Sequential Monte Carlo

Ed Herbst

November 13, 2020

MCMC: What works and what doesn't, Simple Model

- State-space representation:

$$y_t = \begin{bmatrix} 1 & 1 \end{bmatrix} s_t, \quad s_t = \begin{bmatrix} \phi_1 & 0 \\ \phi_3 & \phi_2 \end{bmatrix} s_{t-1} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \epsilon_t. \qquad (1)$$

- The state-space model can be re-written as ARMA(2,1) process

$$(1 - \phi_1 L)(1 - \phi_2 L) y_t = (1 - (\phi_2 - \phi_3) L) \epsilon_t.$$

- Relationship between state-space parameters ϕ and structural parameters θ:

$$\phi_1 = \theta_1^2, \quad \phi_2 = (1 - \theta_1^2), \quad \phi_3 - \phi_2 = -\theta_1 \theta_2.$$

Stylized Example

Model

Reduced form: $(1 - \phi_1 L)(1 - \phi_2 L) y_t = (1 - (\phi_2 - \phi_3) L)\epsilon_t$.

Relationship of ϕ and θ: $\phi_1 = \theta_1^2, \quad \phi_2 = (1 - \theta_1^2), \quad \phi_3 - \phi_2 = -\theta_1 \theta_2$.

- *Local* identification problem arises as $\theta_1 \longrightarrow 0$.

- *Global* identification problem $p(Y|\theta) = p(Y|\tilde{\theta})$:

$$\theta_1^2 = \rho, \quad (1 - \theta_1^2) = \theta_1 \theta_2$$

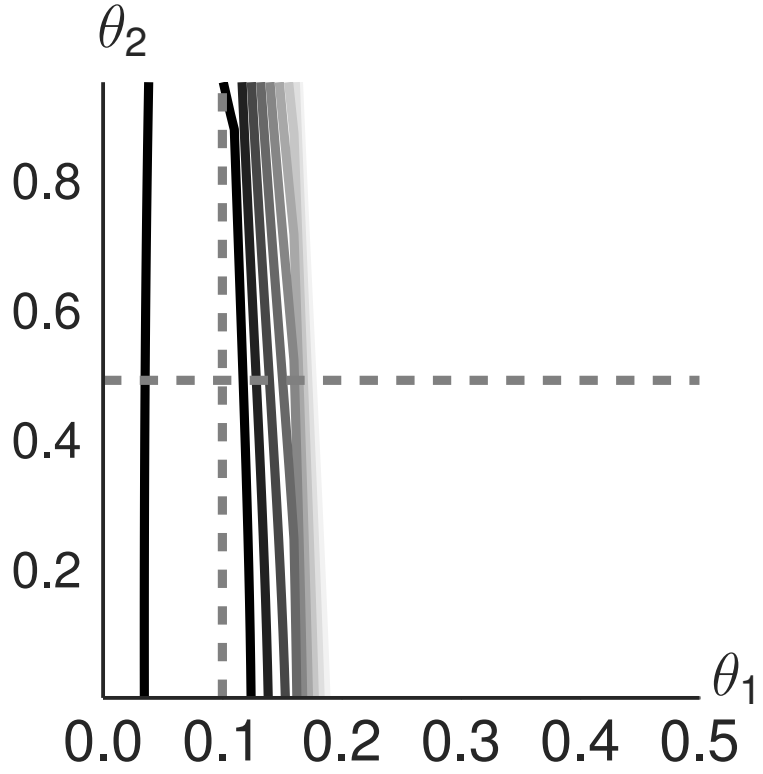
versus

$$\tilde{\theta}_1^2 = 1 - \rho, \quad \tilde{\theta}_1^2 = \tilde{\theta}_1 \tilde{\theta}_2$$

Stylized Example: Likelihood Fcn 100 Obs

Reduced form: $(1 - \phi_1 L)(1 - \phi_2 L) y_t = (1 - (\phi_2 - \phi_3) L) \epsilon_t$.

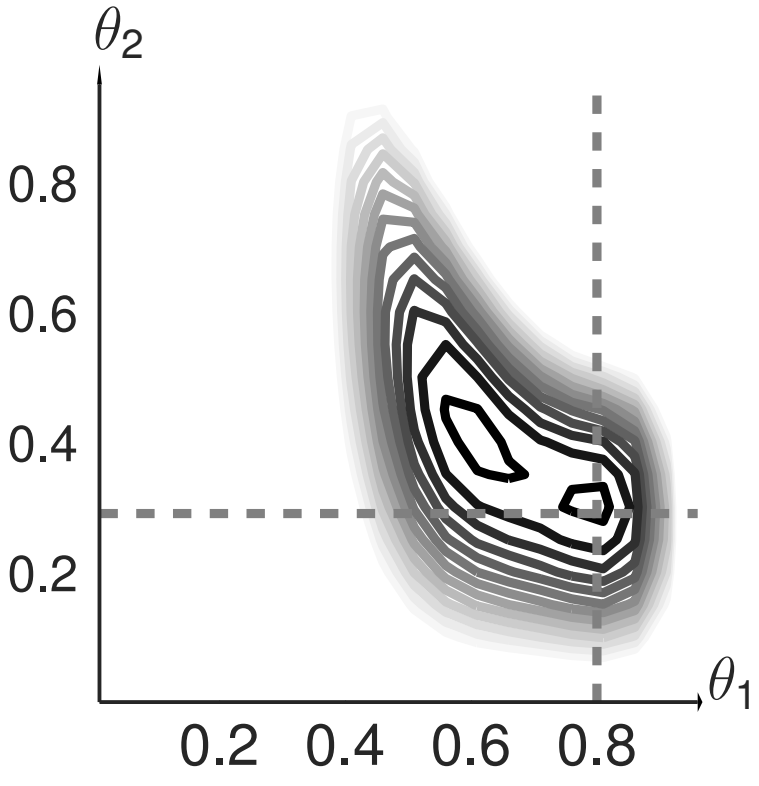
Relationship of ϕ and θ:
$\phi_1 = \theta_1^2, \quad \phi_2 = (1 - \theta_1^2), \quad \phi_3 - \phi_2 = -\theta_1 \theta_2$.



Stylized Example: Likelihood Fcn 500 Obs

Reduced form: $(1 - \phi_1 L)(1 - \phi_2 L) y_t = (1 - (\phi_2 - \phi_3) L) \epsilon_t$.

Relationship of ϕ and θ:
$\phi_1 = \theta_1^2, \quad \phi_2 = (1 - \theta_1^2), \quad \phi_3 - \phi_2 = -\theta_1 \theta_2$.



Introduction

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

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

- "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 $\{\theta^i\}_{i=1}^N$.
- 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 (2010)
  - Curdia and Reis (2010)
  - Herbst (2012)

