Bayesian Estimation of DSGE Models[1]
Chapter 5: Sequential Monte Carlo Methods

Ed Herbst[1] Frank Schorfheide[2]

[1]Federal Reserve Board

[2]University of Pennsylvania

February 17, 2016

[1]The views expressed in this paper are those of the authors and do not necessarily reflect the views of the Federal Reserve Board of Governors or the Federal Reserve System.

Introduction

- 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.

Introduction

- 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)

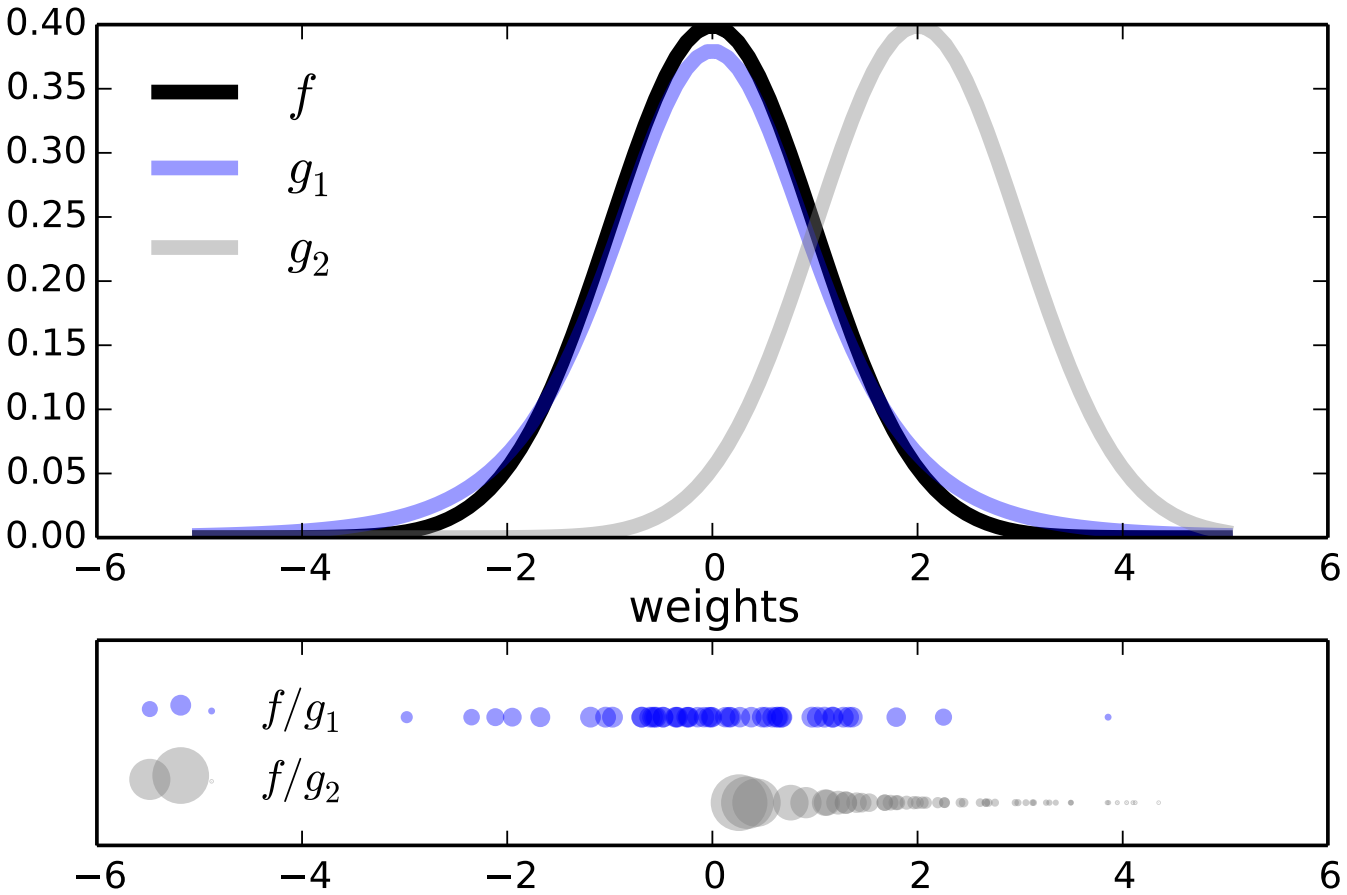
- Now, we use sequential Monte Carlo (SMC) (more precisely, sequential importance sampling) instead:
  - Better suited to handle irregular and multimodal posteriors associated with large DSGE models.
  - Algorithms can be easily parallelized.

