td>$\rho_g$</td>
<td>0.98</td>
<td>0.98</td>
</tr>
<tr>
<td>$\rho_z$</td>
<td>0.93</td>
<td>0.89</td>
<td>$\rho(A)$</td>
<td>0.34</td>
<td>0.19</td>
</tr>
<tr>
<td>$\pi(A)$</td>
<td>3.16</td>
<td>3.29</td>
<td>$\gamma(Q)$</td>
<td>0.51</td>
<td>0.73</td>
</tr>
<tr>
<td>$\sigma_r$</td>
<td>0.19</td>
<td>0.20</td>
<td>$\sigma_g$</td>
<td>0.65</td>
<td>0.58</td>
</tr>
<tr>
<td>$\sigma_z$</td>
<td>0.24</td>
<td>0.29</td>
<td>$\ln p(Y</td>
<td>\theta)$</td>
<td>-306.5</td>
</tr>
</tbody>
</table>
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).
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 \( N_{\text{run}} = 100 \) runs of the PF. Solid line is \( \theta = \theta^m \); dashed line is \( \theta = \theta^l \) (\( M = 40,000 \)).
Notes: Density estimate of $\hat{\Delta}_1 = \ln \hat{p}(Y_{1:T}|\theta) - \ln p(Y_{1:T}|\theta)$ based on $N_{\text{run}} = 100$ 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

<table>
<thead>
<tr>
<th></th>
<th>Bootstrap</th>
<th>Cond. Opt.</th>
<th>Auxiliary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Particles</td>
<td>40,000</td>
<td>400</td>
<td>40,000</td>
</tr>
<tr>
<td>Number of Repetitions</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

**High Posterior Density: \( \theta = \theta^m \)**

<table>
<thead>
<tr>
<th></th>
<th>( \hat{\Delta}_1 )</th>
<th>( \hat{\Delta}_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bias</td>
<td>-1.39</td>
<td>0.32</td>
</tr>
<tr>
<td>StdD</td>
<td>2.03</td>
<td>0.37</td>
</tr>
</tbody>
</table>

**Low Posterior Density: \( \theta = \theta^l \)**

<table>
<thead>
<tr>
<th></th>
<th>( \hat{\Delta}_1 )</th>
<th>( \hat{\Delta}_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bias</td>
<td>-7.01</td>
<td>-0.70</td>
</tr>
<tr>
<td>StdD</td>
<td>4.68</td>
<td>-0.70</td>
</tr>
</tbody>
</table>

**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.
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 \(N_{\text{run}} = 100\) runs of the filters.
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.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Particles $M$</td>
<td>40,000</td>
<td>400,000</td>
</tr>
<tr>
<td>Number of Repetitions</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>High Posterior Density:</td>
<td>$\theta = \theta^m$</td>
<td></td>
</tr>
<tr>
<td>Bias $\hat{\Delta}_1$</td>
<td>-238.49</td>
<td>-118.20</td>
</tr>
<tr>
<td>StdD $\hat{\Delta}_1$</td>
<td>68.28</td>
<td>35.69</td>
</tr>
<tr>
<td>Bias $\hat{\Delta}_2$</td>
<td>-1.00</td>
<td>-1.00</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Posterior Density:</td>
<td>$\theta = \theta^l$</td>
<td></td>
</tr>
<tr>
<td>Bias $\hat{\Delta}_1$</td>
<td>-253.89</td>
<td>-128.13</td>
</tr>
<tr>
<td>StdD $\hat{\Delta}_1$</td>
<td>65.57</td>
<td>41.25</td>
</tr>
<tr>
<td>Bias $\hat{\Delta}_2$</td>
<td>-1.00</td>
<td>-1.00</td>
</tr>
</tbody>
</table>

**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 $N_{run} = 100$. 
• Use sequence of distributions between the forecast and updated state distributions.

• Candidates? Well, the PF will work arbitrarily well when $\Sigma_u \rightarrow \infty$.

• 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

\[
p_n(y_t | s_t, \theta) \propto \phi_n^{d/2} |\Sigma_u(\theta)|^{-1/2} \exp \left\{ -\frac{1}{2} (y_t - \Psi(s_t, t; \theta))' \right. \\
\left. \times \phi_n \Sigma_u^{-1}(\theta)(y_t - \Psi(s_t, t; \theta)) \right\},
\]

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 \( Y_{1:t-1} \):

\[
p_n(s_t | Y_{1:t-1}) = \int p_n(s_t | y_t, s_{t-1}, \theta)p(s_{t-1} | Y_{1:t-1}) ds_{t-1}.
\]
- 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 $\Sigma_u/\phi_1$.

- Then we start the tempering iterations...

