Exact Approximation of Monte Carlo Filters

Adam M. Johansen[⋆], Arnaud Doucet[‡] and Nick Whiteley[†]

[⋆]University of Warwick Department of Statistics,
[†]University of Bristol School of Mathematics,
[‡]University of Oxford Department of Statistics

a.m.johansen@warwick.ac.uk
http://go.warwick.ac.uk/amjohansen/talks

October 11th, 2012
SMC Methods and Efficient Simulation in Finance

Context & Outline

Filering in State-Space Models:

- SIR Particle Filters [GSS93]
- Rao-Blackwellized Particle Filters [AD02, CL00]
- Block-Sampling Particle Filters [DBS06]

Exact Approximation of Monte Carlo Algorithms:

- Particle MCMC [ADH10]

Approximating the RBPF

- Approximated Rao-Blackwellized Particle Filters [CSOL11]
- Exactly-approximated RBPFs [JWD12]

Approximating the BSPF

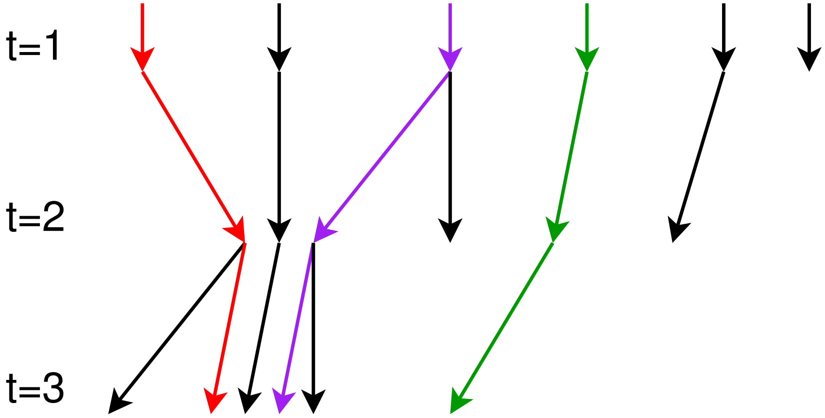
- Local SMC [JD12]

Particle MCMC

- MCMC algorithms which employ SMC proposals [ADH10]
- SMC algorithm as a collection of RVs
  - Values
  - Weights
  - Ancestral Lines
- Construct MCMC algorithms:
  - With many auxiliary variables
  - *Exactly* invariant for distribution on extended space
  - Standard MCMC arguments justify strategy

- Does this suggest anything about SMC?
- Can something similar help with smoothing?

Ancestral Trees



$a_3^1 = 1$ $a_3^4 = 3$ $a_2^1 = 1$ $a_2^4 = 3$

$b_{3,1:3}^2 = (1, 1, 2)$ $b_{3,1:3}^4 = (3, 3, 4)$ $b_{3,1:3}^6 = (4, 5, 6)$

SMC Distributions

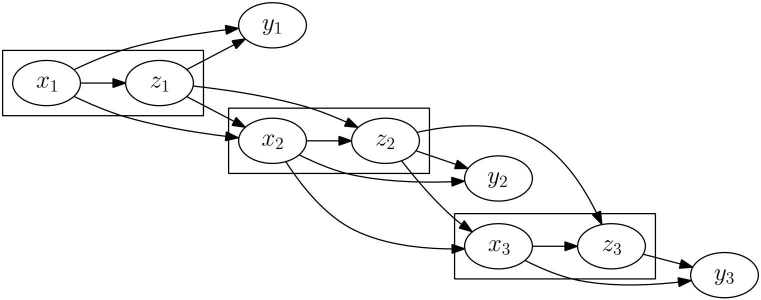
We'll need the **SMC Distribution**:

$$\psi^{M}_{n,L}\left(\overline{\mathbf{a}}_{n-L+2:n}, \overline{\mathbf{x}}_{n-L+1:n}, \overline{k}; x_{n-L}\right)$$
$$= \left[\prod_{i=1}^{M} q(\overline{x}^{i}_{n-L+1} \,|\, \overline{x}_{n-L})\right] \prod_{p=n-L+2}^{n} \left[r(\overline{\mathbf{a}}_p | \overline{\mathbf{w}}_{p-1}) \prod_{i=1}^{M} q\left(\overline{x}^{i}_{p} | \overline{x}^{\overline{a}^{i}_{p}}_{p-1}\right)\right] r(\overline{k} | \overline{\mathbf{w}}_{\mathbf{n}})$$

and the **conditional SMC Distribution**:

$$\widetilde{\psi}^{M}_{n,L}\left(\widetilde{\mathbf{a}}^{\ominus k}_{n-L+2:n}, \widetilde{\mathbf{x}}^{\ominus k}_{n-L+1:n}; x_{n-L} \,\Big|\Big|\, \widetilde{b}^{k}_{n-L+1:n-1}, k, \widetilde{x}^{k}_{n-L+1:n}\right)$$
$$= \frac{\psi^{M}_{n,L}(\widetilde{\mathbf{a}}_{n-L+2:n}, \widetilde{\mathbf{x}}_{n-L+1:n}, k; x_{n-L})}{q\left(\widetilde{x}^{\widetilde{b}^{k}_{n,n-L+1}}_{n-L+1} | x_{n-L}\right) \left[\prod_{p=n-L+2}^{n} r\left(\widetilde{b}^{k}_{n,p} | \widetilde{\mathbf{w}}_{\mathbf{p-1}}\right) q\left(\widetilde{x}^{\widetilde{b}^{k}_{n,p}}_{p} | \widetilde{x}^{\widetilde{b}^{n}_{n,p-1}}_{p-1}\right)\right] r(k | \widetilde{\mathbf{w}}_n)}$$

A (Rather Broad) Class of Hidden Markov Models



- Unobserved Markov chain $\{(X_n, Z_n)\}$ transition f.
- Observed process $\{Y_n\}$ conditional density g.
- Density:

$$p(x_{1:n}, z_{1:n}, y_{1:n}) = f_1(x_1, z_1) g(y_1 | x_1, z_1) \prod_{i=2}^{n} f(x_i, z_i | x_{i-1}, z_{i-1}) g(y_i | x_i, z_i).$$

Formal Solutions

- Filtering and Prediction Recursions:

$$p(x_n, z_n | y_{1:n}) = \frac{p(x_n, z_n | y_{1:n-1}) g(y_n | x_n, z_n)}{\int p(x_n', z_n' | y_{1:n-1}) g(y_n | x_n', z_n') d(x_n', z_n')}$$

$$p(x_{n+1}, z_{n+1} | y_{1:n}) = \int p(x_n, z_n | y_{1:n}) f(x_{n+1}, z_{n+1} | x_n, z_n) d(x_n, z_n)$$

- Smoothing:

$$p((x,z)_{1:n} | y_{1:n}) \propto p((x,z)_{1:n-1} | y_{1:n-1}) f((x,z)_n | (x,z)_{n-1}) g(y_n | (x,z)_n)$$

A Simple SIR Filter

Algorithmically, at iteration n:

- Given $\{W_{n-1}^i, (X,Z)_{1:n-1}^i\}$ for $i = 1, \ldots, N$:
- **Resample**, obtaining $\{\frac{1}{N}, (\widetilde{X}, \widetilde{Z})_{1:n-1}^i\}$.
    - Sample $(X,Z)_n^i \sim q_n(\cdot|(\widetilde{X}, \widetilde{Z})_{n-1}^i)$
    - Weight $W_n^i \propto \frac{f((X,Z)_n^i|(\widetilde{X},\widetilde{Z})_{n-1}^i)g(y_n|(X,Z)_n^i)}{q_n((X,Z)_n^i|(\widetilde{X},\widetilde{Z})_{n-1}^i)}$

Actually:

- Resample efficiently.
- Only resample when necessary.
- ...