- 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

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

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



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 ϕ_n is close to zero.

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

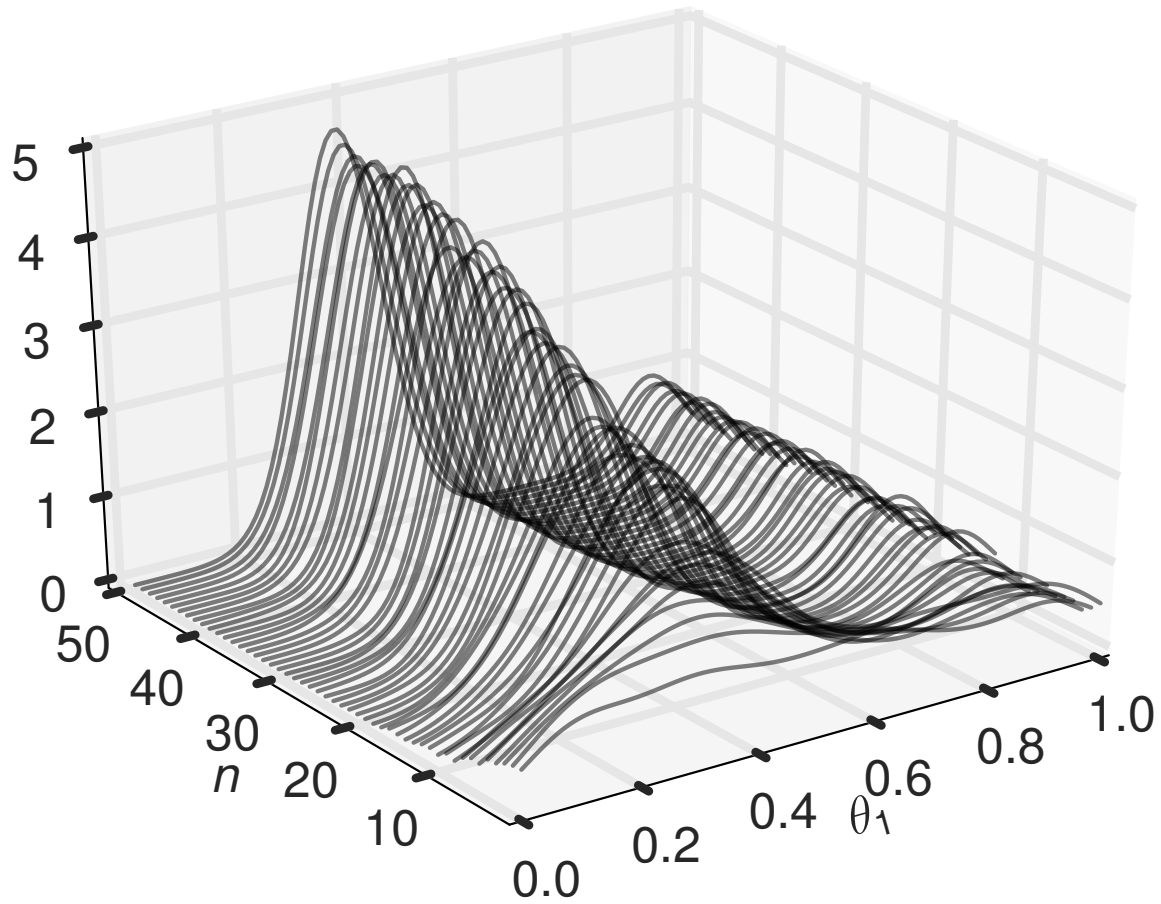
Illustration:

- Our state-space model:

$$y_t = [1\ 1]s_t, \quad s_t = \begin{bmatrix} \theta_1^2 & 0 \\ (1-\theta_1^2) - \theta_1\theta_2 & (1-\theta_1^2) \end{bmatrix} s_{t-1} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \epsilon_t.$$

