

# Sequential Monte Carlo Methods

Frank Schorfheide

University of Pennsylvania

Gerzensee Ph.D. Course on Bayesian Macroeconometrics

May 28, 2019

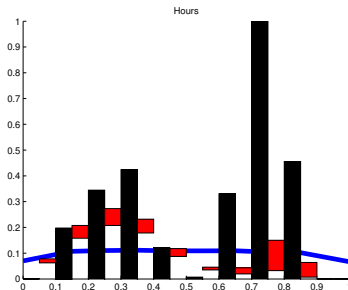
- Posterior expectations can be approximated by Monte Carlo averages.
- If we have draws from  $\{\theta^s\}_{s=1}^N$  from  $p(\theta|Y)$ , then (under some regularity conditions)

$$\frac{1}{N} \sum_{s=1}^N h(\theta^s) \xrightarrow{a.s.} \mathbb{E}[h(\theta)|Y].$$

- “Standard” approach in DSGE model literature (Schorfheide, 2000; Otrok, 2001): use Markov chain Monte Carlo (MCMC) methods to generate a sequence of serially correlated draws  $\{\theta^s\}_{s=1}^N$ .
- Unfortunately, “standard” MCMC can be quite inaccurate, especially in medium and large-scale DSGE models:
  - disentangling importance of internal versus external propagation mechanism;
  - determining the relative importance of shocks.

# What Do We Mean by Inaccurate?

- “News” about the future may be important for business cycle fluctuations.
- Use DSGE model to construct histogram (reflecting posterior parameter uncertainty) for the relative importance of news shocks for fluctuations in hours worked.
- Repeat posterior computations 20 times; black is “inaccurate”; red is “accurate”; blue is prior.



- **Previously:** Modify MCMC algorithms to overcome weaknesses: blocking of parameters; tailoring of (mixture) proposal densities
  - Kohn et al. (2010)
  - Chib and Ramamurthy (2011)
  - Curdia and Reis (2011)
  - Herbst (2012)
- **Now, we use sequential Monte Carlo (SMC)** (more precisely, sequential importance sampling) instead:
  - Better suited to handle irregular and multimodal posteriors associated with large DSGE models.
  - Algorithms can be easily parallelized.
- SMC = Importance Sampling on Steriods. We build on
  - Theoretical work: Chopin (2004); Del Moral, Doucet, Jasra (2006)
  - Applied work: Creal (2007); Durham and Geweke (2011, 2012)

# Review: Importance Sampling

- $\pi(\theta) = p(\theta|Y) = f(\theta)/Z$  is posterior.
- We can approximate  $\mathbb{E}_\pi[h(\theta)]$  using draws from some other distribution  $g(\cdot)$ :

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

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

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

- References:

- General: Kloek and van Dijk (1978), Geweke (1989);
- DSGE model application: Dejong, Ingram, and Whiteman (2000).

# From Importance Sampling to Sequential Importance Sampling

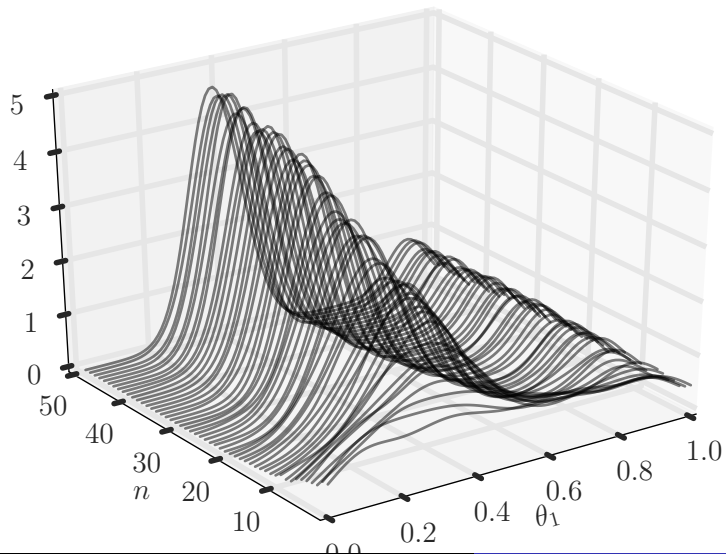
- In general, it's hard to construct a good proposal density  $g(\theta)$ ,
- especially if the posterior has several peaks and valleys.
- **Idea - Part 1:** it might be easier to find a proposal density for

$$\pi_n(\theta) = \frac{[p(Y|\theta)]^{\phi_n} p(\theta)}{\int [p(Y|\theta)]^{\phi_n} p(\theta) d\theta} = \frac{f_n(\theta)}{Z_n}.$$