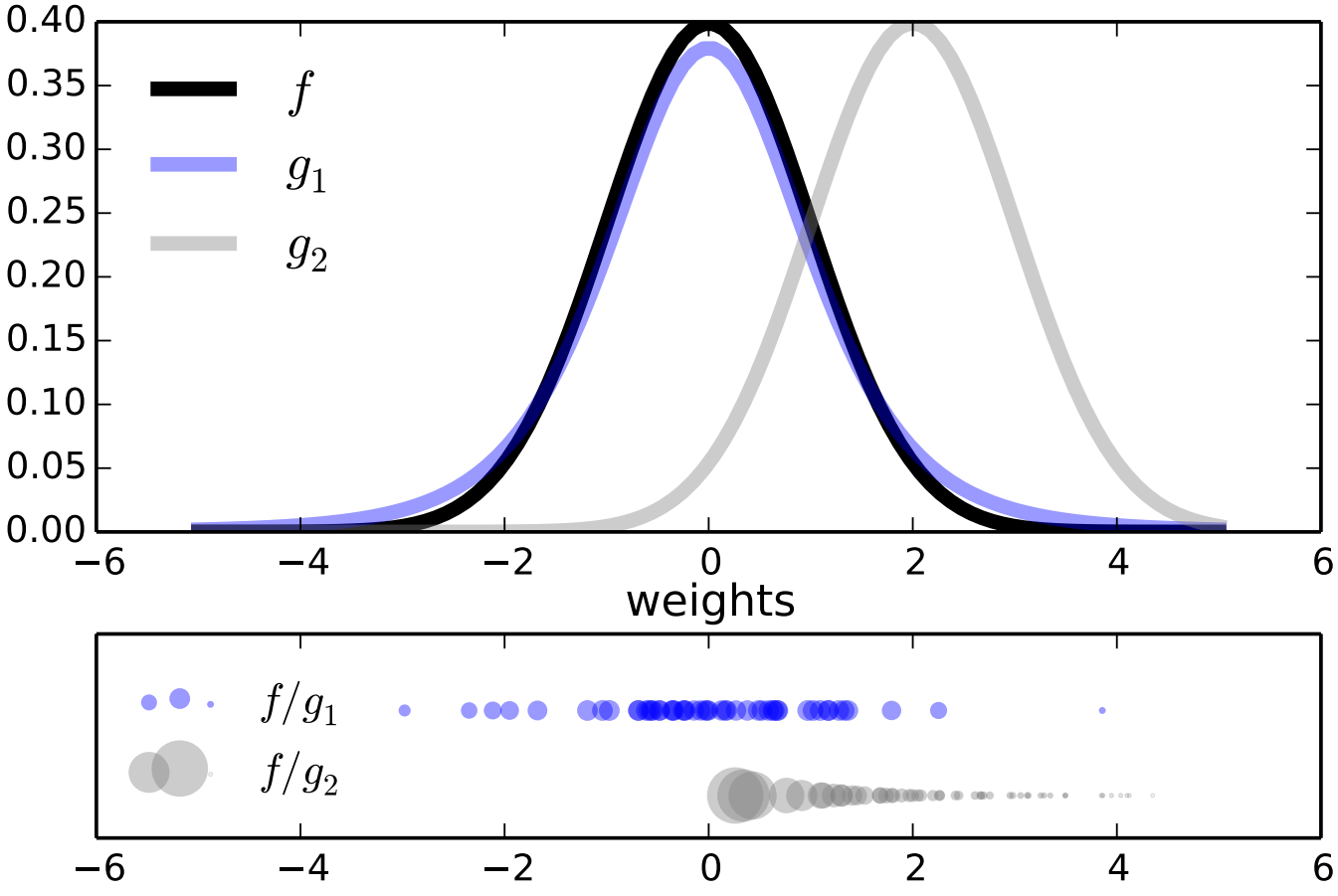
- Now, we use sequential Monte Carlo (SMC) (more precisely, sequential importance sampling) instead:
  - Better suited to handle irregular and multimodal posteriors associated with large DSGE models.
  - Algorithms can be easily parallelized.

- SMC = Importance Sampling on Steriods. We build on
  - Theoretical work: Chopin (2004); Del Moral et al. (2006)
  - Applied work: Creal (2007); Durham and Geweke (2011)

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 densi@@ty for π_n into a proposal density for π_{n+1} and iterate, letting $\phi_n \longrightarrow \phi_N = 1$.

Illustration:

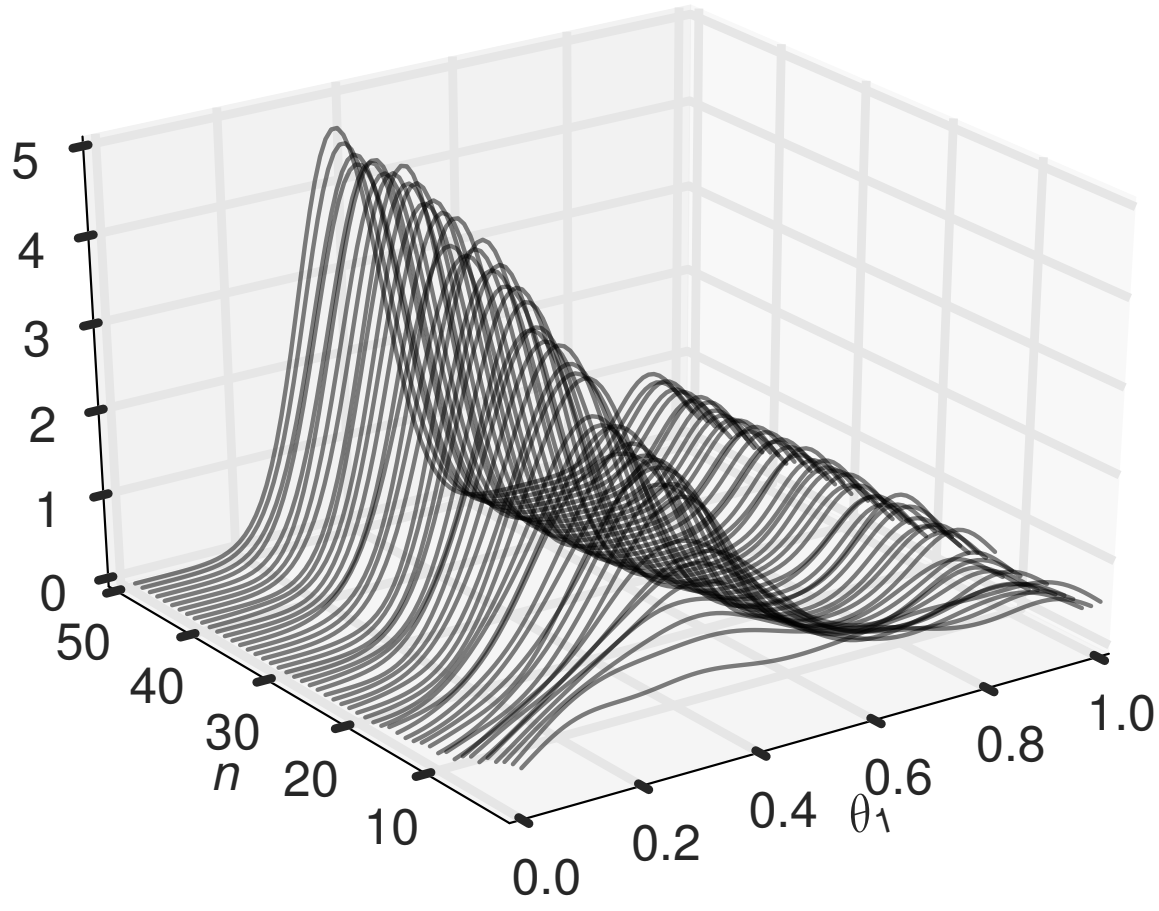
- Our state-space model:

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

- 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

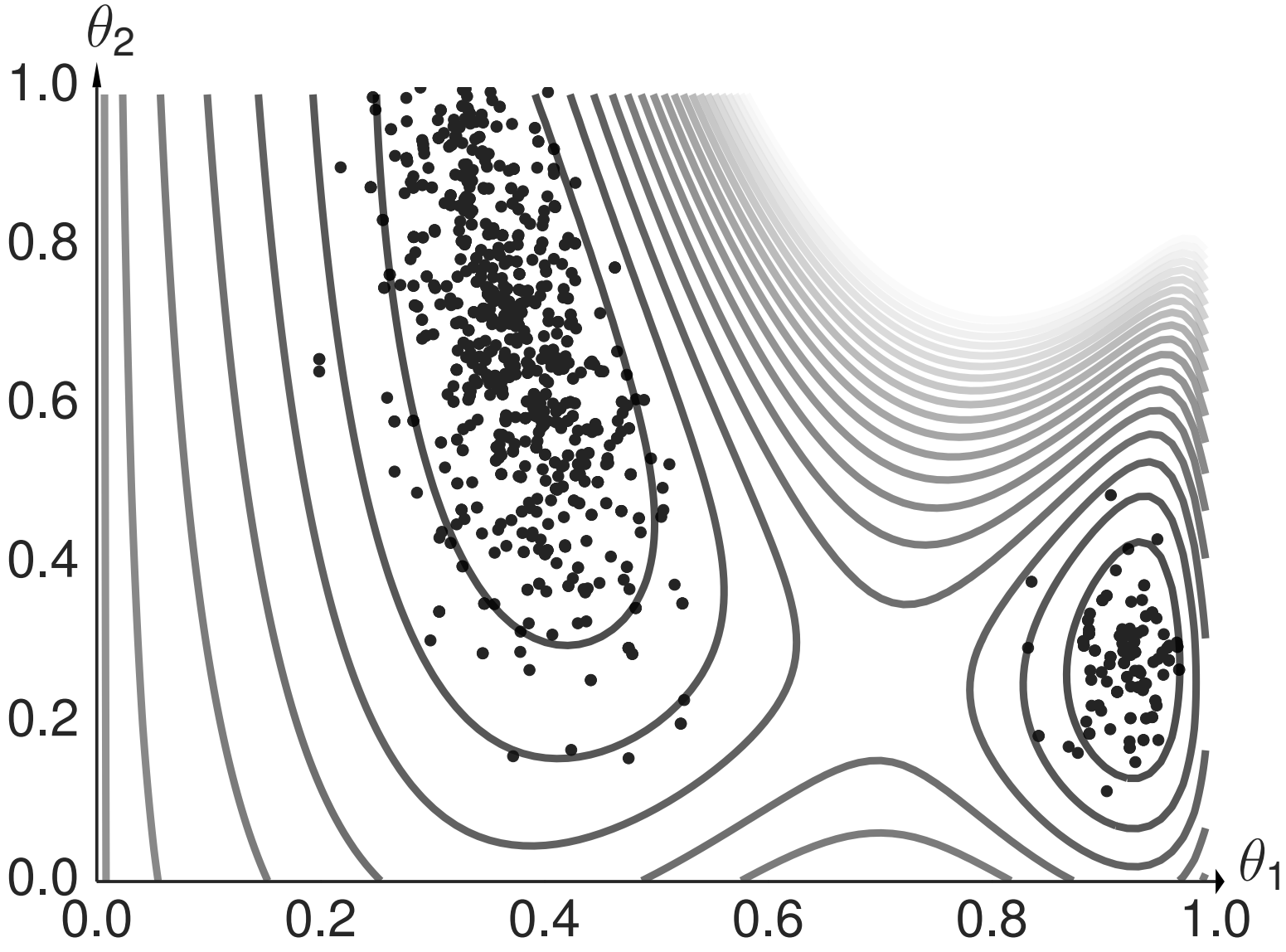
given $\theta = [0.45, 0.45]'$, which is observationally equivalent to $\theta = [0.89, 0.22]'$

