Sequential Posterior Simulators for Bayesian Inference

John Geweke
University of Technology Sydney; Erasmus University;
University of Colorado; Amazon.com
Garland Durham
University of Colorado

Vienna University of Economics and Business
Institute for Statistics and Mathematics

May 17, 2013

Goals and tools

- Goal: Better quantitative decision-making
  - Using formal probability methods
  - When data are an important part of the information (but never all of the information)
  - When data are time series (including longitudinal data)
- Tools
  - Thinking carefully about formal and coherent probability models
  - In my research this leads to
    - Updating and prediction
    - Optimal smoothing and filtering
    - Law of inverse probability
    - Bayesian inference

Notation

- Random vectors and data
  - Y_t $(t = 1, \ldots, T)$: Observable random vectors
  - $Y_{t_1:t_2}$: The collection $\{Y_{t_1}, \ldots, Y_{t_2}\}$
  - y_t: Observed value of Y_t (one data vector)
  - $y_{t_1:t_2}$: The collection $\{y_{t_1}, \ldots, y_{t_2}\}$ (so $y_{1:T}$ is the data set)
- Formal and coherent probability models
  - Models $\mathcal{M}_i$, each with unobservables (often parameters) θ_i $(i = 1, \ldots, M)$
  - Prior density in model $\mathcal{M}_i$:

$$p(\theta_i \mid \mathcal{M}_i)$$

  - Sequence of conditional densities in model M_i:

$$p(Y_t \mid Y_{1:t-1}, \theta_i, \mathcal{M}_i) \quad (t = 1, 2, \ldots),$$

  - Henceforward, notation dispenses with the conditioning on $\mathcal{M}_i$, but the idea is always there.

Implications of the formal probability model

- For any $t = 1, 2, \ldots$. Updated conditional parameter density

$$p\left(\theta \mid y_{1:t-1}\right)$$

represented as

$$\theta_{jn} \sim p\left(\theta \mid y_{1:t-1}\right) \quad (t = 1, 2, \ldots) \qquad (j = 1., \ldots, J; n = 1, \ldots, N);$$

- Predictive density

$$p\left(Y_t \mid y_{1:t-1}\right) = \int_{\Theta} p\left(Y_t \mid y_{1:t-1}, \theta\right) p\left(\theta \mid y_{1:t-1}\right)$$

represented as

$$Y_{t,jn} \sim p\left(Y_t \mid \theta_{jn}\right) \qquad (j = 1., \ldots, J; n = 1, \ldots, N);$$

Implications (continued)

- Predictive likelihood

$$p\left(y_t \mid y_{1:t-1}\right) = \int_{\Theta} p\left(y_t \mid y_{1:t-1}, \theta\right) p\left(\theta \mid y_{1:t-1}\right);$$

represented as

$$(JN)^{-1} \sum_{j=1}^{J} \sum_{n=1}^{N} p\left(y_t \mid y_{1:t-1}, \theta_{jn}\right).$$

- If the sample size is T, then the log marginal likelihood is

$$\log p\left(y_{1:T}\right) = \sum_{t=1}^{T} \log p\left(y_t \mid y_{1:t-1}\right)$$

for Model $\mathcal{M}_i$

Motivation

- Sequential posterior simulators combine ideas from sequential Monte Carlo (particle filtering), importance sampling, resampling and Markov chain Monte Carlo.
- They (now) have a solid theoretical justification.
- They apply immediately to a huge range of models: all that is needed is code to evaluate the prior density and the likelihood function and to simulate from the prior distribution.
- While they are globally sequentially, they are locally parallel and nearly idea for massively parallel processing.
- Recent innovation in hardware (graphics processing units) and software (C/CUDA, Matlab) provide access to this technology that is both quick and inexpensive.

Sequential posterior simulation algorithms: Conditions for some theorems and algorithms

- *Prior distribution:* The model specifies a proper prior distribution that can be evaluated in closed form.
- *Likelihood function evaluation:* The sequence of conditional densities

$$p\left(y_t \mid y_{1:t-1}, \theta\right) \quad (t = 1, \ldots, T)$$

  can be evaluated in closed form. (T is sample size.)
- *Bounded likelihood:* The sequence of densities $p\left(y_{1:T} \mid \theta\right)$ is bounded above by $\overline{p} < \infty$ for all $\theta \in \Theta$.
- *Existence of prior moments:* If the algorithm is used to approximate $\mathrm{E}\left[g\left(\theta\right) \mid y_{1:T}\right]$, then $\mathrm{E}\left[g\left(\theta\right)^{2+\delta}\right] < \infty$ for some $\delta > 0$.