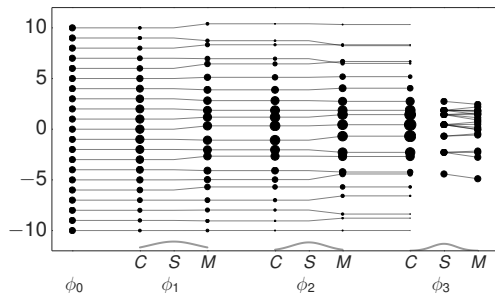
at least if  $\phi_n$  is close to zero.

- **Idea - Part 2:** We can try to turn a proposal density for  $\pi_n$  into a proposal density for  $\pi_{n+1}$  and iterate, letting  $\phi_n \rightarrow \phi_N = 1$ .

# Illustration: Tempered Posteriors of $\theta_1$



# SMC Algorithm: A Graphical Illustration



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

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

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



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

- 1 **Correction.** Reweight the particles from stage  $n - 1$  by defining the incremental weights

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

and the normalized weights

$$\tilde{W}_n^i = \frac{\tilde{w}_n^i W_{n-1}^i}{\frac{1}{N} \sum_{i=1}^N \tilde{w}_n^i W_{n-1}^i}, \quad i = 1, \dots, N. \quad (2)$$

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

$$\tilde{h}_{n,N} = \frac{1}{N} \sum_{i=1}^N \tilde{W}_n^i h(\theta_{n-1}^i). \quad (3)$$

- 2 **Selection.**

① **Initialization.**

② **Recursion.** For  $n = 1, \dots, N_\phi$ ,

① **Correction.**

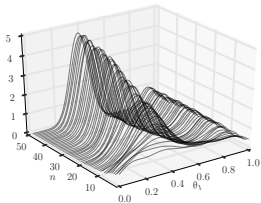
- ② **Selection. (Optional Resampling)** Let  $\{\hat{\theta}_i\}_{i=1}^N$  denote  $N$  iid draws from a multinomial distribution characterized by support points and weights  $\{\theta_{n-1}^i, \tilde{W}_n^i\}_{i=1}^N$  and set  $W_n^i = 1$ . An approximation of  $\mathbb{E}_{\pi_n}[h(\theta)]$  is given by

$$\hat{h}_{n,N} = \frac{1}{N} \sum_{i=1}^N W_n^i h(\hat{\theta}_n^i). \quad (4)$$

- ③ **Mutation.** Propagate the particles  $\{\hat{\theta}_i, W_n^i\}$  via  $N_{MH}$  steps of a MH algorithm with transition density  $\theta_n^i \sim K_n(\theta_n | \hat{\theta}_n^i; \zeta_n)$  and stationary distribution  $\pi_n(\theta)$ . An approximation of  $\mathbb{E}_{\pi_n}[h(\theta)]$  is given by

$$\bar{h}_{n,N} = \frac{1}{N} \sum_{i=1}^N h(\theta_n^i) W_n^i. \quad (5)$$

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



- Goal: strong law of large numbers (SLLN) and central limit theorem (CLT) as  $N \rightarrow \infty$  for every iteration  $n = 1, \dots, N_\phi$ .
- Regularity conditions:
  - proper prior;
  - bounded likelihood function;
  - $2 + \delta$  posterior moments of  $h(\theta)$ .
- Idea of proof (Chopin, 2004):
  - SLLN and CLT can be proved recursively.
  - For step  $n$  assume that  $n - 1$  approximation (with normalized weights) yields

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

- Initialization: SLLN and CLT for *iid* random variables because we sample from prior