Illustration: Tempered Posteriors of θ_1

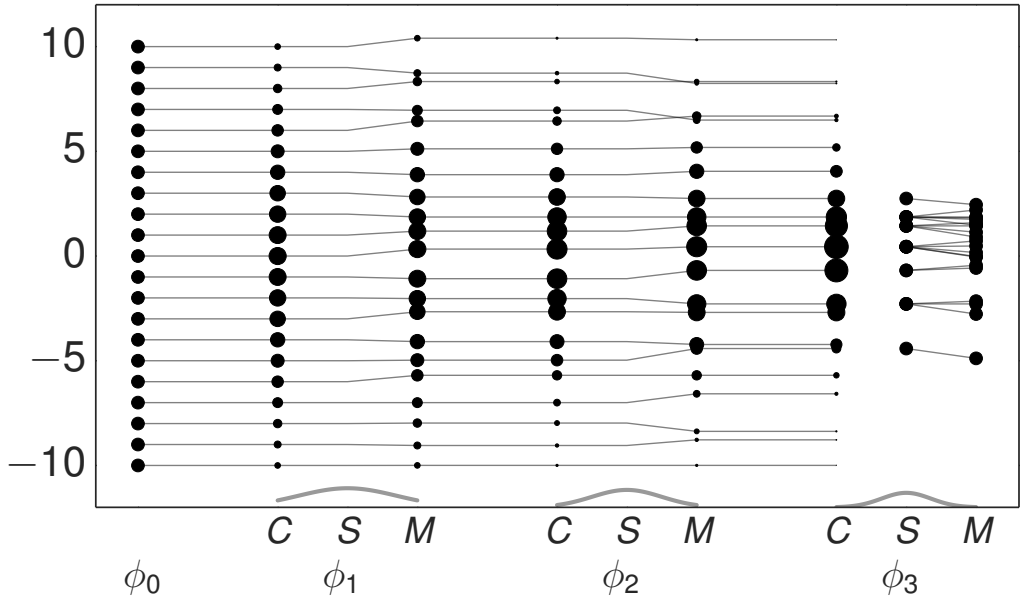


$$\pi_n(\theta) = \frac{[p(Y|\theta)]^{\phi_n} p(\theta)}{\int [p(Y|\theta)]^{\phi_n} p(\theta) d\theta} = \frac{f_n(\theta)}{Z_n}, \quad \phi_n = \left(\frac{n}{N_\phi}\right)^{\lambda}$$

Illustration: Posterior Draws



SMC Algorithm: A Graphical Illustration



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

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

- 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 $n = 1, \ldots, N_\phi$,
   - 2.1 Correction. Reweight the particles from stage $n-1$ by defining the incremental weights

$$\tilde{w}_n^i = [p(Y|\theta_{n-1}^i)]^{\phi_n - \phi_{n-1}} \tag{2}$$

   and the normalized weights

$$\tilde{W}_n^i = \frac{\tilde{w}_n^i W_{n-1}^i}{\frac{1}{N}\sum_{i=1}^N \tilde{w}_n^i W_{n-1}^i}, \quad i = 1, \ldots, N. \tag{3}$$

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

$$\tilde{h}_{n,N} = \frac{1}{N}\sum_{i=1}^N \tilde{W}_n^i h(\theta_{n-1}^i). \tag{4}$$

   - 2.2 Selection.
   - 2.3 Mutation.

SMC Algorithm

1. Initialization.
2. Recursion. For $n = 1, \ldots, N_\phi$,
   - 2.1 Correction.
   - 2.2 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

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

   - 2.3 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

$$\bar{h}_{n,N} = \frac{1}{N} \sum_{i=1}^{N} h(\theta_n^i) W_n^i. \tag{6}$$

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): proceed recursively
  - Initialization: SLLN and CLT for *iid* random variables because we sample from prior.
  - Assume that $n - 1$ approximation (with normalized weights) yields

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

  - Show that

$$\sqrt{N}\left(\frac{1}{N}\sum_{i=1}^{N} h(\theta_n^i) W_n^i - \mathbb{E}_{\pi_n}[h(\theta)]\right) \Longrightarrow N\big(0, \Omega_n(h)\big)$$

Theoretical Properties: Correction Step

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

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

- Then

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

where

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

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

Theoretical Properties: Selection / Resampling

- After resampling by drawing from iid multinomial distribution we obtain

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

  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 $\tilde{\Omega}_{n+1}(h) = \Omega_n\big(v_n(\theta)(h - \mathbb{E}_{\pi_{n+1}}[h])\big)$.

Theoretical Properties: Mutation

- We are using the Markov transition kernel $K_n(\theta|\hat{\theta})$ to transform draws $\hat{\theta}_n^i$ into draws θ_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

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

  - 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 *embarassingly 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 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

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

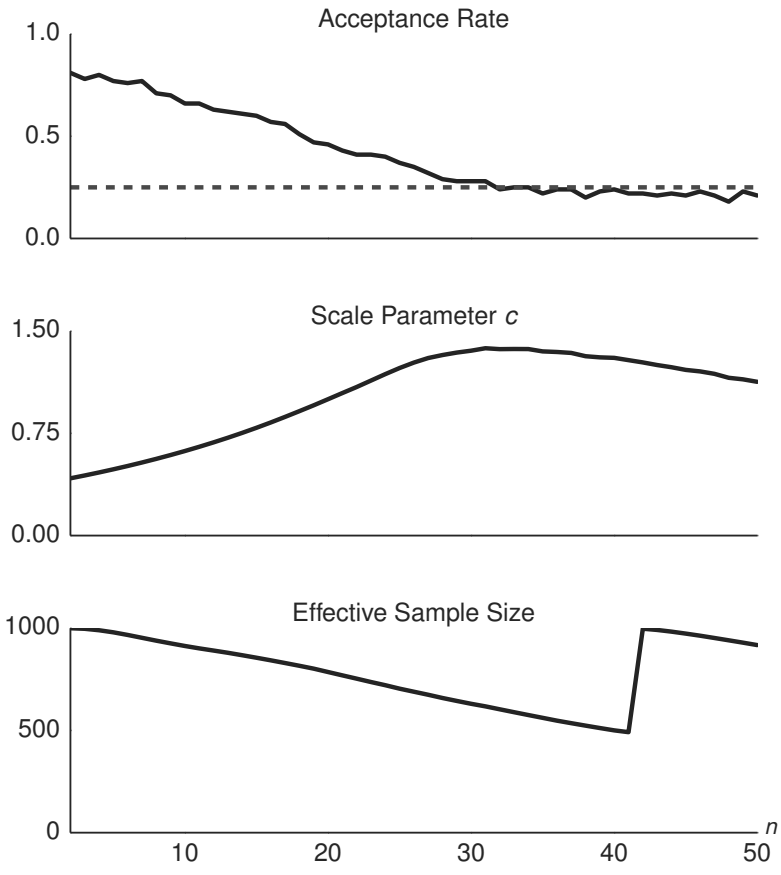
    where $R_{n-1}(\cdot)$ is population rejection rate from iteration $n-1$ and

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

- 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 $\hat{c}_n = \hat{c}_{n-1} f\big(\hat{R}_{n-1}(\hat{\zeta}_{n-1})\big)$.