A Rao-Blackwellized SIR Filter

Algorithmically, at iteration n:

- Given $\{W_{n-1}^{X,i}, (X_{1:n-1}^{i}, p(z_{1:n-1}|X_{1:n-1}^{i}, y_{1:n-1})\}$
- **Resample**, obtaining $\{\frac{1}{N}, (\widetilde{X}_{1:n-1}^{i}, p(z_{1:n-1}|\widetilde{X}_{1:n-1}^{i}, y_{1:n-1}))\}$.
- For $i = 1, \ldots, N$:
  - Sample $X_n^i \sim q_n(\cdot|\widetilde{X}_{n-1}^{i})$
  - Set $X_{1:n}^i \leftarrow (\widetilde{X}_{1:n-1}^{i}, X_n^i)$.
  - Weight $W_n^{X,i} \propto \frac{p(X_n^i, y_n|\widetilde{X}_{n-1}^{i})}{q_n(X_n^i|\widetilde{X}_{n-1}^{i})}$
  - Compute $p(z_{1:n}|y_{1:n}, X_{1:n}^i)$.

Requires analytically tractable substructure.

An Approximate Rao-Blackwellized SIR Filter

Algorithmically, at iteration n:

- Given $\{W_{n-1}^{X,i}, (X_{1:n-1}^i, \widehat{p}(z_{1:n-1}|X_{1:n-1}^i, y_{1:n-1})\}$
- **Resample**, obtaining $\{\frac{1}{N}, (\widetilde{X}_{1:n-1}^i, \widehat{p}(z_{1:n-1}|\widetilde{X}_{1:n-1}^i, y_{1:n-1}))\}$.
- For $i = 1, \ldots, N$:
  - Sample $X_n^i \sim q_n(\cdot|\widetilde{X}_{n-1}^i)$
  - Set $X_{1:n}^i \leftarrow (\widetilde{X}_{1:n-1}^i, X_n^i)$.
  - Weight $W_n^{X,i} \propto \frac{\widehat{p}(X_n^i, y_n|\widetilde{X}_{n-1}^i)}{q_n(X_n^i|\widetilde{X}_{n-1}^i)}$
  - Compute $\widehat{p}(z_{1:n}|y_{1:n}, X_{1:n}^i)$.

Is approximate; how does error accumulate?

Exactly Approximated Rao-Blackwellized SIR Filter

At time $n = 1$

- Sample, $X_1^i \sim q^x\left(\cdot \mid y_1\right)$.
- Sample, $Z_1^{i,j} \sim q^z\left(\cdot \mid X_1^i, y_1\right)$.
- Compute and normalise the local weights

$$w_1^z\left(X_1^i, Z_1^{i,j}\right) := \frac{p(X_1^i, y_1, Z_1^{i,j})}{q^z\left(Z_1^{i,j} \middle| X_1^i, y_1\right)}, \; W_1^{z,i,j} := \frac{w_1^z\left(X_1^i, Z_1^{i,j}\right)}{\sum_{k=1}^M w_1^z\left(X_1^i, Z_1^{i,k}\right)},$$

$$\text{define } \widehat{p}(X_1^i, y_1) := \frac{1}{M}\sum_{j=1}^{M} w_1^z\left(X_1^i, Z_1^{i,j}\right).$$

- Compute and normalise the top-level weights

$$w_1^x\left(X_1^i\right) := \frac{\widehat{p}(X_1^i, y_1)}{q^x\left(X_1^i \mid y_1\right)}, \; W_1^{x,i} := \frac{w_1^x\left(X_1^i\right)}{\sum_{k=1}^N w_1^x\left(X_1^k\right)}.$$

At times $n \geq 2$

- Resample

$$\left\{ W_{n-1}^{x,i}, \left(X_{1:n-1}^{i}, \left\{ W_{n-1}^{z,i,j}, Z_{1:n-1}^{i,j} \right\}_j \right) \right\}_i$$

to obtain

$$\left\{ \frac{1}{N}, \left(\widetilde{X}_{1:n-1}^{i}, \left\{ \overline{W}_{n-1}^{z,i,j}, \overline{Z}_{1:n-1}^{i,j} \right\}_j \right) \right\}_i .$$

- Resample $\{\overline{W}_{n-1}^{z,i,j}, \overline{Z}_{1:n-1}^{i,j}\}_j$ to obtain $\{\frac{1}{M}, \widetilde{Z}_{1:n-1}^{i,j}\}_j$.
- Sample $X_n^i \sim q^x(\cdot|\widetilde{X}_{1:n-1}^i, y_{1:n})$; set $X_{1:n}^i := (\widetilde{X}_{1:n-1}^i, X_n^i)$.
- Sample $Z_n^{i,j} \sim q^z\left(\cdot\middle|\, X_{1:n}^i, y_{1:n}, \widetilde{Z}_{1:n-1}^{i,j}\right)$; set $Z_{1:n}^{i,j} := (\widetilde{Z}_{1:n-1}^{i,j}, Z_n^{i,j})$.

- Compute and normalise the local weights

$$w_n^z\left(X_{1:n}^i, Z_{1:n}^{i,j}\right) := \frac{p\left(X_n^i, y_n, Z_n^{i,j} \,\middle|\, \widetilde{X}_{n-1}^i, \widetilde{Z}_{n-1}^{i,j}\right)}{q^z\left(Z_n^{i,j} \,\middle|\, X_{1:n}^i, y_{1:n}, \widetilde{Z}_{1:n-1}^{i,j}\right)},$$

$$\begin{aligned}\widehat{p}(X_n^i, y_n | \widetilde{X}_{1:n-1}^i, y_{1:n-1}) : &= \frac{1}{M}\sum_{j=1}^{M} w_n^z\left(X_{1:n}^i, Z_{1:n}^{i,j}\right), \\ W_n^{z,i,j} : &= \frac{w_n^z\left(X_{1:n}^i, Z_{1:n}^{i,j}\right)}{\sum_{k=1}^{M} w_n^z\left(X_{1:n}^i, Z_{1:n}^{i,k}\right)}.\end{aligned}$$

- Compute and normalise the top-level weights

$$w_n^x\left(X_{1:n}^i\right) := \frac{\widehat{p}\left(X_n^i, y_n \middle| \widetilde{X}_{1:n-1}^i, y_{1:n-1}\right)}{q^x\left(X_n^i \middle| \widetilde{X}_{1:n-1}^i, y_{1:n}\right)},$$

$$W_n^{x,i} := \frac{w_n^x\left(X_{1:n}^i\right)}{\sum_{k=1}^N w_n^x\left(X_{1:n}^k\right)}.$$

How can this be justified?

- As an extended space SIR algorithm.
- Via unbiased estimation arguments.

Note also the $M = 1$ and $M \to \infty$ cases.

How does this differ from CSOL11?

Principally in the *local* weights, benefits including:

- Valid (N-consistent) for all $M \geq 1$ rather than (M, N)-consistent.
- Computational cost $\mathcal{O}(MN)$ rather than $\mathcal{O}(M^2N)$.
- Only requires knowledge of joint behaviour of x or z; doesn't require say $p(x_n|x_{n-1}, z_{n-1})$.