# Theoretical Properties: Correction Step

- Suppose that the  $n - 1$  approximation (with normalized weights) yields

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

- Then

$$\sqrt{N} \left( \frac{\frac{1}{N} \sum_{i=1}^N h(\theta_{n-1}^i) [\rho(Y|\theta_{n-1}^i)]^{\phi_n - \phi_{n-1}} W_{n-1}^i}{\frac{1}{N} \sum_{i=1}^N [\rho(Y|\theta_{n-1}^i)]^{\phi_n - \phi_{n-1}} W_{n-1}^i} - \mathbb{E}_{\pi_n}[h(\theta)] \right) \\ \implies N(0, \tilde{\Omega}_n(h))$$

where

$$\tilde{\Omega}_n(h) = \Omega_{n-1}(v_{n-1}(\theta)(h - \mathbb{E}_{\pi_n}[h])) \quad v_{n-1}(\theta) = [\rho(Y|\theta)]^{\phi_n - \phi_{n-1}} \frac{Z_{n-1}}{Z_n}$$

- This step relies on likelihood evaluations from iteration  $n - 1$  that are already stored in memory.

- After resampling by drawing from iid multinomial distribution we obtain

$$\sqrt{N} \left( \frac{1}{N} \sum_{i=1}^N h(\hat{\theta}_i) W_n^i - \mathbb{E}_{\pi_n}[h] \right) \implies N(0, \hat{\Omega}(h)),$$

where

$$\hat{\Omega}_n(h) = \tilde{\Omega}(h) + \mathbb{V}_{\pi_n}[h]$$

- **Disadvantage** of resampling: it **adds noise**.
- **Advantage** of resampling: it equalizes the particle weights, reducing the variance of  $v_n(\theta)$  in  $\tilde{\Omega}_{n+1}(h) = \Omega_n(v_n(\theta)(h - \mathbb{E}_{\pi_{n+1}}[h]))$ .

# Theoretical Properties: Mutation

- We are using the Markov transition kernel  $K_n(\theta|\hat{\theta})$  to transform draws  $\hat{\theta}_n^i$  into draws  $\theta_n^i$ .
- To preserve the distribution of the  $\hat{\theta}_n^i$ 's it has to be the case that

$$\pi_n(\theta) = \int K_n(\theta|\hat{\theta})\pi_n(\hat{\theta})d\hat{\theta}.$$

- It can be shown that the overall asymptotic variance after the mutation is the sum of
  - the variance of the approximation of the conditional mean  $\mathbb{E}_{K_n(\cdot|\theta_{n-1})}[h(\theta)]$  which is given by

$$\hat{\Omega}(\mathbb{E}_{K_n(\cdot|\theta_{n-1})}[h(\theta)]);$$

- a weighted average of the conditional variance  $\mathbb{V}_{K_n(\cdot|\theta_{n-1})}[h(\theta)]$ :

$$\int W_{n-1}(\theta_{n-1})v_{n-1}(\theta_{n-1})\mathbb{V}_{K_n(\cdot|\theta_{n-1})}[h(\theta)]\pi_{n-1}(\theta_{n-1}).$$

- This step is *embarrassingly parallelizable*, well designed for single instruction, multiple data (SIMD) processing.

# More on Transition Kernel in Mutation Step

- Transition kernel  $K_n(\theta|\hat{\theta}_{n-1}; \zeta_n)$ : generated by running  $M$  steps of a Metropolis-Hastings algorithm.
- Lessons from DSGE model MCMC:
  - blocking of parameters can reduce persistence of Markov chain;
  - mixture proposal density avoids “getting stuck.”
- Blocking: Partition the parameter vector  $\theta_n$  into  $N_{blocks}$  equally sized blocks, denoted by  $\theta_{n,b}$ ,  $b = 1, \dots, N_{blocks}$ . (We generate the blocks for  $n = 1, \dots, N_\phi$  randomly prior to running the SMC algorithm.)
- Example: Mixture proposal density:

$$\begin{aligned} & \vartheta_b | (\theta_{n,b,m-1}^i, \theta_{n,-b,m}^i, \theta_{n,b}^*, \Sigma_{n,b}^*) \\ & \sim \alpha N\left(\theta_{n,b,m-1}^i, c_n^2 \Sigma_{n,b}^*\right) + \frac{1-\alpha}{2} N\left(\theta_{n,b,m-1}^i, c_n^2 \text{diag}(\Sigma_{n,b}^*)\right) \\ & \quad + \frac{1-\alpha}{2} N\left(\theta_{n,b}^*, c_n^2 \Sigma_{n,b}^*\right). \end{aligned}$$