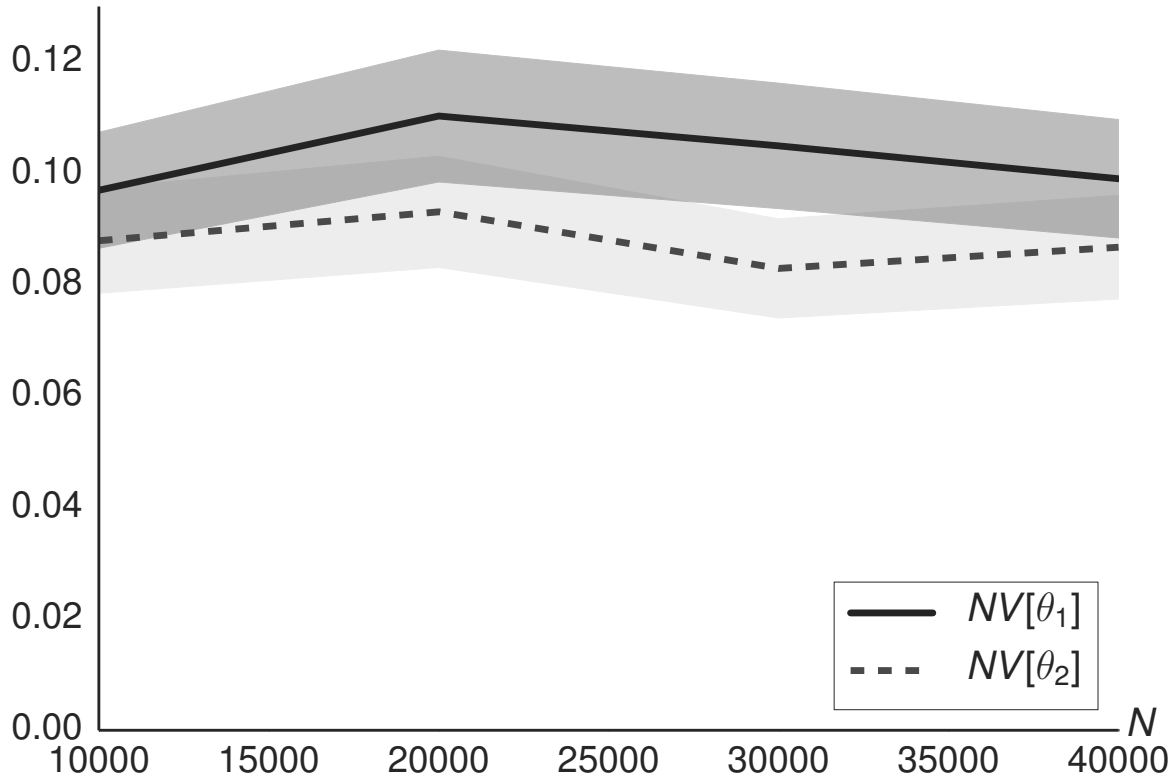
- Result: under suitable regularity conditions replacing ζ_n by $\hat{\zeta}_n$ where $\sqrt{n}(\hat{\zeta}_n - \zeta_n) = O_p(1)$ does not affect the asymptotic variance of the MC approximation.

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.

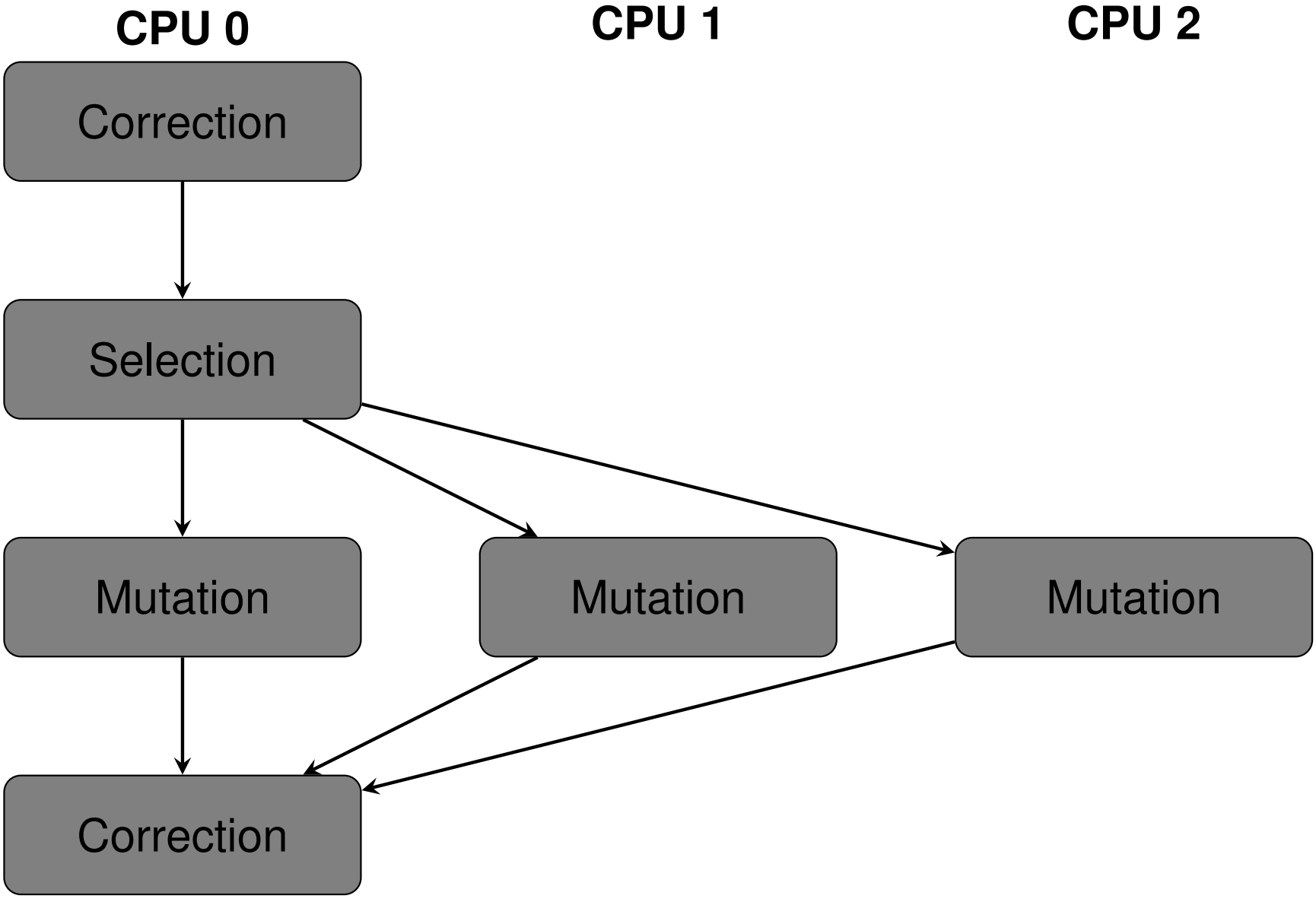
Running Time – It's all about Mutation

- The most time consuming part of (any of) these algorithms, is evaluating the likelihood function, which occurs in the mutation step.
- But each particle is *mutated independently* of the other particles.
- This is extremely easy to parallelize.

How I do it – distributed memory parallelization in `Fortran`

- Use Message Passing Interface (MPI) to scatter particles across many processors (CPUs).
- Execute mutuation across processors.
- Use MPI to gather the newly mutated particles.

Could be better with more programming.


CPU 0
CPU 1
CPU 2
Correction
Selection
Mutation
Mutation
Mutation
Correction

How well does this work?

- The extent to which HPC can help us is determined by the amount of algorithm that can be executed in parallel vs. serial.