Toy Example: Model

We use a simulated sequence of 100 observations from the model defined by the densities:

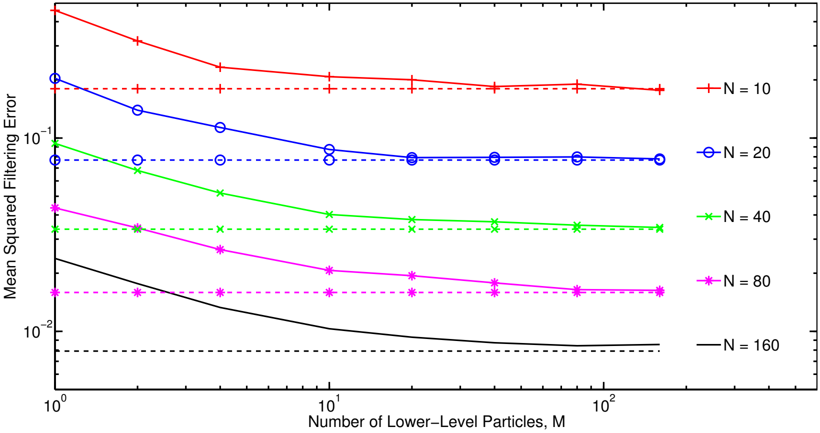
$$\mu(x_1, z_1) = \mathcal{N}\left(\begin{pmatrix} x_1 \\ z_1 \end{pmatrix}; \begin{pmatrix} 0 \\ 0 \end{pmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

$$f(x_n, z_n | x_{n-1}, z_{n-1}) = \mathcal{N}\left(\begin{pmatrix} x_n \\ z_n \end{pmatrix}; \begin{pmatrix} x_{n-1} \\ z_{n-1} \end{pmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

$$g(y_n | x_n, z_n) = \mathcal{N}\left(y_n; \begin{pmatrix} x_n \\ z_n \end{pmatrix}, \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_z^2 \end{bmatrix}\right)$$

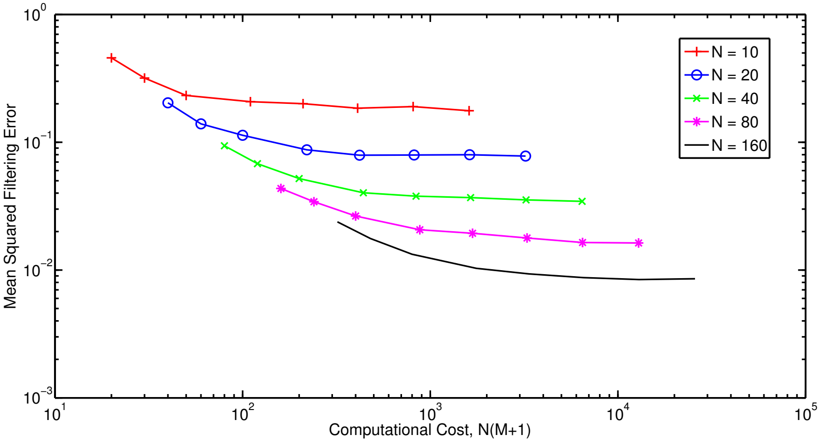
Consider IMSE (relative to optimal filter) of filtering estimate of first coordinate marginals.

Approximation of the RBPF



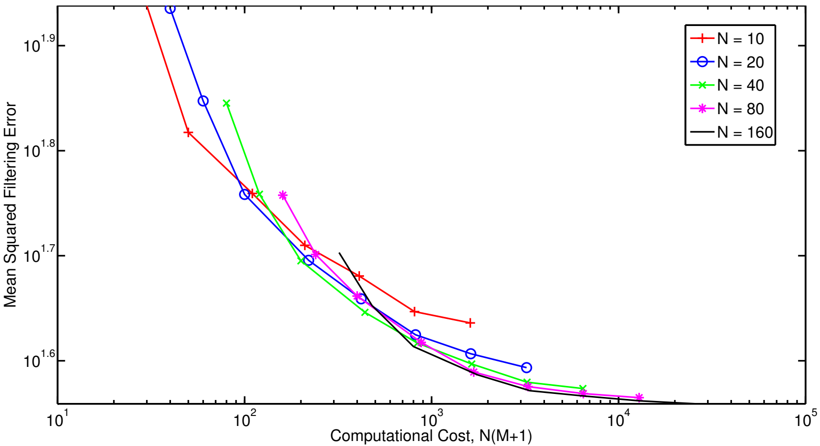
For $\sigma_x^2 = \sigma_z^2 = 1$.

Computational Performance



For $\sigma_x^2 = \sigma_z^2 = 1$.

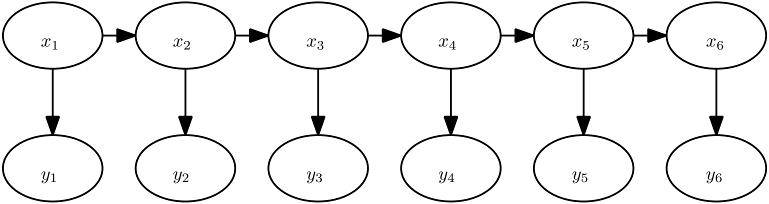
Computational Performance



For $\sigma_x^2 = 10^2$ and $\sigma_z^2 = 0.1^2$.

What About Other HMMs / Algorithms?

Returning to:



- Unobserved Markov chain $\{X_n\}$ transition f.
- Observed process $\{Y_n\}$ conditional density g.
- Density:

$$p(x_{1:n}, y_{1:n}) = f_1(x_1) g(y_1|x_1) \prod_{i=2}^{n} f(x_i|x_{i-1}) g(y_i|z_i).$$

Block Sampling: An Idealised Approach

At time n, given $x_{1:n-1}$; discard $x_{n-L+1:n-1}$:

- Sample from $q(x_{n-L+1:n}|x_{n-L}, y_{n-L+1:n})$.
- Weight with

$$W(x_{1:n}) = \frac{p(x_{1:n}|y_{1:n})}{p(x_{1:n-L}|y_{1:n-1})q(x_{n-L+1:n}|x_{n-L}, y_{1:n-L+1:n})}$$

- Optimally,

$$q(x_{n-L+1:n}|x_{n-L}, y_{n-L+1:n}) = p(x_{n-L+1:n}|x_{n-L}, y_{n-L+1:n})$$
$$W(x_{1:n}) \propto \frac{p(x_{1:n-L}|y_{1:n})}{p(x_{1:n-L}|y_{1:n-1})} = p(y_n|x_{1:n-L}, y_{n-L+1:n-1})$$

- Typically intractable; auxiliary variable approach in [DBS06].

Why *Try* To Block-Sample?