# Adaptive Choice of $\zeta_n = (\theta_n^*, \Sigma_n^*, c_n)$

- **Infeasible adaption:**

- Let  $\theta_n^* = \mathbb{E}_{\pi_n}[\theta]$  and  $\Sigma_n^* = \mathbb{V}_{\pi_n}[\theta]$ .
- Adjust scaling factor according to

$$c_n = c_{n-1} f(R_{n-1}(\zeta_{n-1})),$$

where  $R_{n-1}(\cdot)$  is population rejection rate from iteration  $n - 1$  and

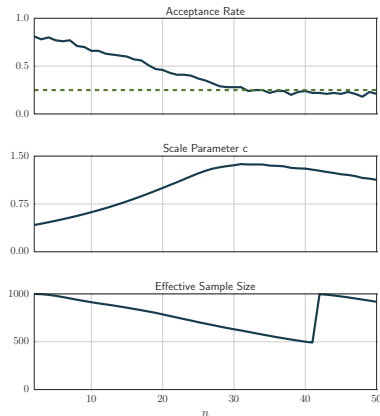
$$f(x) = 0.95 + 0.10 \frac{e^{16(x-0.25)}}{1 + e^{16(x-0.25)}}.$$

- **Feasible adaption** – use output from stage  $n - 1$  to replace  $\zeta_n$  by  $\hat{\zeta}_n$ :

- Use particle approximations of  $\mathbb{E}_{\pi_n}[\theta]$  and  $\mathbb{V}_{\pi_n}[\theta]$  based on  $\{\theta_{n-1}^i, \tilde{W}_n^i\}_{i=1}^N$ .
- Use actual rejection rate from stage  $n - 1$  to calculate  $\hat{c}_n = \hat{c}_{n-1} f(\hat{R}_{n-1}(\hat{\zeta}_{n-1}))$ .

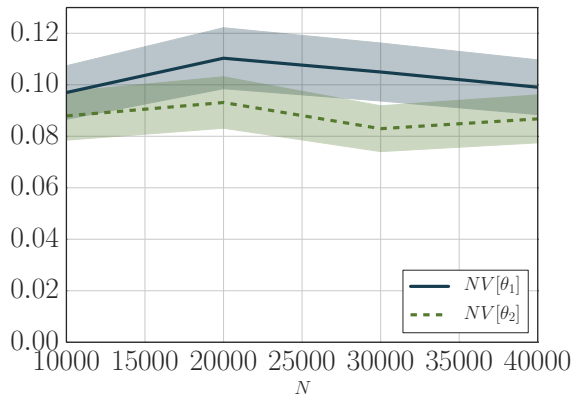
- **Result:** under suitable regularity conditions replacing  $\zeta_n$  by  $\hat{\zeta}_n$  where  $\sqrt{n}(\hat{\zeta}_n - \zeta_n) = O_p(1)$  does not affect the asymptotic variance of the MC approximation.

# Adaption of SMC Algorithm for Stylized State-Space Model



*Notes:* The dashed line in the top panel indicates the target acceptance rate of 0.25.

# Convergence of SMC Approximation for Stylized State-Space Model



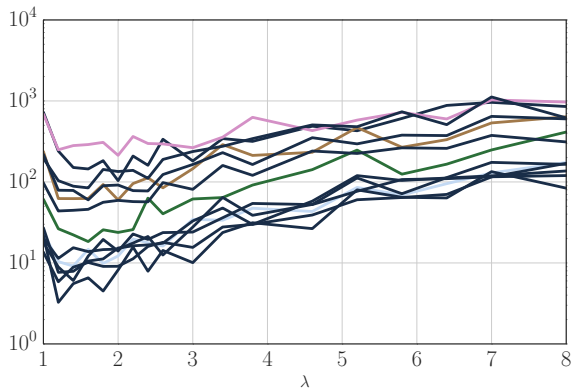
*Notes:* The figure shows  $NV[\bar{\theta}_j]$  for each parameter as a function of the number of particles  $N$ .  $\mathbb{V}[\bar{\theta}_j]$  is computed based on  $N_{run} = 1,000$  runs of the SMC algorithm with  $N_\phi = 100$ . The width of the bands is  $(2 \cdot 1.96) \sqrt{3/N_{run}} (NV[\bar{\theta}_j])$ .