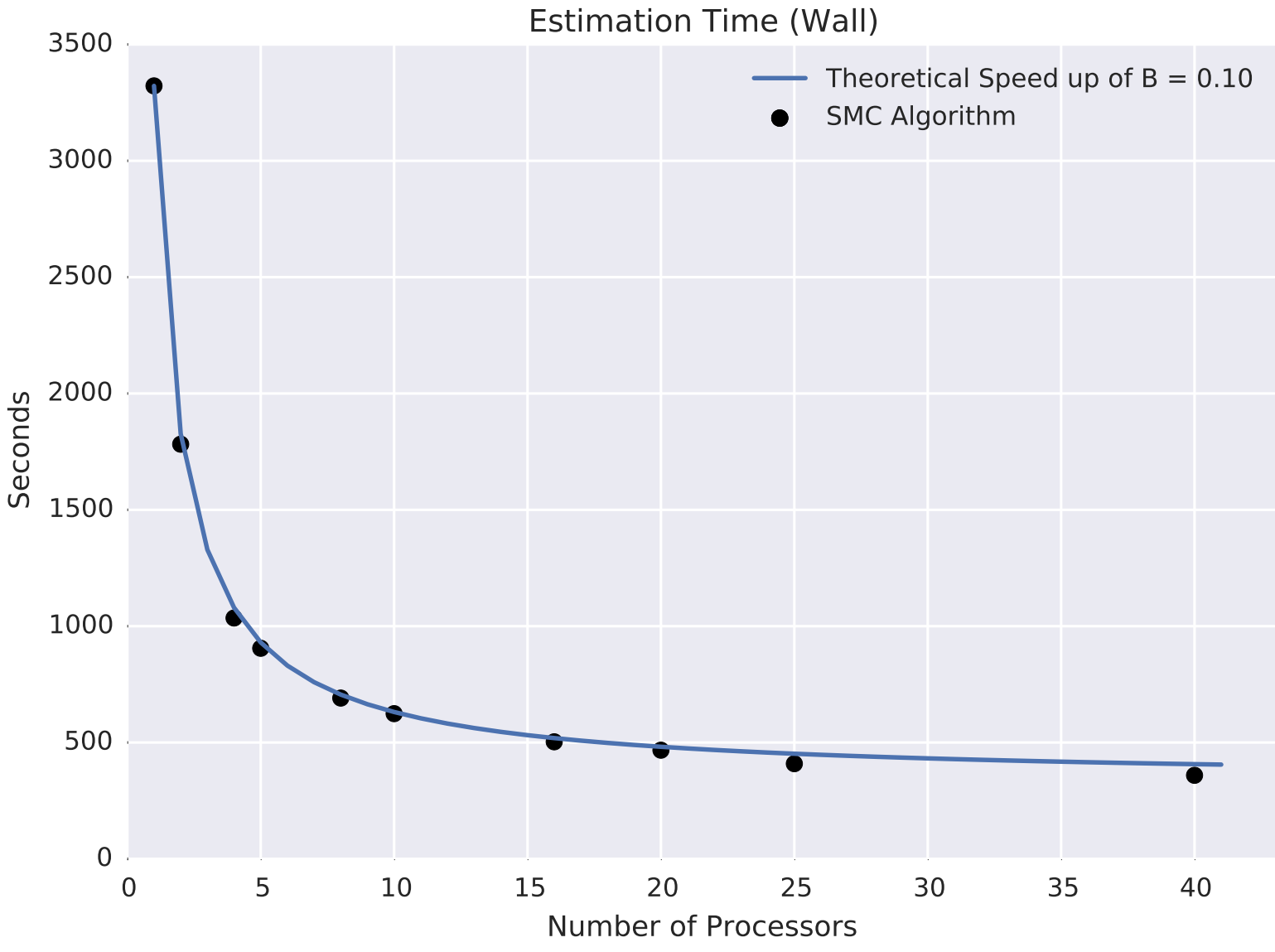
- Suppose a fraction $B \in [0, 1]$ must executed in serial fashion for a particular algorithm.

- Amdahls Law: Theoretical gain from using N processors in an algorithm is given by:

$$R(N) = B + \frac{1}{N}(1 - B)$$

- Question: What is B for our SMC algorithm?
  Answer: about 0.1!

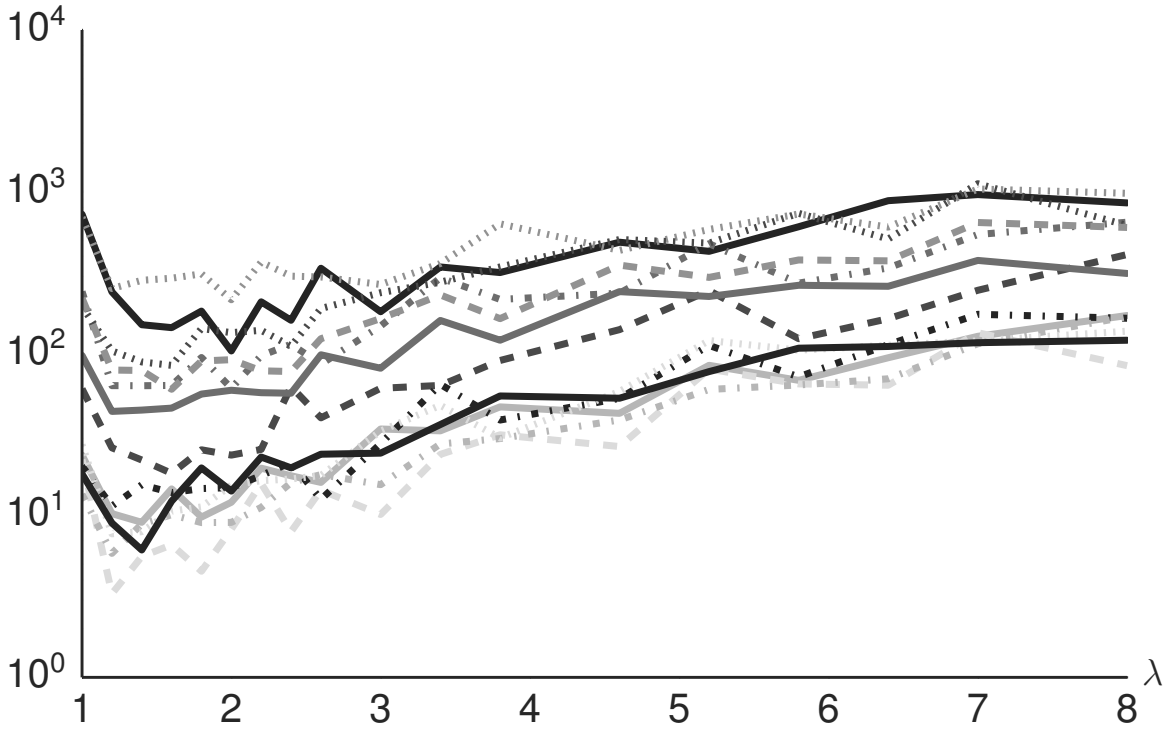
Gains from Parallelization



Application 1: Small Scale New Keynesian Model

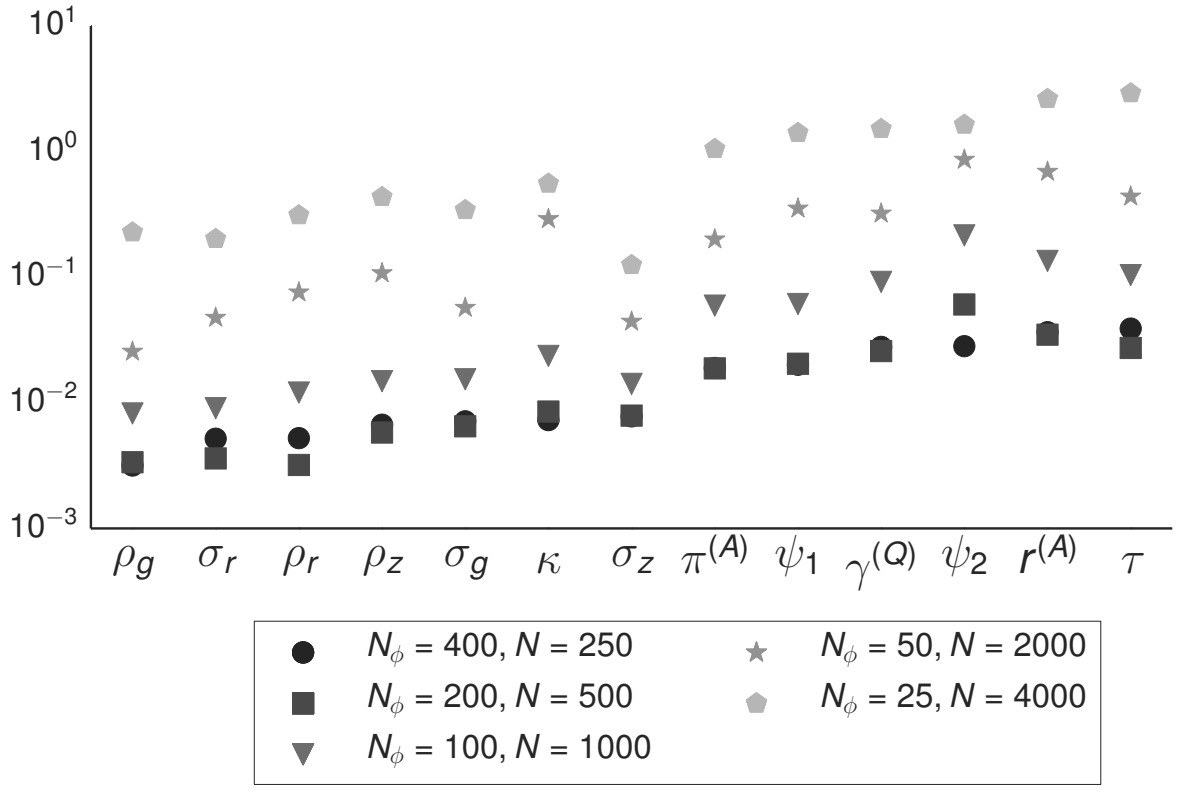
- We will take a look at the effect of various tuning choices on accuracy:
  - Tempering schedule λ: $\lambda = 1$ is linear, $\lambda > 1$ is convex.
  - Number of stages N_ϕ versus number of particles N.
  - Number of blocks in mutation step versus number of particles.