Explicit motivation from the linear Gaussian case:

$$
\begin{aligned}
&\mathrm{Var}_{p(x_{n-L}|y_{1:n-1})}\left[w_{n,L}(X_{n-L})\right] \\
=&\int_{-\infty}^{\infty} \frac{\mathcal{N}^2(x_{n-L};\mu_{n-L|n},\Sigma_{n-L|n})}{\mathcal{N}(x_{n-L};\mu_{n-L|n-1},\Sigma_{n-L|n-1})} dx_{n-L} - 1 \\
=&\frac{\Sigma_{n-L|n-1}}{\sqrt{\Sigma_{n-L|n}(2\Sigma_{n-L|n-1}-\Sigma_{n-L|n})}} \exp\left(\frac{(\mu_{n-L|n}-\mu_{n-L|n-1})^2}{2\Sigma_{n-L|n-1}-\Sigma_{n-L|n}}\right) - 1.
\end{aligned}
$$

Optimal Block Sampling Central Limit Theorem

- Let $\varphi_n : \mathcal{X}^n \to \mathbb{R}$, $\bar{\varphi}_n = \int \varphi_n(x_{1:n}) p(x_{1:n}|y_{1:n}) dx_{1:n}$.
- Allow $\widehat{\varphi}^N_{n,L}$ to denote the estimate obtained with lag-L.
- Then:

$$\widehat{\varphi}^N_{n,L} = \frac{\sum_{i=1}^{N} \varphi(X^i_{n,\star}) \prod_{p=L+1}^{n} \left[w_{p,L}(X^i_{p-1,p-L}) \right]}{\sum_{i=L+1}^{N} \prod_{p=L+1}^{n} w_{p,L}(X^i_{p-1,p-L})}$$

where $X^i_{n,\star} := (X^i_{L,1}, X^i_{L+1,2}, \ldots, X^i_{n-1,n-L}, X^i_{n,n-L+1:n})$.

- Under basic regularity conditions:

$$\lim_{N\to\infty} \sqrt{N}(\widehat{\varphi}^N_{n,L} - \bar{\varphi}_n) \xrightarrow{d} \mathcal{N}(0, V_n(\varphi_n))$$

$$\text{where } V(\widehat{\varphi}) = \int \prod_{p=L+1}^{n} \frac{p(x_{p-L}|y_{1:p})^2}{p(x_{p-L}|y_{1:p-1})} (\varphi(x_{1:n}) - \bar{\varphi})^2 dx_{1:n}.$$

Toy Model: Linear Gaussian HMM

- Linear, Gaussian state transition:

$$f(x_t|x_{t-1}) = \mathcal{N}(x_t; x_{t-1}, 1)$$

- and likelihood

$$g(y_t|x_t) = \mathcal{N}(y_t; x_t, 1)$$

- Analytically: Kalman filter/smoother/etc.
- Simple bootstrap PF:
  - Proposal:

$$q(x_t|x_{t-1}, y_t) = f(x_t|x_{t-1})$$

  - Weighting:

$$W(x_{t-1}, x_t) \propto g(y_t|x_t)$$

  - Resample residually every iteration.

More than one SMC Algorithm?

- Standard approach:
  - Run an SIR algorithm with N particles.
  - Use

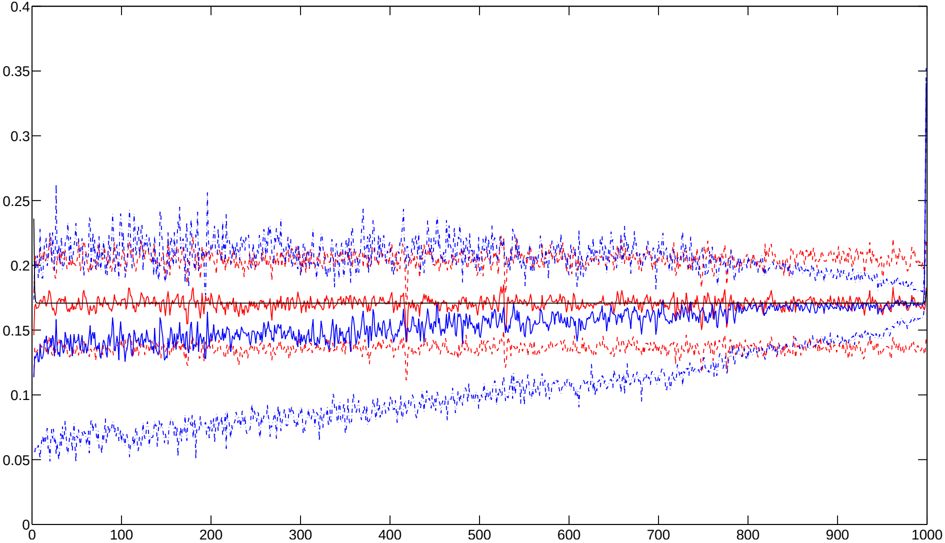
$$\pi_n^N(dx_{1:n}) = \sum_{i=1}^{N} W_n^i \delta_{X_{1:n}^i}(dx_{1:n}).$$

- A crude alternative:
  - Run $L = \lfloor N/M \rfloor$ algorithms with M particles.
  - Use

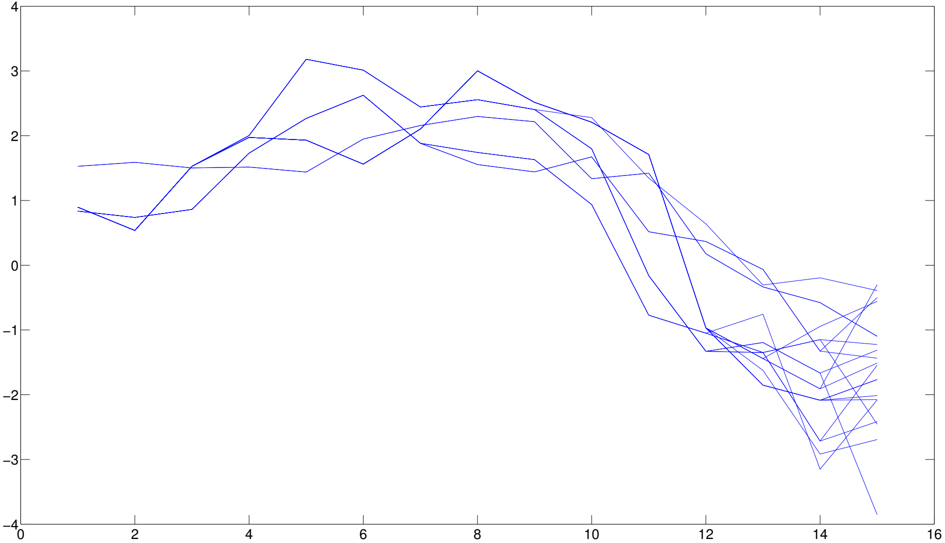
$$\pi_n^{M,l}(dx_{1:n}) = \sum_{i=1}^{M} W_n^{l,i} \delta_{X_{1:n}^{l,i}}(dx_{1:n}).$$

  - Guarantees L i.i.d. samples.
  - For small M their distribution may be poor.

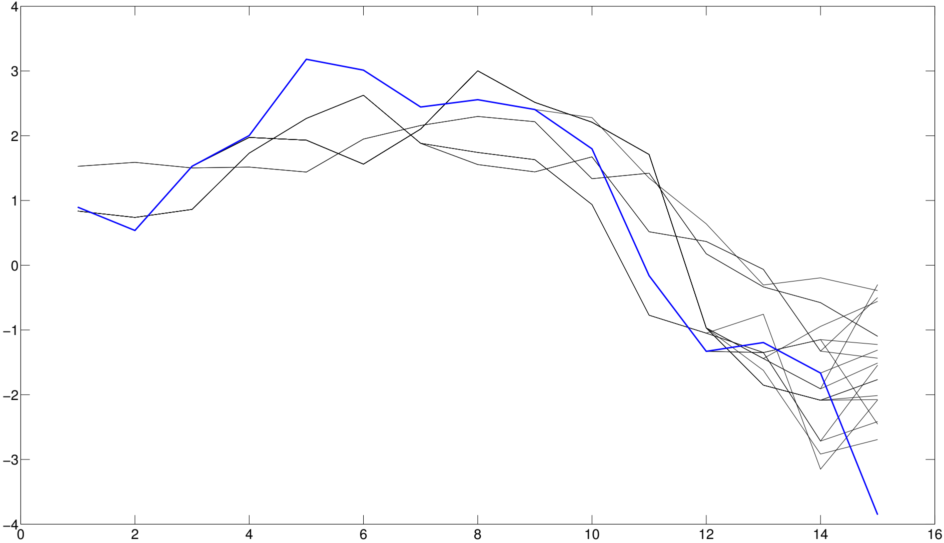
Covariance Estimation: 1d Linear Gaussian Model