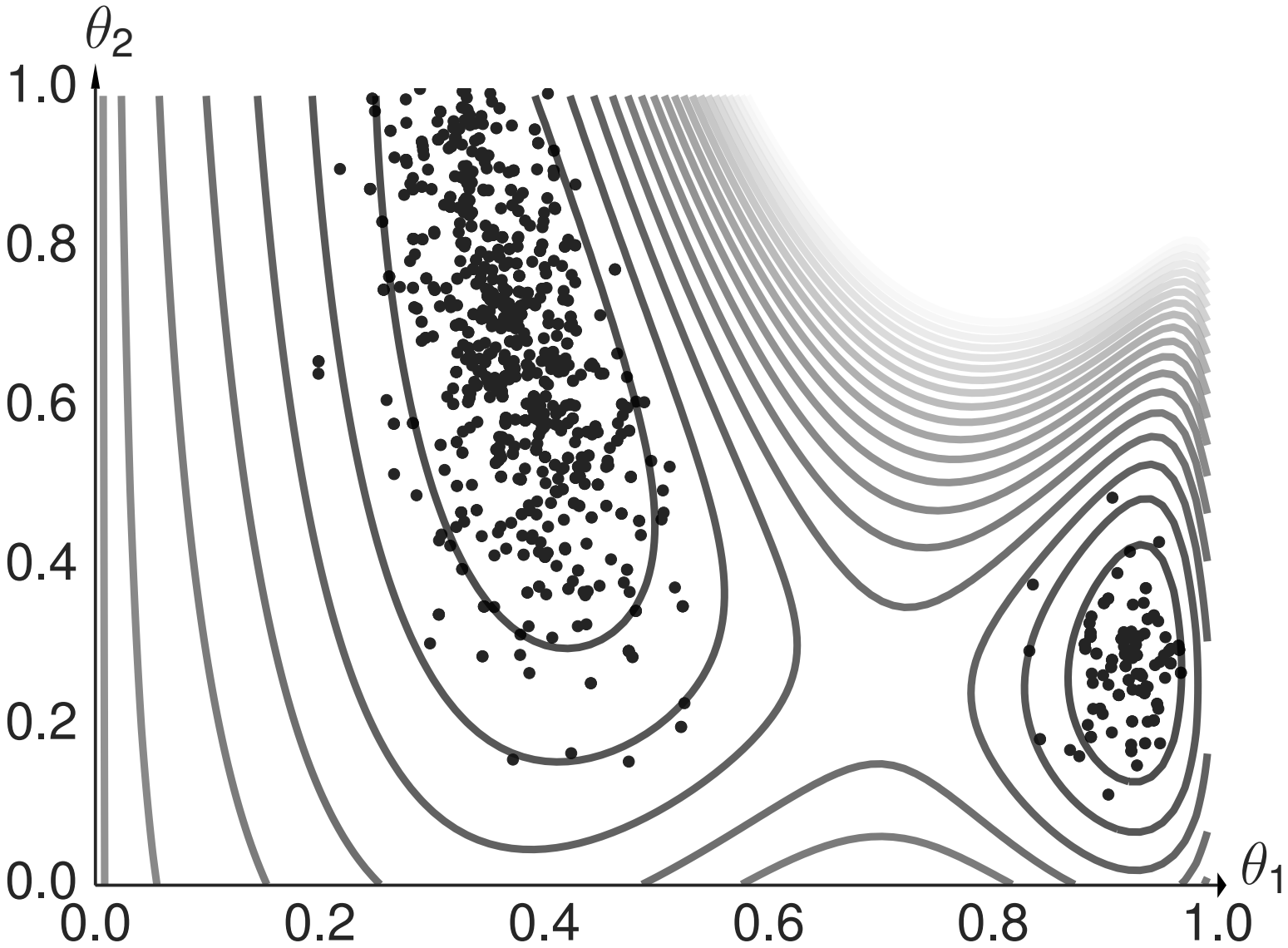
- Innovation: $\epsilon_t \sim iidN(0,1)$.
- Prior: uniform on the square $0 \leq \theta_1 \leq 1$ and $0 \leq \theta_2 \leq 1$.
- Simulate $T = 200$ observations given $\theta = [0.45, 0.45]'$, which is observationally equivalent to $\theta = [0.89, 0.22]'$