- So far, we have used *multinomial resampling*. It's fairly intuitive and it is straightforward to obtain a CLT.
- But: *multinomial resampling is not particularly efficient*.
- The book contains a section on alternative resampling schemes (*stratified resampling, residual resampling...*)
- These alternative techniques are designed to achieve a variance reduction.
- Most resampling algorithms are not parallelizable because they rely on the normalized particle weights.

# Application 1: Small Scale New Keynesian Model

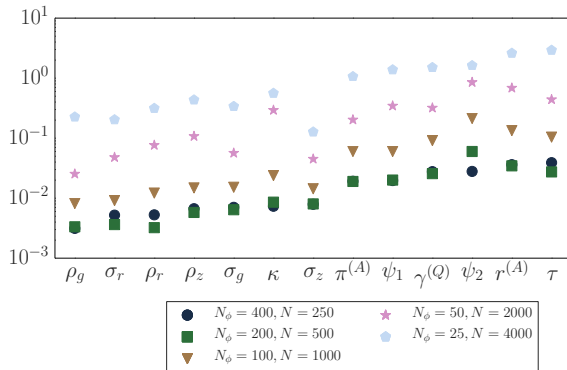
- We will take a look at the effect of various tuning choices on accuracy:
  - Tempering schedule  $\lambda$ :  $\lambda = 1$  is linear,  $\lambda > 1$  is convex.
  - Number of stages  $N_\phi$  versus number of particles  $N$ .
  - Number of blocks in mutation step versus number of particles.

# Effect of $\lambda$ on Inefficiency Factors $\text{InEff}_N[\bar{\theta}]$



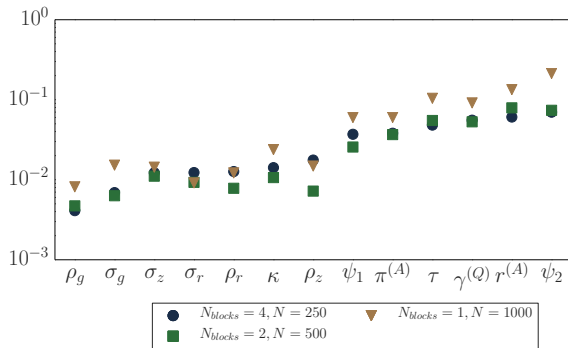
*Notes:* The figure depicts hairs of  $\text{InEff}_N[\bar{\theta}]$  as function of  $\lambda$ . The inefficiency factors are computed based on  $N_{run} = 50$  runs of the SMC algorithm. Each hair corresponds to a DSGE model parameter.

# Number of Stages $N_\phi$ vs Number of Particles $N$



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

# Number of blocks $N_{blocks}$ in Mutation Step vs Number of Particles $N$



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



# Marginal Likelihood Approximation

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

- Then

$$\begin{aligned} \frac{1}{N} \sum_{i=1}^N \tilde{w}_n^i W_{n-1}^i &\approx \int [p(Y|\theta)]^{\phi_n - \phi_{n-1}} \frac{p^{\phi_{n-1}}(Y|\theta)p(\theta)}{\int p^{\phi_{n-1}}(Y|\theta)p(\theta)d\theta} d\theta \\ &= \frac{\int p(Y|\theta)^{\phi_n} p(\theta) d\theta}{\int p(Y|\theta)^{\phi_{n-1}} p(\theta) d\theta} \end{aligned}$$

- Thus,

$$\prod_{n=1}^{N_\phi} \left( \frac{1}{N} \sum_{i=1}^N \tilde{w}_n^i W_{n-1}^i \right) \approx \int p(Y|\theta)p(\theta)d\theta.$$