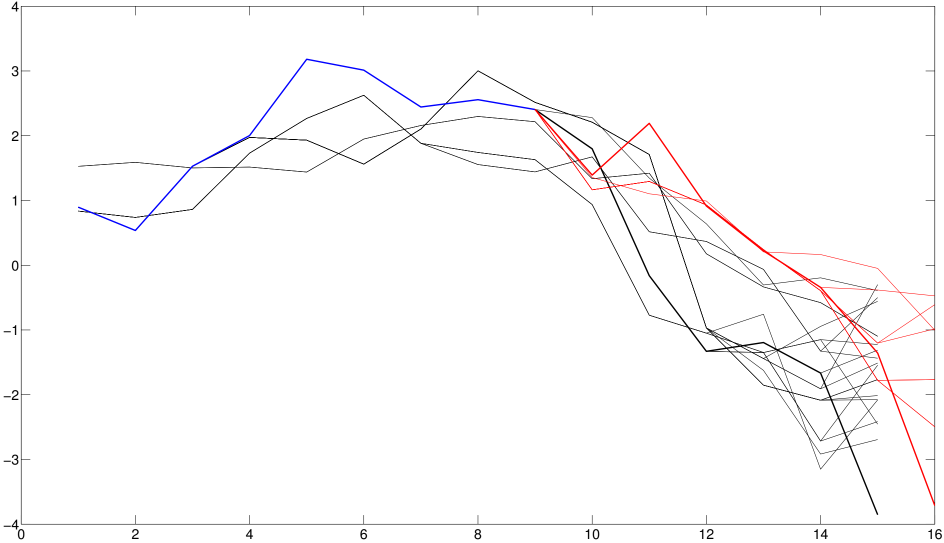
Local Particle Filtering: Current Trajectories



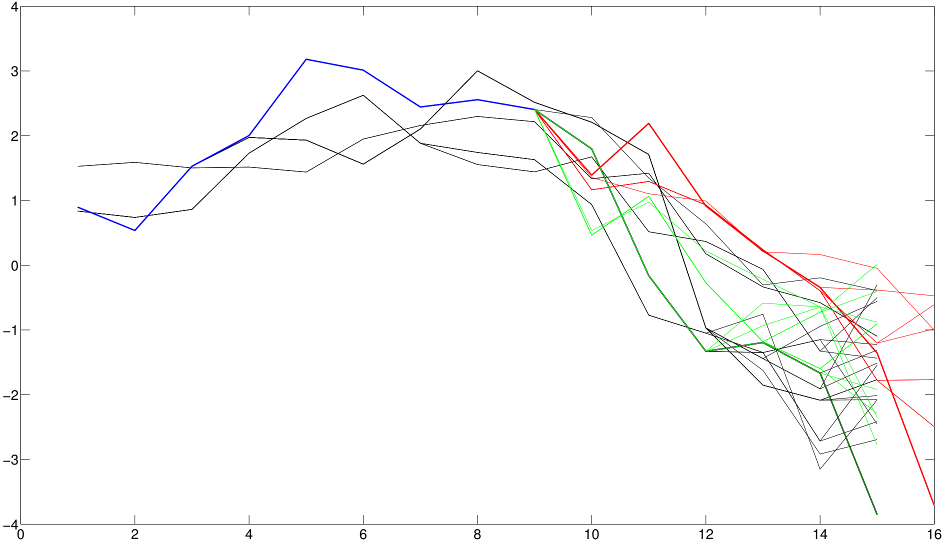
Local Particle Filtering: First Particle



Local Particle Filtering: SMC Proposal



Local Particle Filtering: CSMC Auxiliary Proposal



Local SMC

- Propose from:

$$\mathcal{U}_{1:M}^{\otimes n-1}(b_{1:n-2}, k)p(x_{1:n-1}|y_{1:n-1})\psi_{n,L}^{M}(\overline{\mathbf{a}}_{n-L+2:n}, \overline{\mathbf{x}}_{n-L+1:n}, \overline{k}; x_{n-L})$$
$$\widetilde{\psi}_{n-1,L-1}^{M}\left(\widetilde{\mathbf{a}}_{n-L+2:n-1}^{\ominus k}, \widetilde{\mathbf{x}}_{n-L+1:n-1}^{\ominus k}; x_{n-L} \,||\, b_{n-L+2:n-1}, x_{n-L+1:n-1}\right)$$

- Target:

$$\mathcal{U}_{1:M}^{\otimes n}(b_{1:n-L}, \bar{b}_{n,n-L+1:n-1}^{\bar{k}}, \bar{k})p(x_{1:n-L}, \overline{x}_{n-L+1:n}^{\bar{b}_{n,n-L+1:n}^{\bar{k}}}|y_{1:n})$$
$$\widetilde{\psi}_{n,L}^{M}\left(\overline{\mathbf{a}}_{n-L+2:n}^{\ominus\bar{k}}, \overline{\mathbf{x}}_{n-L+1:n}^{\ominus\bar{k}}; x_{n-L} \,\middle|\middle|\, \bar{b}_{n,n-L+1:n}^{\bar{k}}, \overline{x}_{n-L+1:n}^{\bar{b}_{n,n-L+1:n}^{\bar{k}}}\right)$$
$$\psi_{n-1,L-1}^{M}\left(\widetilde{\mathbf{a}}_{n-L+2:n-1}, \widetilde{\mathbf{x}}_{n-L+1:n-1}, k; x_{n-L}\right).$$

- Weight: $\bar{Z}_{n-L+1:n}/\widetilde{Z}_{n-L+1:n-1}$.

Key Identity

$$\frac{\psi_{n,L}^{M}(\mathbf{a}_{n-L+2:n},\mathbf{x}_{n-L+1:n},k;x_{n-L})}{p(x_{n-L+1:n}|x_{n-L},y_{n-L+1:n})\widetilde{\psi}_{n,L}^{M}(\mathbf{a}_{n-L+2:n}^{\ominus k},\mathbf{x}_{n-L+1:n}^{\ominus k},k;x_{n-L}\,||\ldots\,)}$$

$$=\frac{q\left(x_{n-L+1}^{b_{n,n-L+1}^{k}}|x_{n-L}\right)\left[\prod_{p=n-L+2}^{n} r\left(b_{n,p}^{k}|\mathbf{w_{p-1}}\right)q\left(x_{p}^{b_{n,p}^{k}}|x_{p-1}^{b_{n,p-1}^{n}}\right)\right]r(k|\mathbf{w}_{n})}{p(x_{n-L+1:n}|x_{n-L},y_{n-L+1:n})}$$

$$=\widehat{Z}_{n-L+1:n}/p(y_{n-L+1:n}|x_{n-L})$$

Bootstrap Local SMC

- Top Level:
  - Local SMC proposal.
  - Stratified resampling when ESS$< N/2$.
- Local SMC Proposal:
  - Proposal:

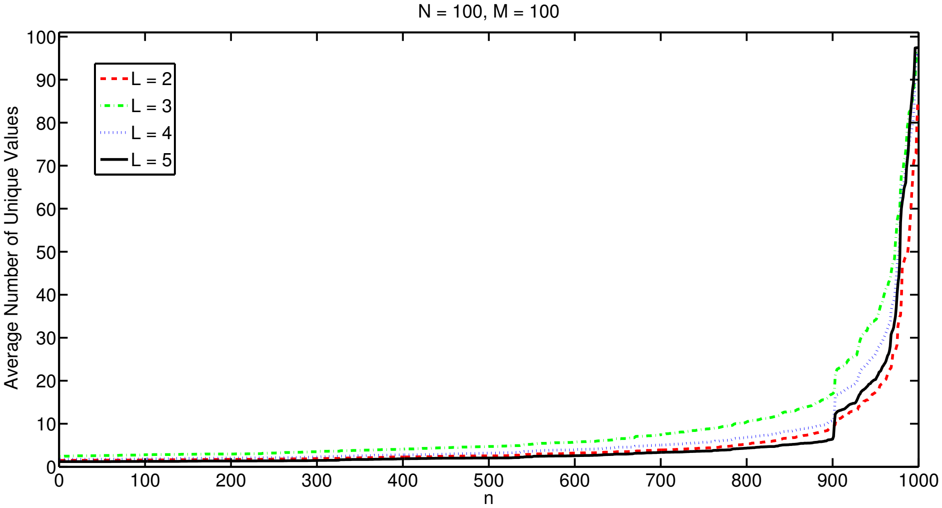
$$q(x_t|x_{t-1}, y_t) = f(x_t|x_{t-1})$$

  - Weighting:

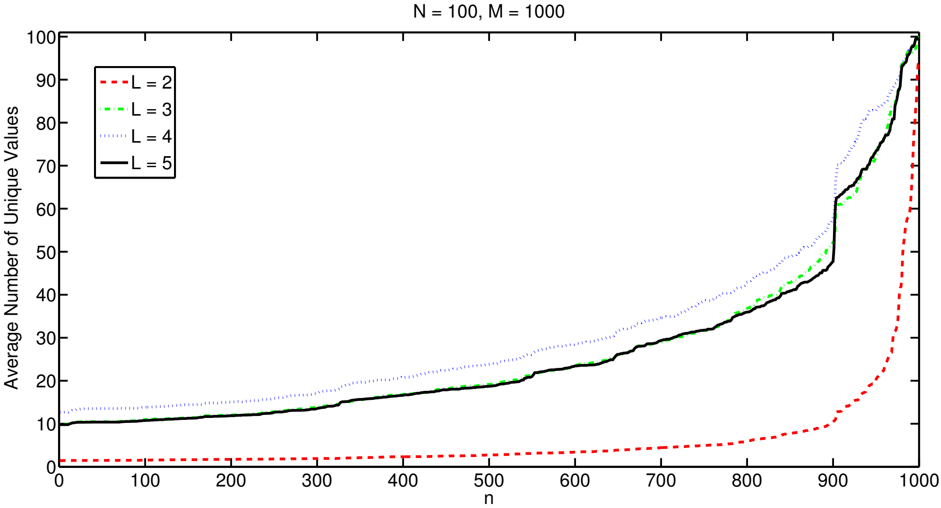
$$W(x_{t-1}, x_t) \propto \frac{f(x_t|x_{t-1})g(y_t|x_t)}{f(x_t|x_{t-1})} = g(y_t|x_t)$$

  - Resample multinomially every iteration.

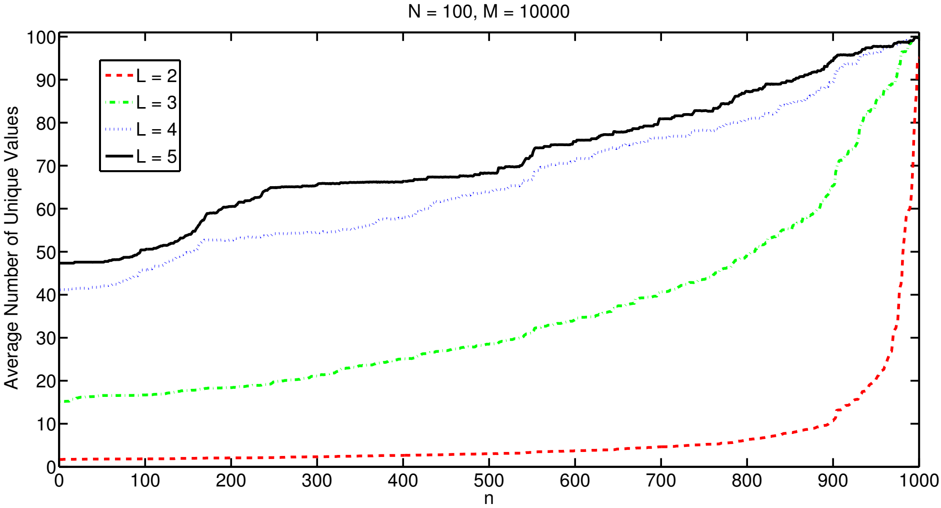
Bootstrap Local SMC: M=100



Bootstrap Local SMC: M=1000



Bootstrap Local SMC: M=10000



Stochastic Volatility Bootstrap Local SMC

- Model:

$$f(x_i|x_{i-1}) = \mathcal{N}\left(\phi x_{i-1}, \sigma^2\right)$$
$$g(y_i|x_i) = \mathcal{N}\left(0, \beta^2 \exp(x_i)\right)$$

- Top Level:
  - Local SMC proposal.
  - Stratified resampling when ESS$< N/2$.
- Local SMC Proposal:
  - Proposal:

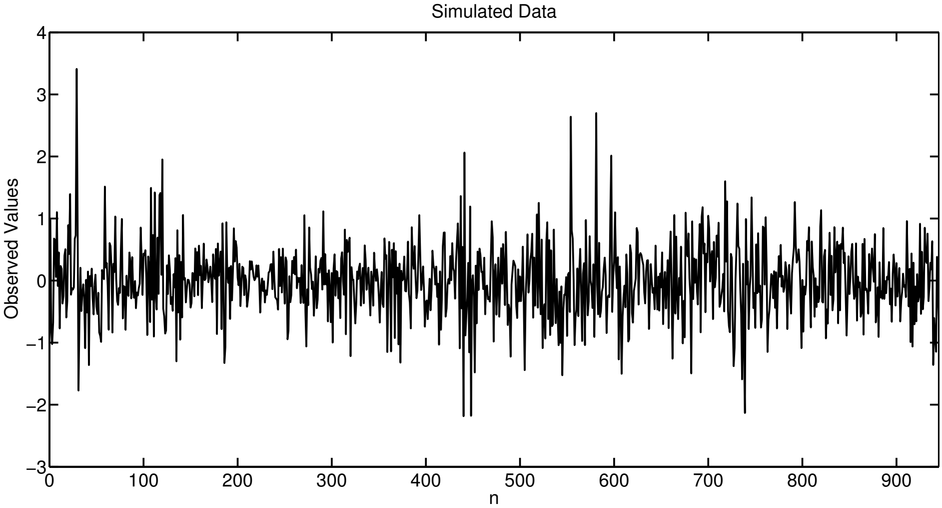
$$q(x_t|x_{t-1}, y_t) = f(x_t|x_{t-1})$$

  - Weighting:

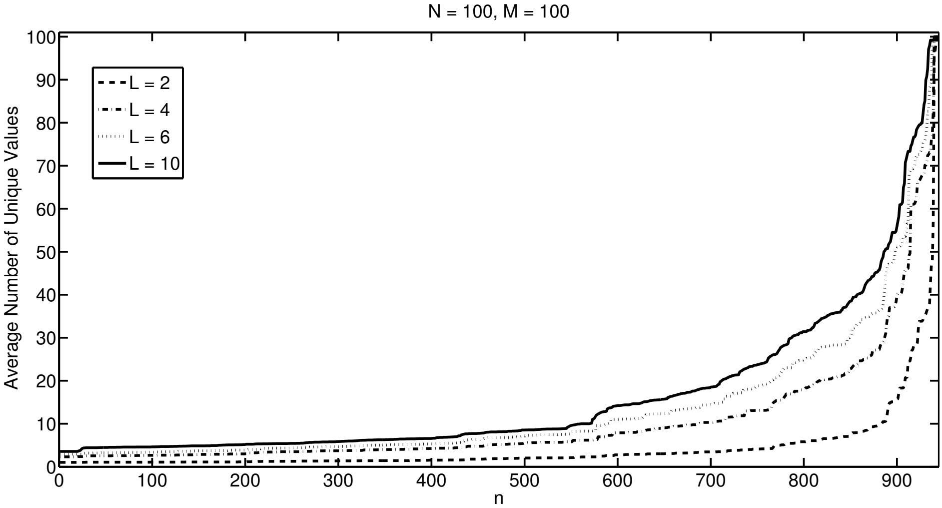
$$W(x_{t-1}, x_t) \propto \frac{f(x_t|x_{t-1})g(y_t|x_t)}{f(x_t|x_{t-1})} = g(y_t|x_t)$$

  - Resample residually every iteration.

