

Particle Filtering

Frank Schorfheide
University of Pennsylvania
Bradley Visitor Lectures

October 23, 2017

Approximating the Likelihood Function

- 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:**

$$y_t = \Psi(s_t, t; \theta) + u_t, \quad u_t \sim F_u(\cdot; \theta)$$

$$s_t = \Phi(s_{t-1}, \epsilon_t; \theta), \quad \epsilon_t \sim F_\epsilon(\cdot; \theta).$$

- 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

$$\text{Measurement Eq. : } y_t = \Psi(s_t, t; \theta) + u_t, \quad u_t \sim F_u(\cdot; \theta)$$

$$\text{State Transition : } s_t = \Phi(s_{t-1}, \epsilon_t; \theta), \quad \epsilon_t \sim F_\epsilon(\cdot; \theta).$$

- 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:

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

- 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:

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

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

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

① **Initialization.**

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

① **Forecasting** s_t .

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

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

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

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

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

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

where n here denotes the dimension of y_t .

① **Initialization.**

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

① **Forecasting** s_t .

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

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

③ **Updating.** Define the normalized weights

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

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

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

① **Initialization.**

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

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 $j = 1, \dots, M$.

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

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

- The approximation of the **log likelihood function** is given by

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

- 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.

- ① **Forecasting** s_t . Draw \tilde{s}_t^j from density $g_t(\tilde{s}_t^j | s_{t-1}^j, \theta)$ and define

$$\omega_t^j = \frac{p(\tilde{s}_t^j | s_{t-1}^j, \theta)}{g_t(\tilde{s}_t^j | s_{t-1}^j, \theta)}. \quad (11)$$

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

$$\hat{h}_{t,M} = \frac{1}{M} \sum_{j=1}^M h(\tilde{s}_t^j) \omega_t^j W_{t-1}^j. \quad (12)$$

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

$$\tilde{w}_t^j = p(y_t | \tilde{s}_t^j, \theta) \omega_t^j. \quad (13)$$

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

$$\hat{p}(y_t | Y_{1:t-1}, \theta) = \frac{1}{M} \sum_{j=1}^M \tilde{w}_t^j W_{t-1}^j. \quad (14)$$

- 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:

$$y_t = \Psi_0(m_t) + \Psi_1(m_t)t + \Psi_2(m_t)s_t + u_t, \quad u_t \sim N(0, \Sigma_u),$$

$$s_t = \Phi_0(m_t) + \Phi_1(m_t)s_{t-1} + \Phi_\epsilon(m_t)\epsilon_t, \quad \epsilon_t \sim N(0, \Sigma_\epsilon),$$

where m_t follows a discrete Markov-switching process.

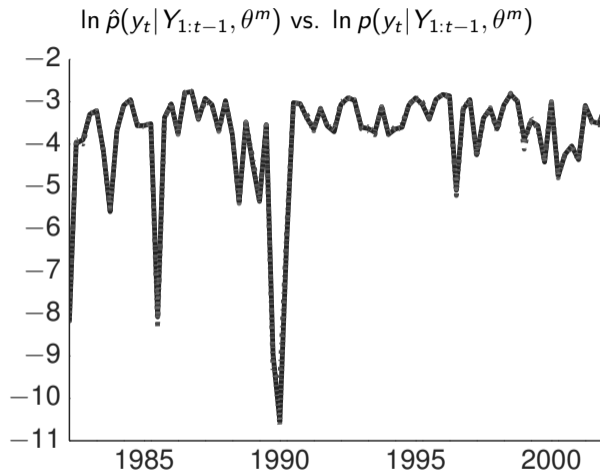
- We will now apply PFs to linearized DSGE models.
- This allows us to compare the Monte Carlo approximation to the “truth.”
- Small-scale New Keynesian DSGE model
- Smets-Wouters model

Illustration 1: Small-Scale DSGE Model

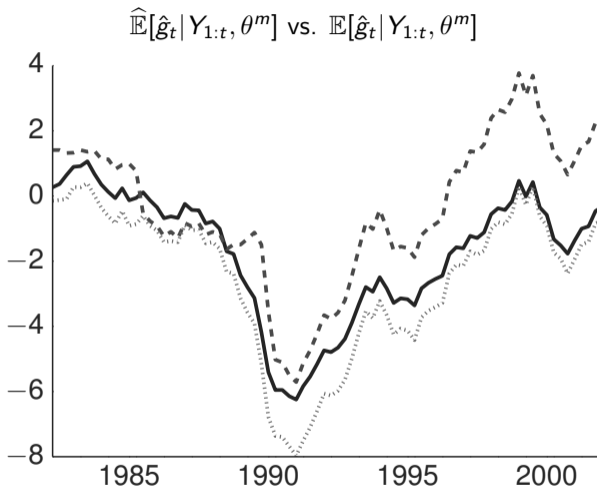
Parameter Values For Likelihood Evaluation