# SMC Marginal Data Density Estimates

$N$	$N_\phi = 100$		$N_\phi = 400$	
	Mean( $\ln \hat{p}(Y)$ )	SD( $\ln \hat{p}(Y)$ )	Mean( $\ln \hat{p}(Y)$ )	SD( $\ln \hat{p}(Y)$ )
500	-352.19	(3.18)	-346.12	(0.20)
1,000	-349.19	(1.98)	-346.17	(0.14)
2,000	-348.57	(1.65)	-346.16	(0.12)
4,000	-347.74	(0.92)	-346.16	(0.07)

*Notes:* Table shows mean and standard deviation of log marginal data density estimates as a function of the number of particles  $N$  computed over  $N_{run} = 50$  runs of the SMC sampler with  $N_{blocks} = 4$ ,  $\lambda = 2$ , and  $N_{MH} = 1$ .

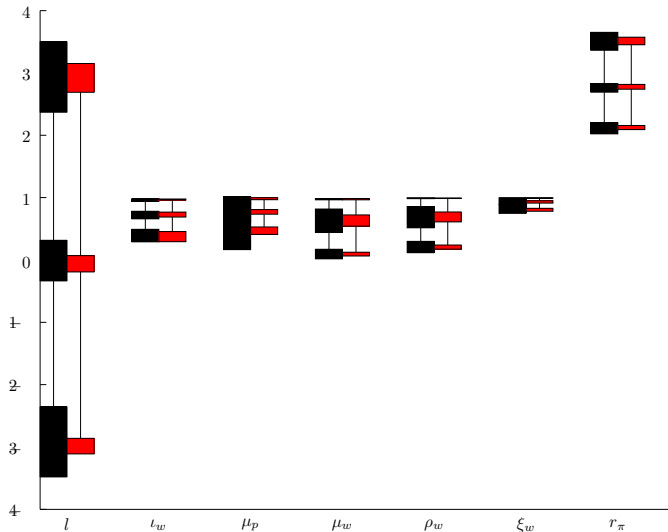
## Application 2: Estimation of Smets and Wouters (2007) Model

- Benchmark macro model, has been estimated many (many) times.
- “Core” of many larger-scale models.
- 36 estimated parameters.
- RWMH: 10 million draws (5 million discarded); SMC: 500 stages with 12,000 particles.
- We run the RWM (using a particular version of a parallelized MCMC) and the SMC algorithm on 24 processors for the same amount of time.
- We estimate the SW model twenty times using RWM and SMC and get essentially identical results.

## Application 2: Estimation of Smets and Wouters (2007) Model

- More interesting question: how does quality of posterior simulators change as one makes the priors more diffuse?
- Replace Beta by Uniform distributions; increase variances of parameters with Gamma and Normal prior by factor of 3.

# SW Model with DIFFUSE Prior: Estimation stability RWH (black) versus SMC (red)



# A Measure of Effective Number of Draws

- Suppose we could generate *iid*  $N_{eff}$  draws from posterior, then

$$\hat{\mathbb{E}}_{\pi}[\theta] \overset{approx}{\sim} N\left(\mathbb{E}_{\pi}[\theta], \frac{1}{N_{eff}} \mathbb{V}_{\pi}[\theta]\right).$$

- We can measure the variance of  $\hat{\mathbb{E}}_{\pi}[\theta]$  by running SMC and RWM algorithm repeatedly.
- Then,

$$N_{eff} \approx \frac{\mathbb{V}_{\pi}[\theta]}{\mathbb{V}[\hat{\mathbb{E}}_{\pi}[\theta] ]}$$

# Effective Number of Draws

Parameter	SMC			RWMH		
	Mean	STD(Mean)	$N_{eff}$	Mean	STD(Mean)	$N_{eff}$
$\sigma_l$	3.06	0.04	1058	3.04	0.15	60
$l$	-0.06	0.07	732	-0.01	0.16	177
$\nu_p$	0.11	0.00	637	0.12	0.02	19
$h$	0.70	0.00	522	0.69	0.03	5
$\Phi$	1.71	0.01	514	1.69	0.04	10
$r_\pi$	2.78	0.02	507	2.76	0.03	159
$\rho_b$	0.19	0.01	440	0.21	0.08	3
$\varphi$	8.12	0.16	266	7.98	1.03	6
$\sigma_p$	0.14	0.00	126	0.15	0.04	1
$\xi_p$	0.72	0.01	91	0.73	0.03	5
$\nu_w$	0.73	0.02	87	0.72	0.03	36
$\mu_p$	0.77	0.02	77	0.80	0.10	3
$\rho_w$	0.69	0.04	49	0.69	0.09	11
$\mu_w$	0.63	0.05	49	0.63	0.09	11
$\xi_w$	0.93	0.01	43	0.93	0.02	8