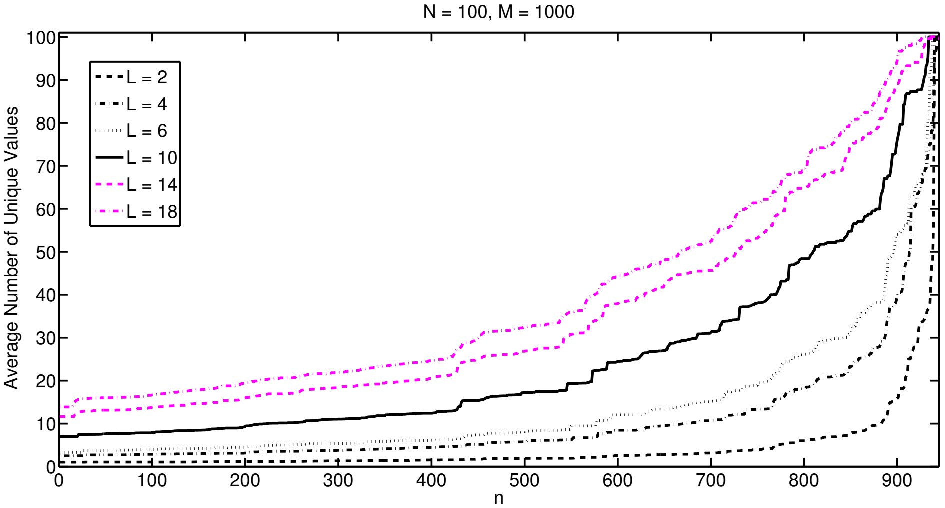
SV Simulated Data



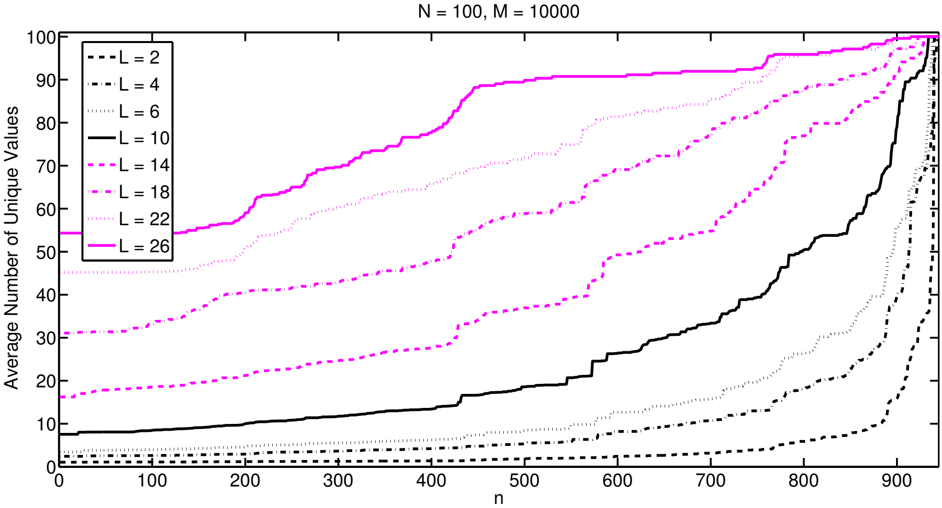
SV Bootstrap Local SMC: M=100



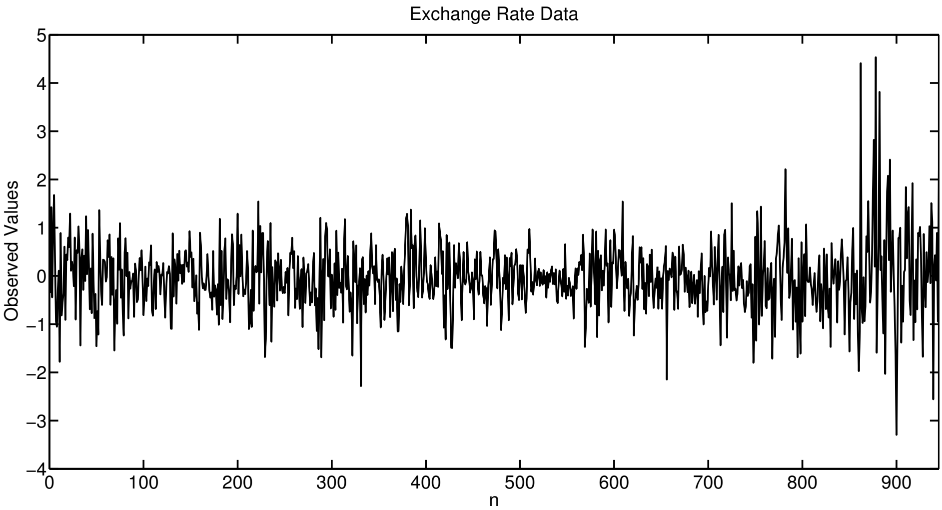
SV Bootstrap Local SMC: M=1000



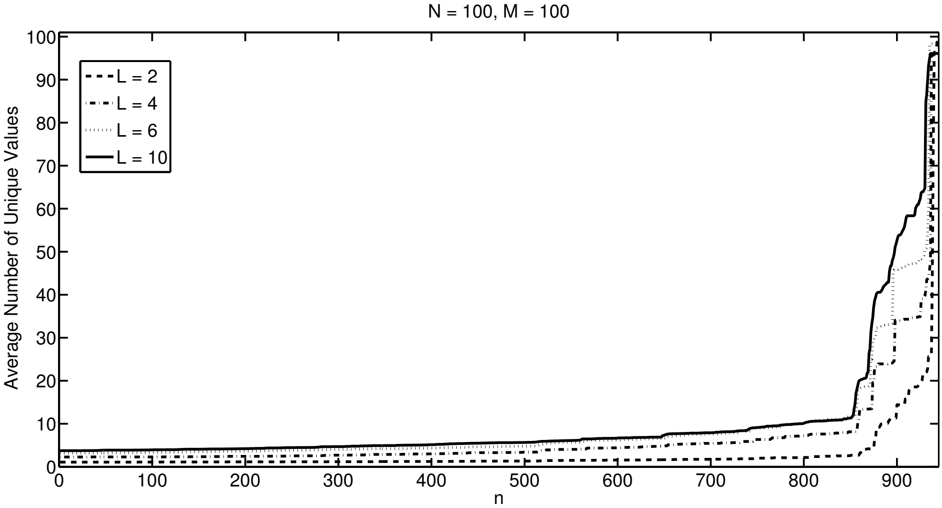
SV Bootstrap Local SMC: M=10000



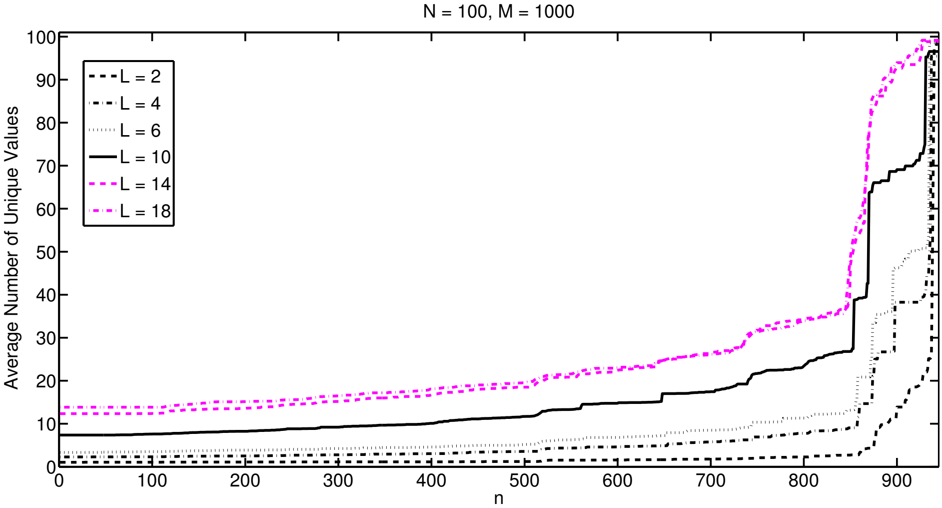
SV Exchange Rata Data



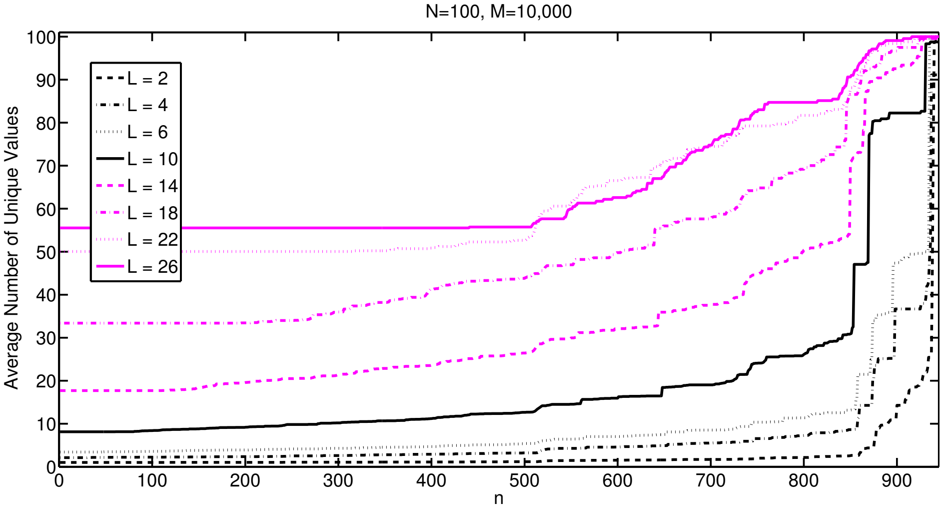
SV Bootstrap Local SMC: M=100



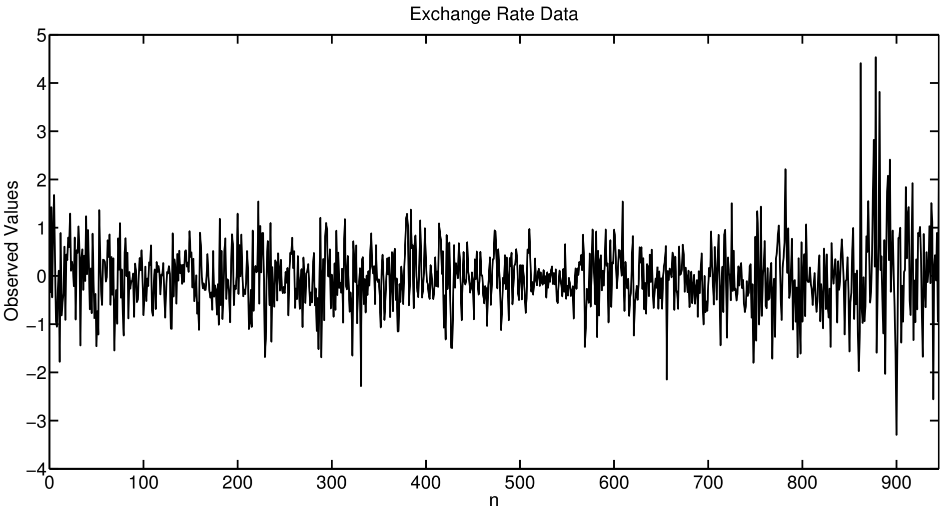
SV Bootstrap Local SMC: M=1000



SV Bootstrap Local SMC: M=10000



SV Exchange Rata Data



In Conclusion

- SMC can be used hierarchically.
- Software implementation is not difficult [Joh09].
- The Rao-Blackwellized particle filter can be approximated *exactly*
  - Can reduce estimator variance at fixed cost.
  - Attractive for distributed/parallel implementation.
  - Allows combination of different sorts of particle filter.
  - Can be combined with other techniques for parameter estimation etc..
- The optimal block-sampling particle filter can be approximated *exactly*
  - Requiring only simulation from the transition and evaluation of the likelihood
  - Easy to parallelise
  - Low storage cost

References I

- C. Andrieu and A. Doucet. Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society B*, 64(4):827–836, 2002.
- C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo. *Journal of the Royal Statistical Society B*, 72(3):269–342, 2010.
- R. Chen and J. S. Liu. Mixture Kalman filters. *Journal of the Royal Statistical Society B*, 62(3):493–508, 2000.
- T. Chen, T. Schön, H. Ohlsson, and L. Ljung. Decentralized particle filter with arbitrary state decomposition. *IEEE Transactions on Signal Processing*, 59(2):465–478, February 2011.
- A. Doucet, M. Briers, and S. Sénécal. Efficient block sampling strategies for sequential Monte Carlo methods. *Journal of Computational and Graphical Statistics*, 15(3):693–711, 2006.
- N. J. Gordon, S. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, April 1993.
- A. M. Johansen and A. Doucet. Hierarchical particle sampling for intractable state-space models. CRiSM working paper, University of Warwick, 2012. In preparation.
- A. M. Johansen. SMCTC: Sequential Monte Carlo in C++. *Journal of Statistical Software*, 30(6):1–41, April 2009.
- A. M. Johansen, N. Whiteley, and A. Doucet. Exact approximation of Rao-Blackwellised particle filters. In *Proceedings of 16th IFAC Symposium on Systems Identification*. IFAC, 2012.