Illustration: Tempered Posteriors of θ_1

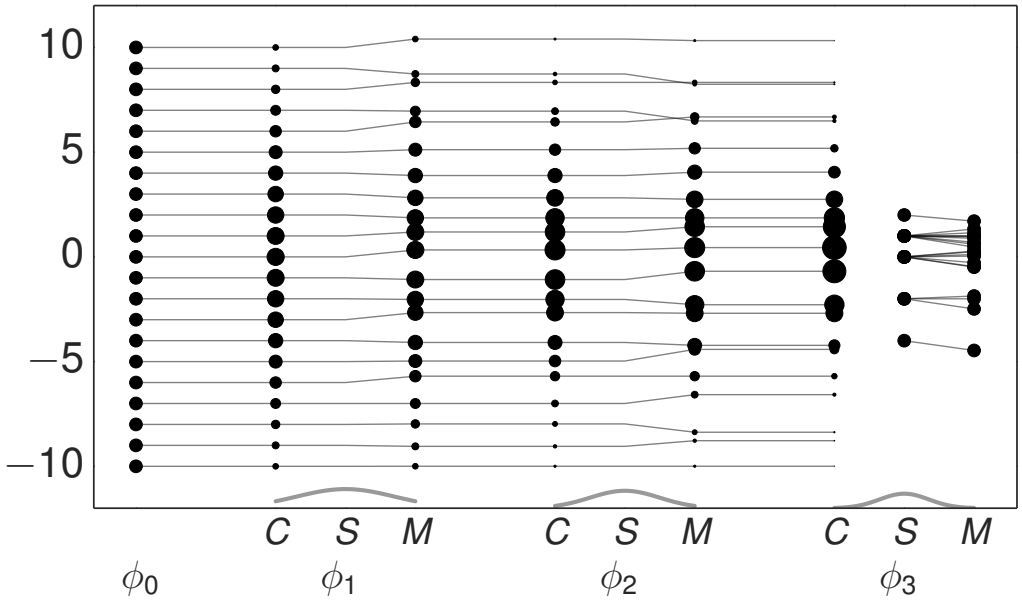


$$\pi_n(\theta) = \frac{[p(Y|\theta)]^{\phi_n} p(\theta)}{\int [p(Y|\theta)]^{\phi_n} p(\theta) d\theta} = \frac{f_n(\theta)}{Z_n}, \quad \phi_n = \left(\frac{n}{N_\phi}\right)^{\lambda}$$

Illustration: Posterior Draws



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.

SMC Algorithm

1. **Initialization.** ($\phi_0 = 0$). Draw the initial particles from the prior: $\theta_1^i \overset{iid}{\sim} p(\theta)$ and $W_1^i = 1$, $i = 1, \ldots, N$.
2. **Recursion.** For $n = 1, \ldots, N_\phi$,
   1. **Correction.** Reweight the particles from stage $n-1$ by defining the incremental weights

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

   and the normalized weights

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

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

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

   2. **Selection.**

SMC Algorithm

1. **Initialization.**
2. **Recursion.** For $n = 1, \ldots, N_\phi$,
   1. **Correction.**
   2. **Selection. (Optional Resampling)** Let $\{\hat{\theta}\}_{i=1}^{N}$ 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

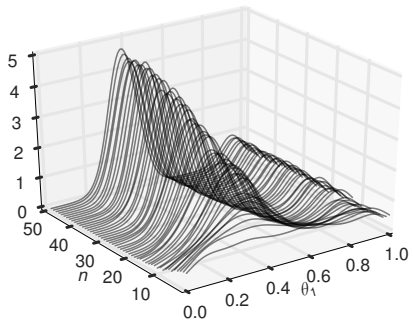
$$\hat{h}_{n,N} = \frac{1}{N} \sum_{i=1}^{N} W_n^i h(\hat{\theta}_n^i). \tag{4}$$

   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

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

Remarks

- 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!



Theoretical Properties

- Goal: strong law of large numbers (SLLN) and central limit theorem (CLT) as $N \longrightarrow \infty$ for every iteration $n = 1, \ldots, N_\phi$.
- 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

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

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

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 reduces persistence of Markov chain;
  - mixture proposal density avoids "getting stuck."
- Blocking: Partition the parameter vector θ_n into N_{blocks} equally sized blocks, denoted by $\theta_{n,b}$, $b = 1, \ldots, N_{blocks}$. (We generate the blocks for $n = 1, \ldots, N_\phi$ randomly prior to running the SMC algorithm.)
- Example: random walk proposal density:

$$\vartheta_b|(\theta^i_{n,b,m-1}, \theta^i_{n,-b,m}, \Sigma^*_{n,b}) \quad \sim \quad N\left(\theta^i_{n,b,m-1}, c_n^2\Sigma^*_{n,b}\right).$$