- After the tempering iterations are completed we proceed to $t + 1$...
• 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 [See Lecture 1]:

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: $p_n(s_t \mid Y_{1:t}), \ n = 1, \ldots, N_\phi$. 
Choice of $\phi_n$

- Based on Geweke and Frischknecht (2014).

- Express post-correction inefficiency ratio as

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

where

$$e_{j,t} = \frac{1}{2}(y_t - \Psi(s_{t}^{j,n-1}, t; \theta))^\prime \Sigma_u^{-1}(y_t - \Psi(s_{t}^{j,n-1}, t; \theta))$$

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

### Small-Scale Model: PF Summary Statistics

<table>
<thead>
<tr>
<th></th>
<th>BSPF</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of Particles</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M$</td>
<td>40k</td>
<td>4k</td>
<td>4k</td>
<td>40k</td>
<td>40k</td>
</tr>
<tr>
<td><strong>Target Ineff. Ratio</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r^*$</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

#### High Posterior Density: $\theta = \theta^m$

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bias</strong></td>
<td>-1.4</td>
<td>-0.9</td>
<td>-1.5</td>
<td>-0.3</td>
<td>-0.05</td>
</tr>
<tr>
<td><strong>StdD</strong></td>
<td>1.9</td>
<td>1.4</td>
<td>1.7</td>
<td>0.4</td>
<td>0.6</td>
</tr>
<tr>
<td>$T^{-1} \sum_{t=1}^T N_{\phi,t}$</td>
<td>1.0</td>
<td>4.3</td>
<td>3.2</td>
<td>4.3</td>
<td>3.2</td>
</tr>
<tr>
<td><strong>Average Run Time (s)</strong></td>
<td>0.8</td>
<td>0.4</td>
<td>0.3</td>
<td>4.0</td>
<td>3.3</td>
</tr>
</tbody>
</table>

#### Low Posterior Density: $\theta = \theta^l$

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bias</strong></td>
<td>-6.5</td>
<td>-2.1</td>
<td>-3.1</td>
<td>-0.3</td>
<td>-0.6</td>
</tr>
<tr>
<td><strong>StdD</strong></td>
<td>5.3</td>
<td>2.1</td>
<td>2.6</td>
<td>0.8</td>
<td>1.0</td>
</tr>
<tr>
<td>$T^{-1} \sum_{t=1}^T N_{\phi,t}$</td>
<td>1.0</td>
<td>4.4</td>
<td>3.3</td>
<td>4.4</td>
<td>3.3</td>
</tr>
<tr>
<td><strong>Average Run Time (s)</strong></td>
<td>1.6</td>
<td>0.4</td>
<td>0.3</td>
<td>3.7</td>
<td>2.9</td>
</tr>
</tbody>
</table>

E. Herbst and F. Schorfheide  
SMC for DSGE Models
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 $SMC^2$. 
• 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 $\vartheta$ 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.
• Results are based on $N_{\text{run}} = 20$ 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 $N_{\text{run}} = 20$ runs to compute posterior statistics.
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Posterior Mean (Pooled)</th>
<th>Inefficiency Factors</th>
<th>Std Dev of Means</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KF</td>
<td>CO-PF</td>
<td>BS-PF</td>
</tr>
<tr>
<td>$\tau$</td>
<td>2.63</td>
<td>2.62</td>
<td>2.64</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>0.82</td>
<td>0.81</td>
<td>0.82</td>
</tr>
<tr>
<td>$\psi_1$</td>
<td>1.88</td>
<td>1.88</td>
<td>1.87</td>
</tr>
<tr>
<td>$\psi_2$</td>
<td>0.64</td>
<td>0.64</td>
<td>0.63</td>
</tr>
<tr>
<td>$\rho_r$</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>$\rho_g$</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
</tr>
<tr>
<td>$\rho_z$</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
</tr>
<tr>
<td>$r(A)$</td>
<td>0.44</td>
<td>0.44</td>
<td>0.44</td>
</tr>
<tr>
<td>$\pi(A)$</td>
<td>3.32</td>
<td>3.33</td>
<td>3.32</td>
</tr>
<tr>
<td>$\gamma(Q)$</td>
<td>0.59</td>
<td>0.59</td>
<td>0.59</td>
</tr>
<tr>
<td>$\sigma_r$</td>
<td>0.24</td>
<td>0.24</td>
<td>0.24</td>
</tr>
<tr>
<td>$\sigma_g$</td>
<td>0.68</td>
<td>0.68</td>
<td>0.67</td>
</tr>
<tr>
<td>$\sigma_z$</td>
<td>0.32</td>
<td>0.32</td>
<td>0.32</td>
</tr>
<tr>
<td>$\ln \hat{p}(Y)$</td>
<td>-357.14</td>
<td>-357.17</td>
<td>-358.32</td>
</tr>
</tbody>
</table>
• 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.