An algorithm with all design parameters fixed

- Always fixed:
  - Number and organization of particles: J, N
- Cycles in the algorithm
  - $\ell = 1, \ldots, L$ cycles
  - Cycle ℓ processes observations $y_{t_{\ell-1}+1:t_\ell}$
  - $0 = t_0 < t_1 < \cdots < t_{L-1} < t_L = T$
- Fixed algorithm parameters in this but not in the adaptive version of the algorithm
  - Number L and timing of cycles: $t_1, \ldots, t_{L-1}$
  - Number of Metropolis steps in each cycle: $R_1, \ldots, R_L$
  - Metropolis random walk variance matrices in each step, each cycle: $\Sigma_{\ell r}$ $(r = 1, \ldots R_\ell; \ell = 1, \ldots, L)$

A little more detail

- Particles in cycle ℓ of the algorithm $(\ell = 1, \ldots, L)$
  - C phase: $\theta_{jn}^{(\ell-1)}$
    - At the start, $\theta_{jn}^{(\ell-1)} \sim p\left(\theta \mid y_{1:t_{\ell-1}}\right)$
    - Use importance sampling to update to $\theta_{jn}^{(\ell-1)} \sim p\left(\theta \mid y_{1:t_\ell}\right)$
  - Transition from C to S phase: $\theta_{jn}^{(\ell-1)}$ with associated weights $w\left(\theta_{jn}^{(\ell-1)}\right)$
  - S phase: Select particles $\theta_{jn}^{(\ell,0)}$ (no weights)
    - For each $\theta_{jn}^{(\ell,0)}$ there exists some n' such that $\theta_{jn}^{(\ell,0)} = \theta_{jn'}^{(\ell-1)}$
  - M phase
    - At the end of Metropolis step r denote the particles $\theta_{jn}^{(\ell,r)}$.
    - End of M phase and cycle (Metropolis step R_ℓ): $\theta_{jn}^{(\ell)} = \theta_{jn}^{(\ell,R_\ell)}$
- At the conclusion of the algorithm: $\theta_{jn} = \theta_{jn}^{(L)}$

Convergence and posterior moments

- Because there is no adaptation, existing convergence results ($N \to \infty$; Chopin 2004 Theorem 2) apply.
- Posterior moment: $\overline{g} = \mathrm{E}\left[g\left(\theta\right) \mid y_{1:T}\right]$
- Approximation: $\overline{g}^{(J,N)} = \left(NJ\right)^{-1} \sum_{j=1}^{J} \sum_{n=1}^{N} g\left(\theta_{jn}\right)$
- Chopin 2004 Theorem 2: $\left(JN\right)^{1/2} \left(\overline{g}^{(J,N)} - \overline{g}\right) \xrightarrow{d} N\left(0, v\right)$
  - Our v not the same as v in Chopin 2004 Theorem 2
  - But that paper does not show how to approximate v (nor does any other, to our knowledge) and this is required for practical application in any event.

Assessing the accuracy of posterior moments

- Define within-group sample means $\overline{g}_j^N = N^{-1} \sum_{n=1}^N g(\theta_{jn})$.
- Chopin 2004 Theorem 2 applies to each one $(N \to \infty)$, so limiting distribution is normal.
- Independence of particles θ_{jn} across groups in every cycle of every phase (recall: residual sampling done group-by-group), hence $\overline{g}_1^N, \ldots, \overline{g}_J^N$ mutually independent.
- Variance estimate appropriate to $\overline{g}^{(J,N)}$ is

$$\widehat{v}^N = \sum_{j=1}^J \left(\overline{g}_j^N - \overline{g}^N \right)^2 / J(J-1)$$

- and

$$(J-1)\,\widehat{v}^N / v \xrightarrow{d} \chi^2(J-1).$$

Problem with the fixed design algorithm

- Sound theoretical foundations, but it does not work as a practical matter.
- These must be specified before the algorithm executes:
  - Number L and timing of cycles: $t_1, \ldots, t_{L-1}$
  - Number of Metropolis steps in each cycle: $R_1, \ldots, R_L$
  - Metropolis random walk variance matrices in each step, each cycle: $\Sigma_{\ell r}$ $(r = 1, \ldots R_\ell; \ell = 1, \ldots, L)$
- This never works