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

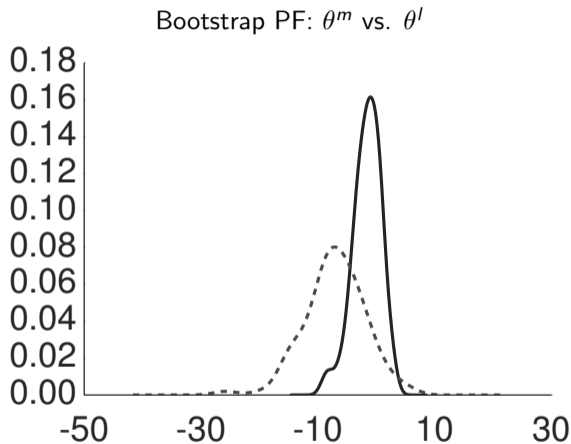


Notes: The results depicted in the figure are based on a single run of the bootstrap PF (dashed, $M = 40,000$), the conditionally-optimal PF (dotted, $M = 400$), and the Kalman filter (solid).



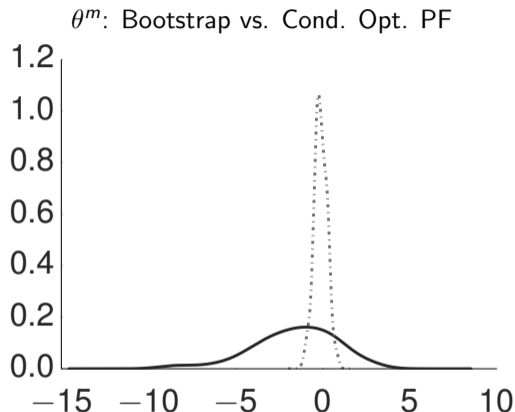
Notes: The results depicted in the figure are based on a single run of the bootstrap PF (dashed, $M = 40,000$), the conditionally-optimal PF (dotted, $M = 400$), and the Kalman filter (solid).

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 $N_{run} = 100$ 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 ($M = 400$).

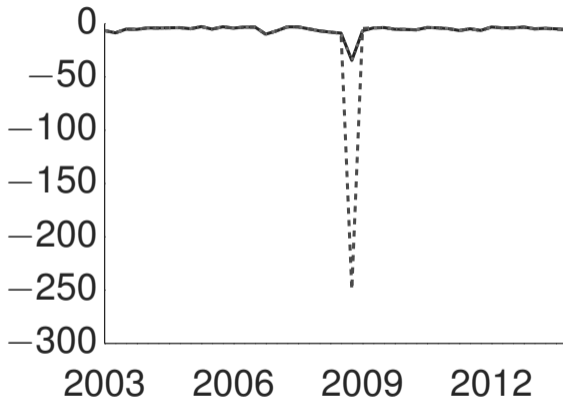
Summary Statistics for Particle Filters

	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.