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

- Infeasible adaption:
  - Let $\Sigma_n^* = \mathbb{V}_{\pi_n}[\theta]$.
  - Adjust scaling factor according to

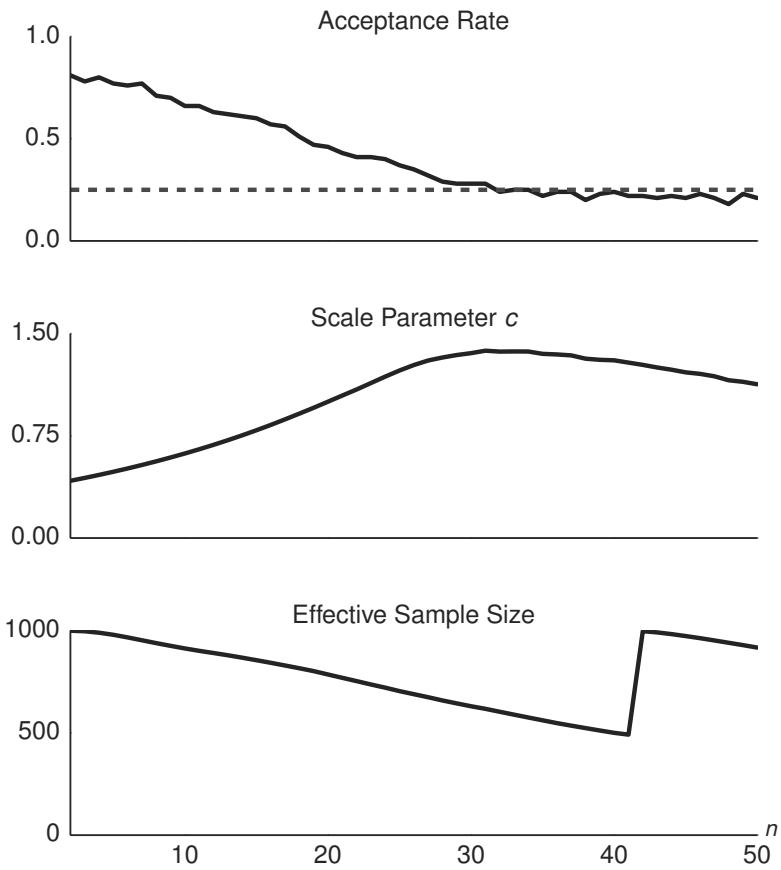
$$c_n = c_{n-1} f\big(R_{n-1}(\zeta_{n-1})\big),$$