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



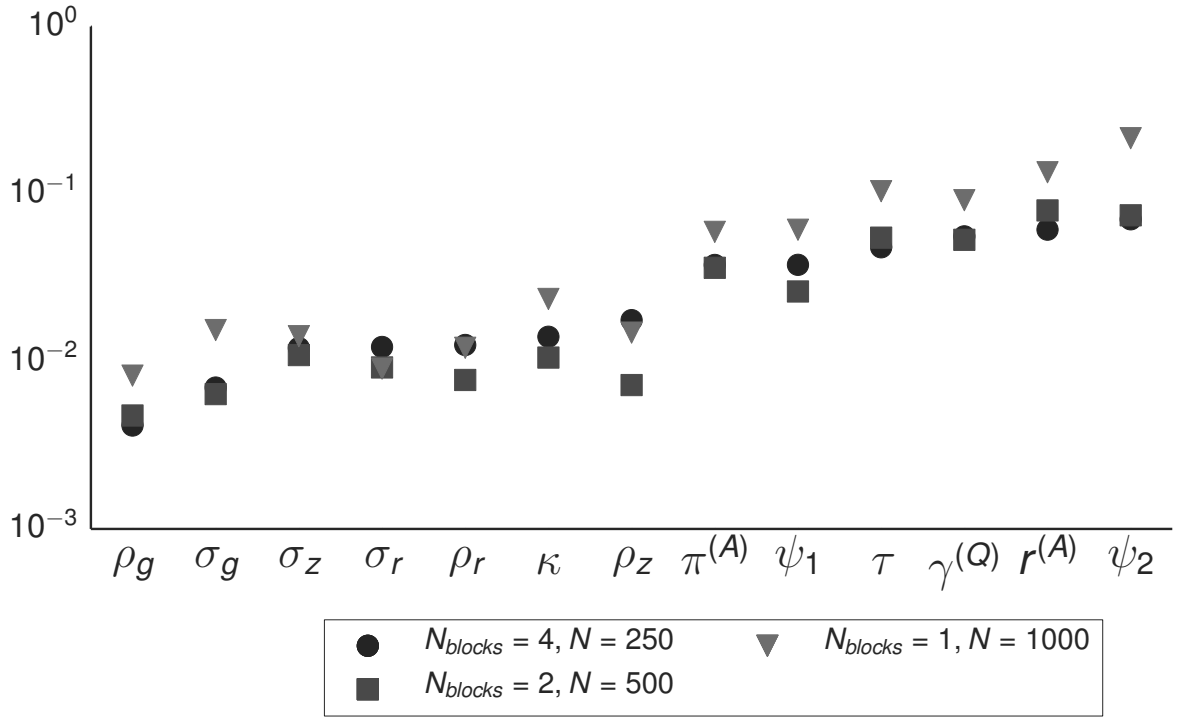
Notes: The figure depicts hairs of $\text{InEff}_N[\bar{\theta}]$ as function of λ. The inefficiency factors are computed based on $N_{run} = 50$ runs of the SMC algorithm. Each hair corresponds to a DSGE model parameter.

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:

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

- 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 $N_{MH} = 1$.

Different Kinds of Tempering

$$\text{Likelihood Tempering: } p_n(Y|\theta) = [p(Y|\theta)]^{\phi_n}, \quad \phi_n \uparrow 1. \tag{7}$$

- Can easily control how "close" consecutive posteriors are to one another.
- Need to pick ϕ_n (though we have some experience).

$$\text{Data Tempering: } p_n(Y|\theta) = p(y_{1:\lfloor \phi_n T \rfloor}), \quad \phi_n \uparrow 1. \tag{8}$$

- Arguably more natural for time series application.
- Typically produces mores inefficient samples of θ.

Cai et al. (2019) generalize *both* likelihood and data tempering!

Generalized Data Tempering

Imagine one has draws from the posterior

$$\tilde{\pi}(\theta) \propto \tilde{p}(\tilde{Y}|\theta)p(\theta), \tag{9}$$

where the posterior $\tilde{\pi}(\theta)$ differs from the posterior $\pi(\theta)$ because of:

1. The sample (Y versus $\tilde{Y}$), or,
2. the model ($p(Y|\theta)$ versus $\tilde{p}(\tilde{Y}|\theta)$), or,
3. of both

Define the stage-n likelihood function:

$$p_n(Y|\theta) = [p(Y|\theta)]^{\phi_n}[\tilde{p}(\tilde{Y}|\theta)]^{1-\phi_n}, \quad \phi_n \uparrow 1. \tag{10}$$

Generalized Data Tempering: SMC that use this likelihood.

Some Comments

$$p_n(Y|\theta) = [p(Y|\theta)]^{\phi_n}[\tilde{p}(\tilde{Y}|\theta)]^{1-\phi_n}$$

1. With $\tilde{p}(\cdot) = 1$: identical to likelihood tempering.

2. With $\tilde{p}(\cdot) = p(\cdot), Y = y_{1:\lfloor \phi_m T \rfloor}$, and $\tilde{Y} = y_{1:\lfloor \phi_{m-1} T \rfloor}$, generalizes data tempering by allowing for a gradual transition between $y_{1:\lfloor \phi_{m-1} T \rfloor}$ and $y_{1:\lfloor \phi_m T \rfloor}$.

3. By allowing Y to differ from $\tilde{Y}$: incorporate data revisions between time $\lfloor \phi_{m-1} T \rfloor$ and $\lfloor \phi_m T \rfloor$.

4. $p(\cdot) \neq \tilde{p}(\cdot)$: one can transition between the posterior distribution of two models with the same parameters.

Evergreen Problem: How to Pick Tuning Parameters:

The SMC algorithm have a number of tuning parameters:

1. Number of Particles N: cite:Chopin2004a provides a CLT for Monte Carlo averages in N.

2. Hyperparameters determining mutation phase.

3. The number of stages, N_ϕ and the schedule $\{\phi_n\}_{n=1}^{N}$.

This paper: choose ϕ_n adaptively, with no fixed N_ϕ.

Key idea: choose ϕ_n to target a desired level $\widehat{ESS}_n^*$.

the closer the desired $\widehat{ESS}_n^*$ to the previous $\widehat{ESS}_{n-1}$, the smaller the increment $\phi_n - \phi_{n-1}$

An implementation of this

$$w^i(\phi) = [p(Y|\theta^i_{n-1})]^{\phi-\phi_{n-1}}, \quad W^i(\phi) = \frac{w^i(\phi)W^i_{n-1}}{\frac{1}{N}\sum_{i=1}^{N} w^i(\phi)W^i_{n-1}},$$

$$\widehat{ESS}(\phi) = N \Big/ \left(\frac{1}{N}\sum_{i=1}^{N}(W^i_n(\phi))^2\right)$$

We will choose ϕ to target a desired level of ESS:

$$f(\phi) = \widehat{ESS}(\phi) - \alpha\widehat{ESS}_{n-1} = 0, \tag{11}$$

where α (≤ 1) is a tuning constant:

- everything about the tempering is summarized in α
- closer α is to 1, the smaller the desired ESS reduction
- No fixed runtime!