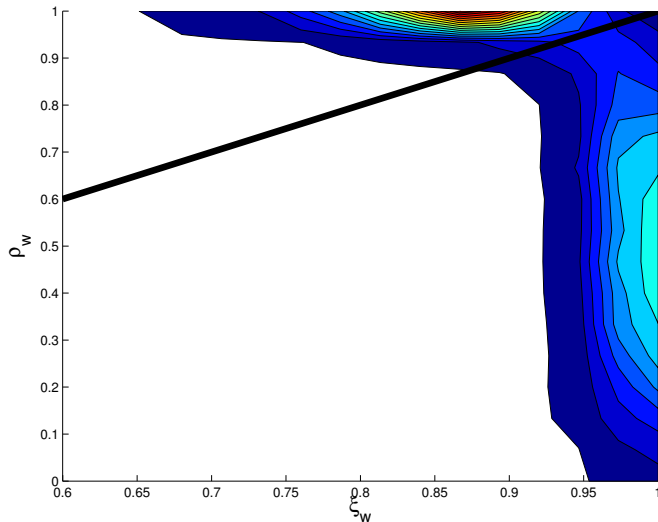
# A Closer Look at the Posterior: Two Modes

Parameter	Mode 1	Mode 2
$\xi_w$	0.844	0.962
$\iota_w$	0.812	0.918
$\rho_w$	0.997	0.394
$\mu_w$	0.978	0.267
Log Posterior	-804.14	-803.51

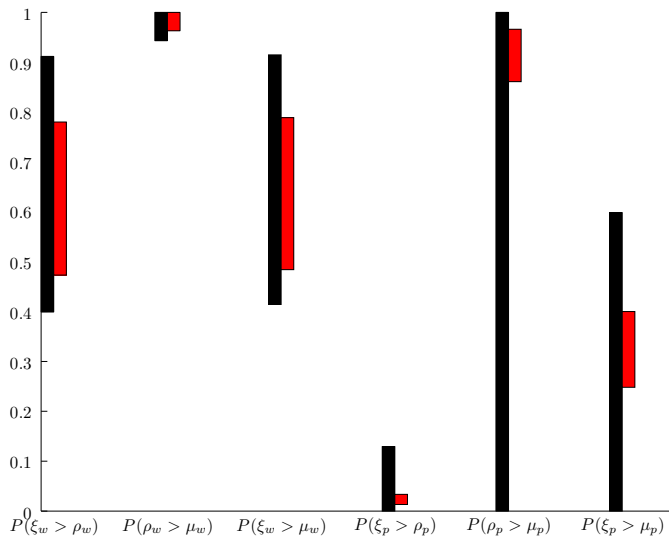
- **Mode 1** implies that wage persistence is driven by extremely **exogenous** persistent wage markup shocks.
- **Mode 2** implies that wage persistence is driven by **endogenous** amplification of shocks through the wage Calvo and indexation parameter.
- SMC is able to capture the two modes.



# A Closer Look at the Posterior: Internal $\xi_w$ versus External $\rho_w$ Propagation



# Stability of Posterior Computations: RWH (black) versus SMC (red)



- Bayesian model evaluation is based on marginal data density

$$p_n(Y) = \int [p(Y|\theta)]^{\phi_n} p(\theta) d\theta.$$

- Recall that marginal data density is a by-product of the importance sampler:

$$\hat{p}_n(Y) = \hat{Z}_n = \frac{1}{N} \sum_{i=1}^N \tilde{W}_n^i.$$

Algorithm, Method	MDD Estimate	Standard Deviation
SMC, Particle Estimate	-873.46	0.24
RWM, Harmonic Mean	-874.56	1.20