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

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

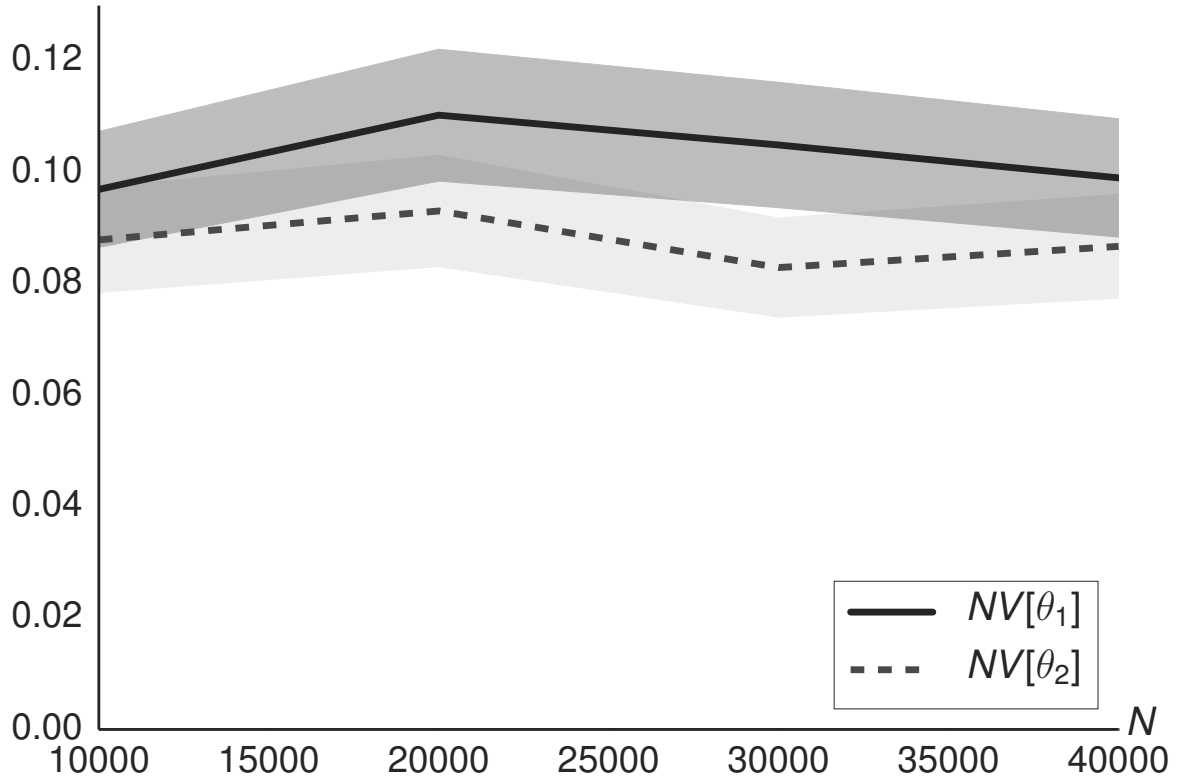
- Feasible adaption – use output from stage $n-1$ to replace ζ_n by $\hat{\zeta}_n$:
  - Use particle approximations of $\mathbb{E}_{\pi_n}[\theta]$ and $\mathbb{V}_{\pi_n}[\theta]$ based on $\{\theta_{n-1}^i, \tilde{W}_n^i\}_{i=1}^N$.
  - Use actual rejection rate from stage $n-1$ to calculate $\hat{c}_n = \hat{c}_{n-1} f\big(\hat{R}_{n-1}(\hat{\zeta}_{n-1})\big)$.
- Result: under suitable regularity conditions replacing ζ_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 $N\mathbb{V}[\bar{\theta}_j]$ 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 $(2 \cdot 1.96)\sqrt{3/N_{run}}(N\mathbb{V}[\bar{\theta}_j])$.