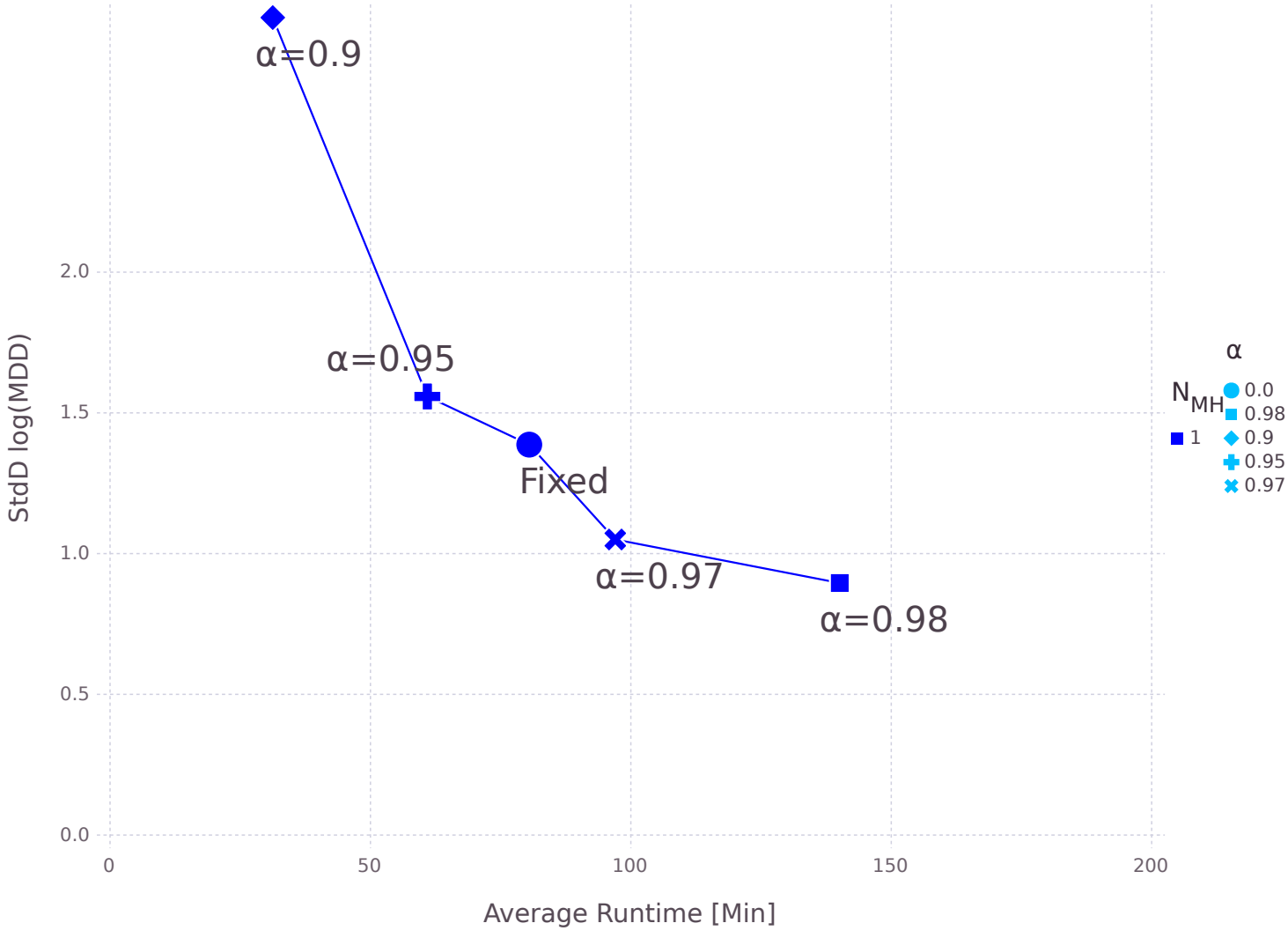
Assessing α.

In time-honored tradition of macroeconometrics, let's estimate the Smets and Wouters (2007) model.

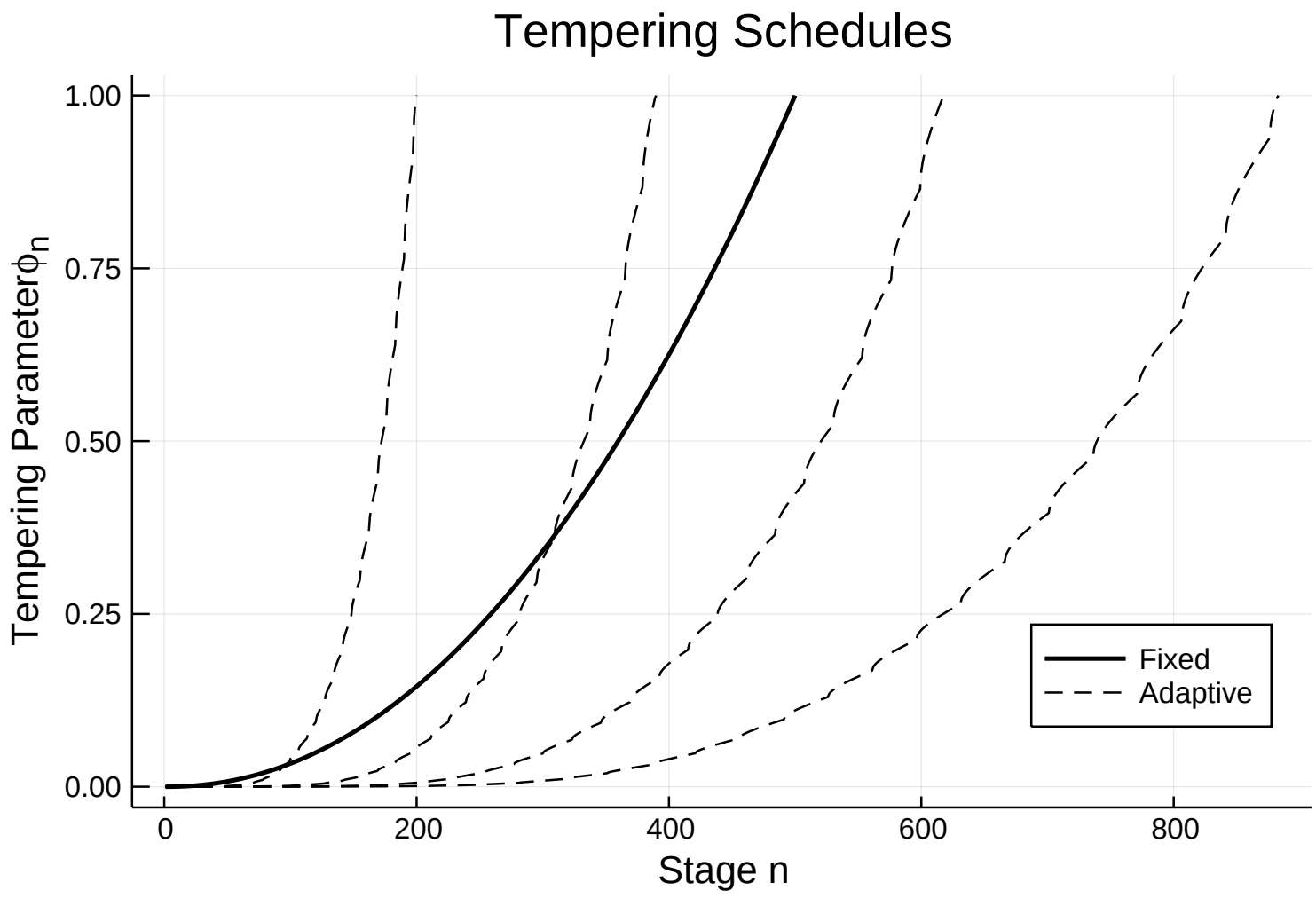
Compare the accuracy (and precision) of SMC algorithm versus the speed:

- $\alpha \in \{0.9, 0.95, 0.97, 0.98\}$.
- Fixed tempering schedule, $N_\phi = 500$, from Herbst and Schorfheide (2014).
- Measure of accuracy: *std of log MDD*, computed across 50 runs of SMC algorithm.
- measure of speed: *average* runtime across these runs

Trade-Off Between Runtime and Accuracy



Tempering Schedules



Some Comments

- Time-accuracy curve is convex:
  - $\alpha = 0.90 \rightarrow \alpha = 0.95$ generates a drastic increase in accuracy, while doubling runtime.
  - $\alpha = 0.97 \rightarrow \alpha = 0.98$ not much increase in accuracy, with substantial increase in runtime.
- Fixed schedule is slightly inefficient.