Adaptive design algorithm: Main features

- User specifies: J, N, R
- In the C phase, determine the effective sample size

$$ESS = \left[\sum_{j=1}^{J}\sum_{n=1}^{N} w_s\left(\theta_{jn}^{(\ell-1)}\right)\right]^2 \Big/ \sum_{j=1}^{J}\sum_{n=1}^{N} w_s\left(\theta_{jn}^{(\ell-1)}\right)^2$$

  after each observation is added. Proceed to the CS transition if $ESS/NJ < D_1 = 0.5$.
- In the M phase, Metropolis step r:
  - $\Sigma_{\ell r} = h_{\ell r} \cdot \text{var}\left(\theta_{jn}^{(\ell,r-1)}\right)$;
  - $h_{\ell 1} = 0.5$;
  - After each Metropolis step r is completed compute the acceptance rate across all particles in all groups. If this exceeds 0.25, $h_{\ell,r+1} = h_{\ell r} + 0.01$; else $h_{\ell,r+1} = h_{\ell r} - 0.01$;
  - R Metropolis steps if $ESS/NJ > D_2$, otherwise $3R$ steps.

Problem with the adaptive algorithm

- It "works" in a very wide variety of models.
  - But it does not have sound theoretical foundations
  - nor is it likely to in the foreseeable future
- Why we care
  - The adaptive algorithm works well on conventional CPU's
  - and it works very well in cheap desktop parallel computing with graphical processing units (GPUs).
  - and (I conjecture) algorithms like this will be in widespread application in 5 to 10 years

Providing a sound theoretical foundation for the adaptive algorithm

- Run the adaptive version of the algorithm.
- Save the design parameters
  - Number L and timing of cycles: $t_1, \ldots, t_{L-1}$
  - Number of Metropolis steps in each cycle: $R_1, \ldots, R_L$
  - Metropolis random walk variance matrices in each step, each cycle: $\Sigma_{\ell r}$ $(r = 1, \ldots R_\ell; \ell = 1, \ldots, L)$
- Discard the particles
- Run the fixed design algorithm using the saved design parameters and new random number generator seeds.

Exponential generalized autoregressive conditional heteroskedasticity (EGARCH) model

- Background:
  - Decision-making about allocation and pricing of financial assets, and economic policy
  - Mean returns are almost unpredictable, and we know why.
  - The spread of the distribution (loosely, volatility) is constantly changing.
  - This matters – greatly! – for pricing and allocation of asset derivatives and economic policy
- EGARCH model
  - Variant here is a substantial generalization of a successful model
  - Strong competitor with other models (Geweke and Durham pooling paper, 2011)
  - Model is well suited to show robustness properties of the sequential posterior simulator

Model

- Sequence of observed asset returns $\{y_t\}$
- Evolution of volatility factors

$$v_{kt} = \alpha_k v_{k,t-1} + \beta_k \left(|\varepsilon_{t-1}| - (2/\pi)^{1/2} \right) + \gamma_k \varepsilon_{t-1} \quad (k = 1, \ldots, K)$$

- Then

$$y_t = \mu_Y + \sigma_Y \exp\left(\sum_{k=1}^{K} v_{kt}/2 \right) \varepsilon_t$$