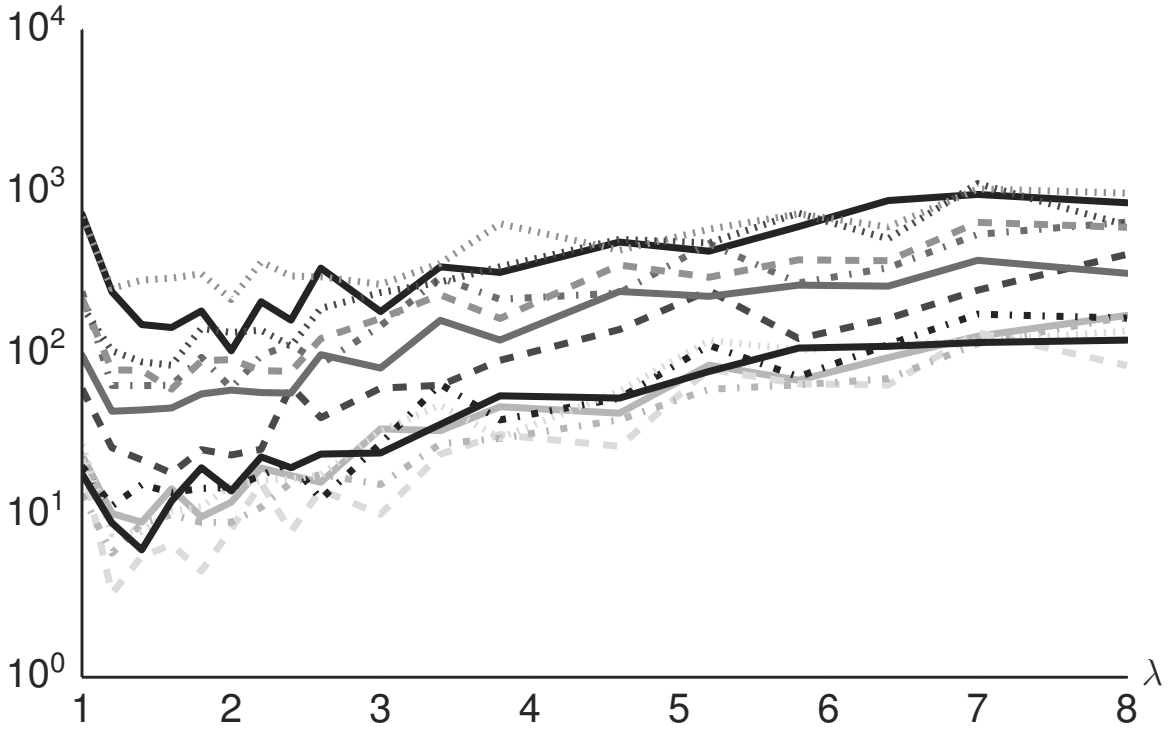
More on Resampling

- So far, we have used *multinomial resampling*. It's fairly intuitive and it is straightforward to obtain a CLT.
- But: *multinominal 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

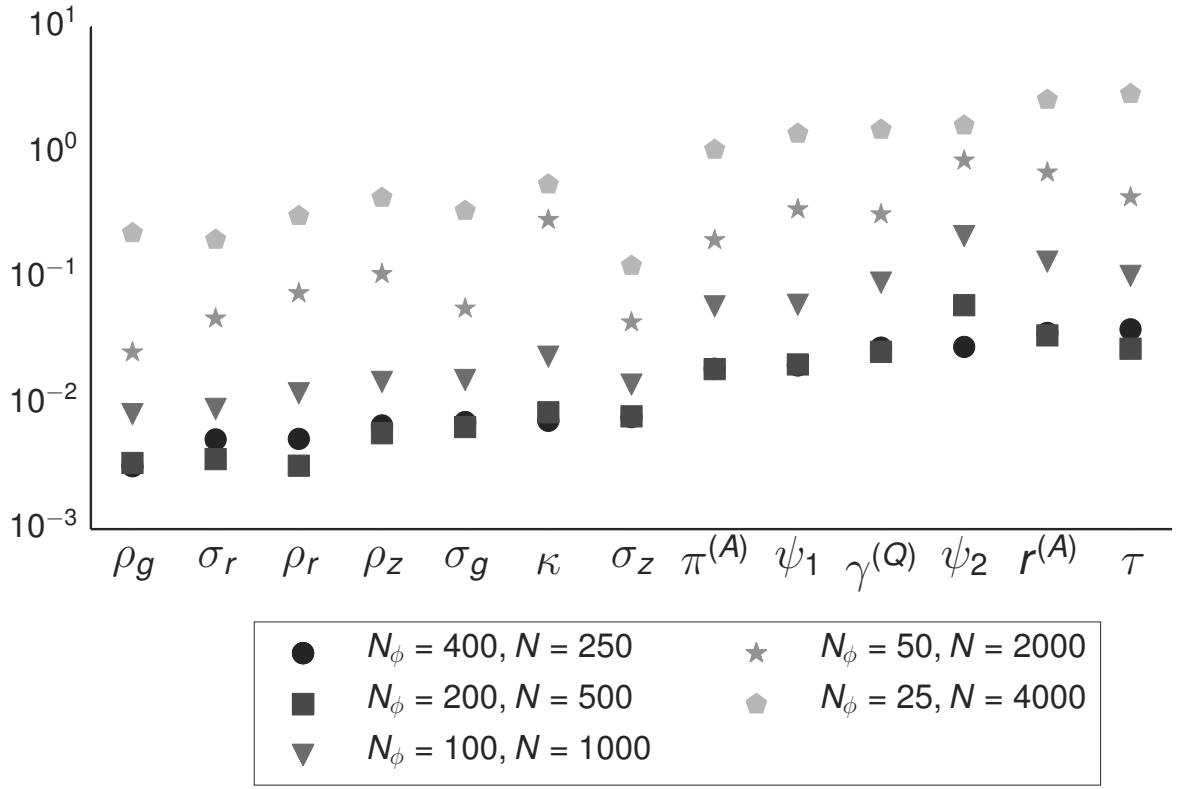
- We will take a look at the effect of various tuning choices on accuracy:
  - Tempering schedule λ: $\lambda = 1$ is linear, $\lambda > 1$ is convex.
  - Number of stages N_ϕ versus number of particles N.
  - Number of blocks in mutation step versus number of particles.