- All of the adaptive schedules are convex.
- Very little information (relative to fixed schedule) are added to likelihood function initially.
- Towards the end, a lot of information is added.

Illustration of Generalized Data Tempering

Scenario 1:

- partition the sample into two subsamples: $t = 1, \ldots, T_1$ and $t = T_1 + 1, \ldots, T$
- allow for data revisions by the statistical agencies between periods $T_1 + 1$ and T.
- Assume that the second part of the sample becomes available after the model has been estimated on the first part of the sample using the data vintage available at the time, $\tilde{y}_{1:T_1}$.
- In period T we already have a swarm of particles $\{\theta^i_{T_1}, W^i_{T_1}\}_{i=1}^N$ that approximates the posterior

$$p(\theta|\tilde{y}_{1:T_1}) \propto p(\tilde{y}_{1:T_1}|\theta)p(\theta).$$

More Details

Let $Y = y_{1:T}$ and $\tilde{Y} = \tilde{y}_{1:T_1}$, define the stage (n) posterior:

$$\pi_n(\theta) = \frac{p(y_{1:T}|\theta)^{\phi_n} p(\tilde{y}_{1:T_1}|\theta)^{1-\phi_n} p(\theta)}{\int p(y_{1:T}|\theta)^{\phi_n} p(\tilde{y}_{1:T_1}|\theta)^{1-\phi_n} p(\theta) d\theta}.$$

The incremental weights are given by

$$\tilde{w}_n^i(\theta) = p(y_{1:T}|\theta)^{\phi_n - \phi_{n-1}} p(\tilde{y}_{1:T_1}|\theta)^{\phi_{n-1} - \phi_n}$$

Define the Conditional Marginal Data Density (CMDD)

$$\text{CMDD}_{2|1} = \prod_{n=1}^{N_\phi} \left(\frac{1}{N} \sum_{i=1}^{N} \tilde{w}_{(n)}^i W_{(n-1)}^i \right) \tag{12}$$

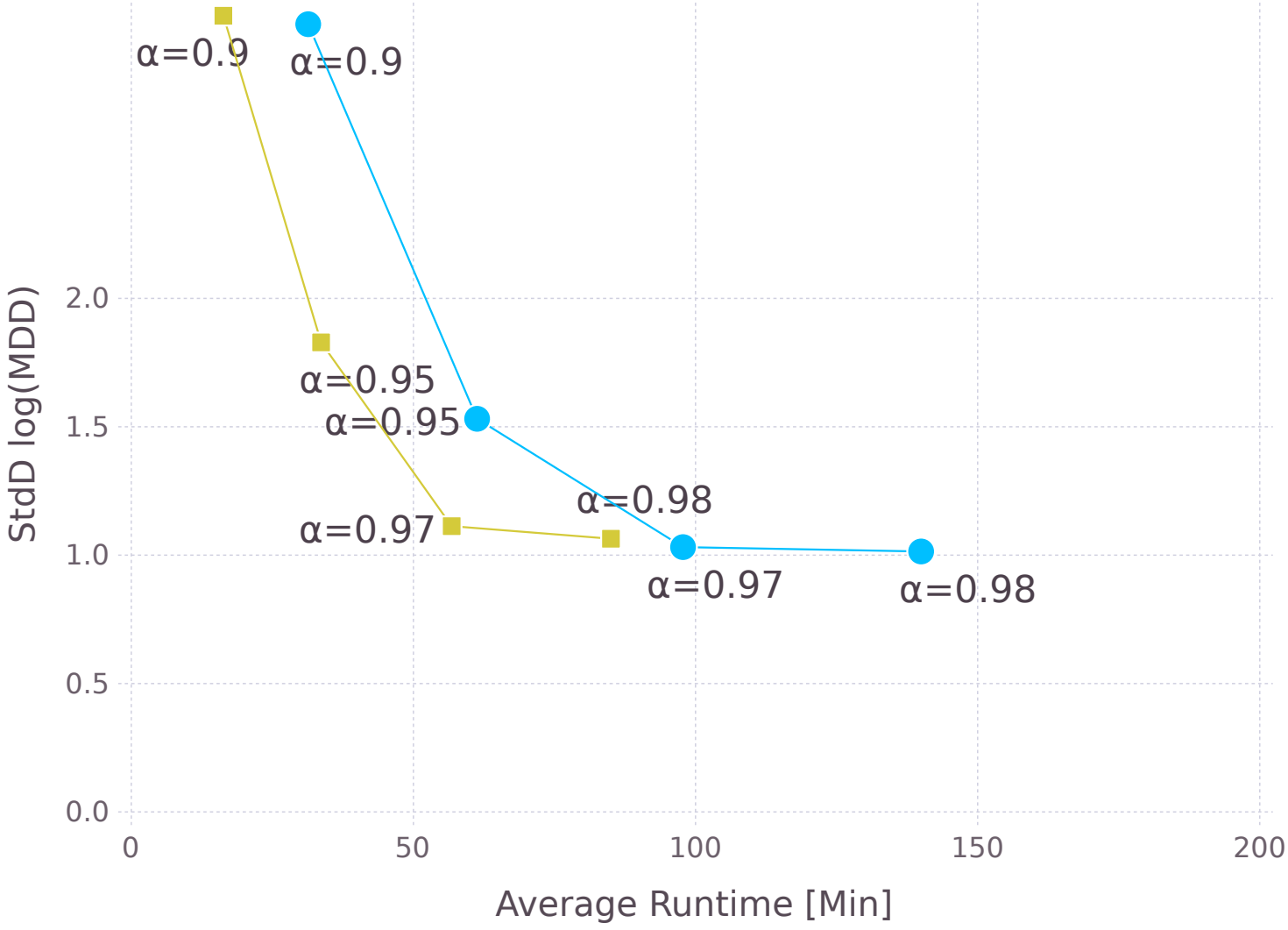
Thus:

$$\text{CMDD}_{2|1} \approx \frac{\int p(y_{1:T}|\theta) p(\theta) d\theta}{\int p(\tilde{y}_{1:T_1}|\theta) p(\theta) d\theta} = \frac{p(y_{1:T})}{p(\tilde{y}_{1:T_1})}. \tag{13}$$

An experiment

- We assume that the DSGE model has been estimated using likelihood tempering based on the sample $y_{1:T_1}$, where $t = 1$ corresponds to 1966:Q4 and $t = T_1$ corresponds to 2007:Q1.
- The second sample, $y_{T_1+1:T}$, starts in 2007:Q2 and ends in 2016:Q3.
- Compare two estimates of MDD
  - Full Sample Likelihood: likelihood-tempering-based estimates using the full sample.
  - GDT: $\log p(y_{1:T_1}) + \log CMDD_{2|1}$.
- Arguably stacked against GDT!

Trade-Off Between Runtime and Accuracy



References I

CAI, M., M. DEL NEGRO, E. HERBST, E. MATLIN, R. SARFATI, AND F. SCHORFHEIDE (2019): "Online Estimation of DSGE Models," *SSRN Electronic Journal*.

CHIB, S. AND S. RAMAMURTHY (2010): "Tailored Randomized Block MCMC Methods with Application to DSGE Models," *Journal of Econometrics*, 155, 19–38.

CHOPIN, N. (2004): "A Sequential Particle Filter for Static Models," *Biometrika*, 89, 539–551.

CREAL, D. (2007): "Sequential Monte Carlo Samplers for Bayesian DSGE Models," *Unpublished Manuscript, Vrije Universitiet*.

CURDIA, V. AND R. REIS (2010): "Correlated Disturbances and U.S. Business Cycles," *Manuscript, Columbia University and FRB New York*.

References II

DEL MORAL, P., A. DOUCET, AND A. JASRA (2006): "Sequential Monte Carlo Samplers," *Journal of the Royal Statistical Society, Series B*, 68, 411–436.

DURHAM, G. AND J. GEWEKE (2011): "Massively Parallel Sequential Monte Carlo Bayesian Inference," *Unpublished Manuscript*.

HERBST, E. (2012): "Gradient and Hessian-based MCMC for DSGE Models," Unpublished Manuscript, Federal Reserve Board.

HERBST, E. AND F. SCHORFHEIDE (2014): "Sequential Monte Carlo Sampling for DSGE Models," *Journal of Applied Econometrics*, 29, 1073 1098.

KOHN, R., P. GIORDANI, AND I. STRID (2010): "Adaptive Hybrid Metropolis-Hastings Samplers for DSGE Models," *Working Paper*.

References III

SMETS, F. AND R. WOUTERS (2007): "Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach," *American Economic Review*, 97, 586–608.