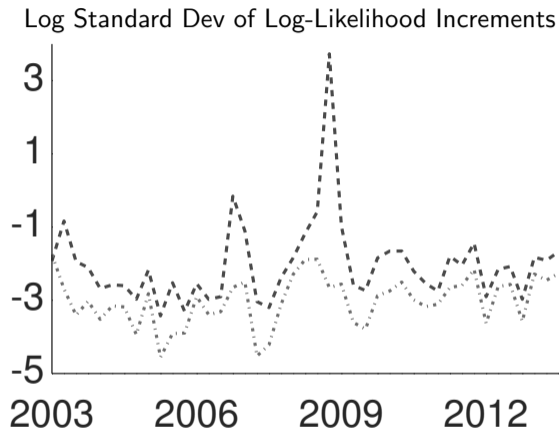
Great Recession and Beyond

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



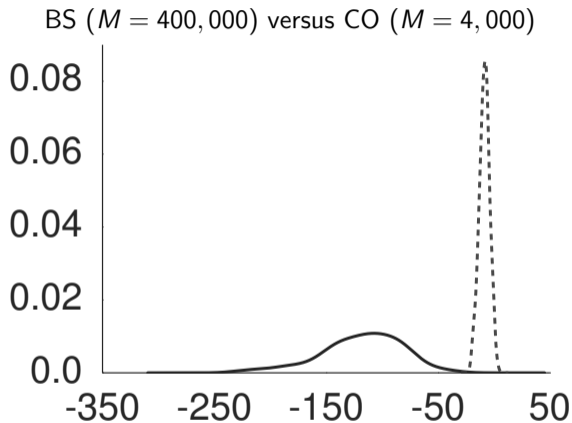
Notes: Solid lines represent results from Kalman filter. Dashed lines correspond to bootstrap particle filter ($M = 40,000$) and dotted lines correspond to conditionally-optimal particle filter ($M = 400$). Results are based on $N_{run} = 100$ runs of the filters.

Great Recession and Beyond



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: 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 $N_{run} = 100$.

- 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 $\Sigma_u(\theta)$.**

- 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:

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

- 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 $Y_{1:t-1}$:

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

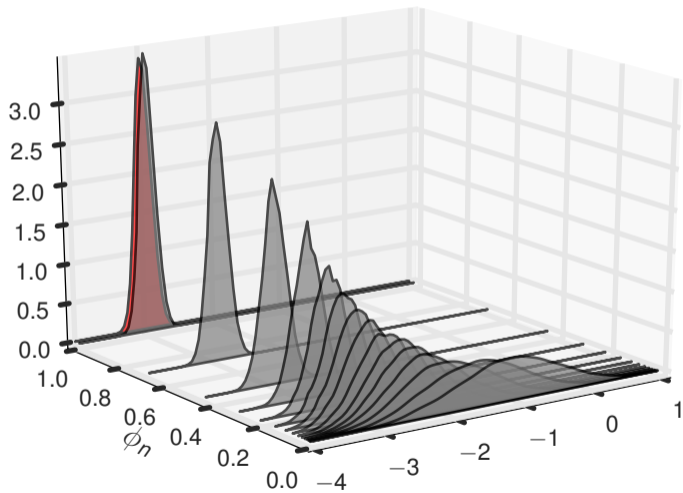
- For each t we start with the BS-PF iteration by simulating the state-transition equation forward.
- Incremental weights are obtained based on inflated measurement error variance Σ_u/ϕ_1 .
- Then we start the tempering iterations...
- After the tempering iterations are completed we proceed to $t + 1$...

- 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 $n = 1, \dots, N_\phi$:

- **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: $p_n(s_t | Y_{1:t})$, $n = 1, \dots, N_\phi$.



- 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

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

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

- 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 $p(\theta|Y)$.

Embedding PF Likelihoods into Posterior Samplers

- Key idea: express the particle filter approximation of the likelihood function as

$$\hat{p}(Y_{1:T}|\theta) = g(Y_{1:T}|\theta, U_{1:T}), \quad U_{1:T} \sim p(U_{1:T}) = \prod_{t=1}^T p(U_t).$$

- The PFMH algorithm samples from the joint posterior

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

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

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

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

For $i = 1$ to N :

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

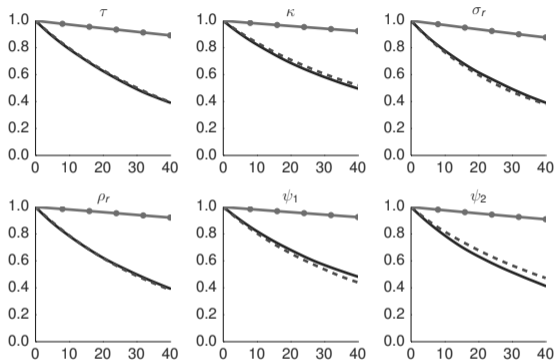
$$\alpha(\vartheta|\theta^{i-1}) = \min \left\{ 1, \frac{\hat{p}(Y|\vartheta)p(\vartheta)/q(\vartheta|\theta^{i-1})}{\hat{p}(Y|\theta^{i-1})p(\theta^{i-1})/q(\theta^{i-1}|\vartheta)} \right\}$$

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

Small-Scale DSGE: Accuracy of MH Approximations

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

Autocorrelation of PFMH Draws



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

Small-Scale DSGE: Accuracy of MH Approximations

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

- 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.