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



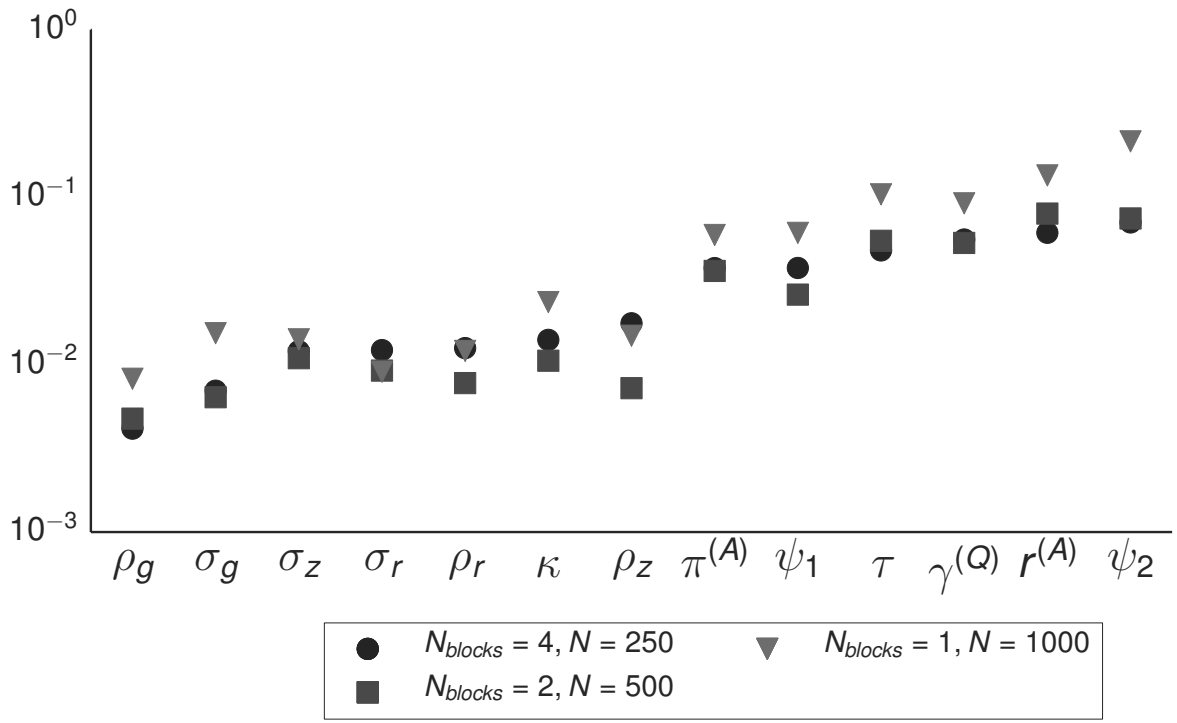
Notes: The figure depicts hairs of $\text{InEff}_N[\bar{\theta}]$ as function of λ. The inefficiency factors are computed based on $N_{run} = 50$ runs of the SMC algorithm. Each hair corresponds to a DSGE model parameter.

Number of Stages N_ϕ 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. $N_{\phi} = 100$, $\lambda = 2$, $N_{MH} = 1$.

A Few Words on Posterior Model Probabilities

- Posterior model probabilities

$$\pi_{i,T} = \frac{\pi_{i,0} p(Y_{1:T}|\mathcal{M}_i)}{\sum_{j=1}^{M} \pi_{j,0} p(Y_{1:T}|\mathcal{M}_j)}$$

  where

$$p(Y_{1:T}|\mathcal{M}_i) = \int p(Y_{1:T}|\theta_{(i)}, \mathcal{M}_i) p(\theta_{(i)}|\mathcal{M}_i) d\theta_{(i)}$$

- For any model:

$$\ln p(Y_{1:T}|\mathcal{M}_i) = \sum_{t=1}^{T} \ln \int p(y_t|\theta_{(i)}, Y_{1:t-1}, \mathcal{M}_i) p(\theta_{(i)}|Y_{1:t-1}, \mathcal{M}_i) d\theta_{(i)}$$

- Marginal data density $p(Y_{1:T}|\mathcal{M}_i)$ arises as a by-product of SMC.

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

	$N_\phi = 100$		$N_\phi = 400$	
N	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 $N_{MH} = 1$.