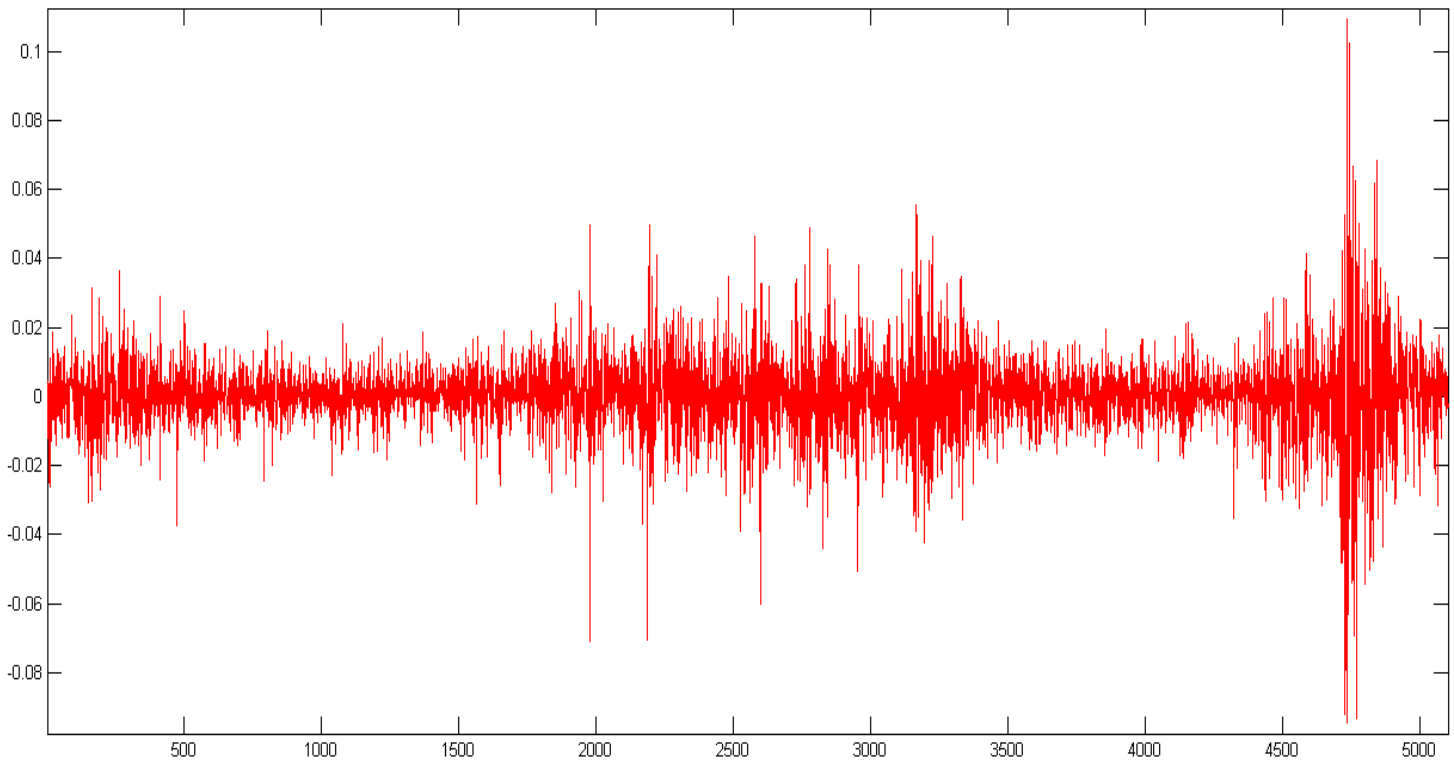
- with

$$p(\varepsilon_t) = \sum_{i=1}^{I} p_i \phi(x_i; \mu_i, \sigma_i^2); \;\; \mathrm{E}(\varepsilon_t) = 0, \;\; \mathrm{var}(\varepsilon_t) = 1.$$

- Notation: This is the EGARCH(K,I) model.

Data

S&P 500 index closing value p_t on trading days from January 2, 1990 $(t = 0)$ through March 31, 2010 $(t = T = 5100)$.
The returns are $y_t = \log(p_t / p_{t-1})$ $(t = 1, \ldots, T)$.



Performance

Model	Time (secs.)	Cycles L	log ML	NSE
EGARCH$(1,1)$	716	52	16,641.76	0.01
EGARCH$(1,2)$	1224	68	16,712.55	0.02
EGARCH$(2,1)$	849	59	16,669.50	0.03
EGARCH$(2,2)$	1377	73	16,735.80	0.04
EGARCH$(2,3)$	1796	76	16,750.73	0.03
EGARCH$(3,2)$	1455	73	16,733.94	0.04
EGARCH$(3,3)$	1857	74	16,748.62	0.04
EGARCH$(3,4)$	2492	80	16,748.42	0.04
EGARCH$(4,3)$	2074	75	16,745.68	0.05
EGARCH$(4,4)$	2613	78	16,745.34	0.04

$J = 64$; $N = 4096$; $J \cdot N = 262,144$; $R = 55$

Moment of interest: $g(\theta) = 100 \cdot P(Y_{t+1} < -0.03 \mid y_{1:t}, \theta)$

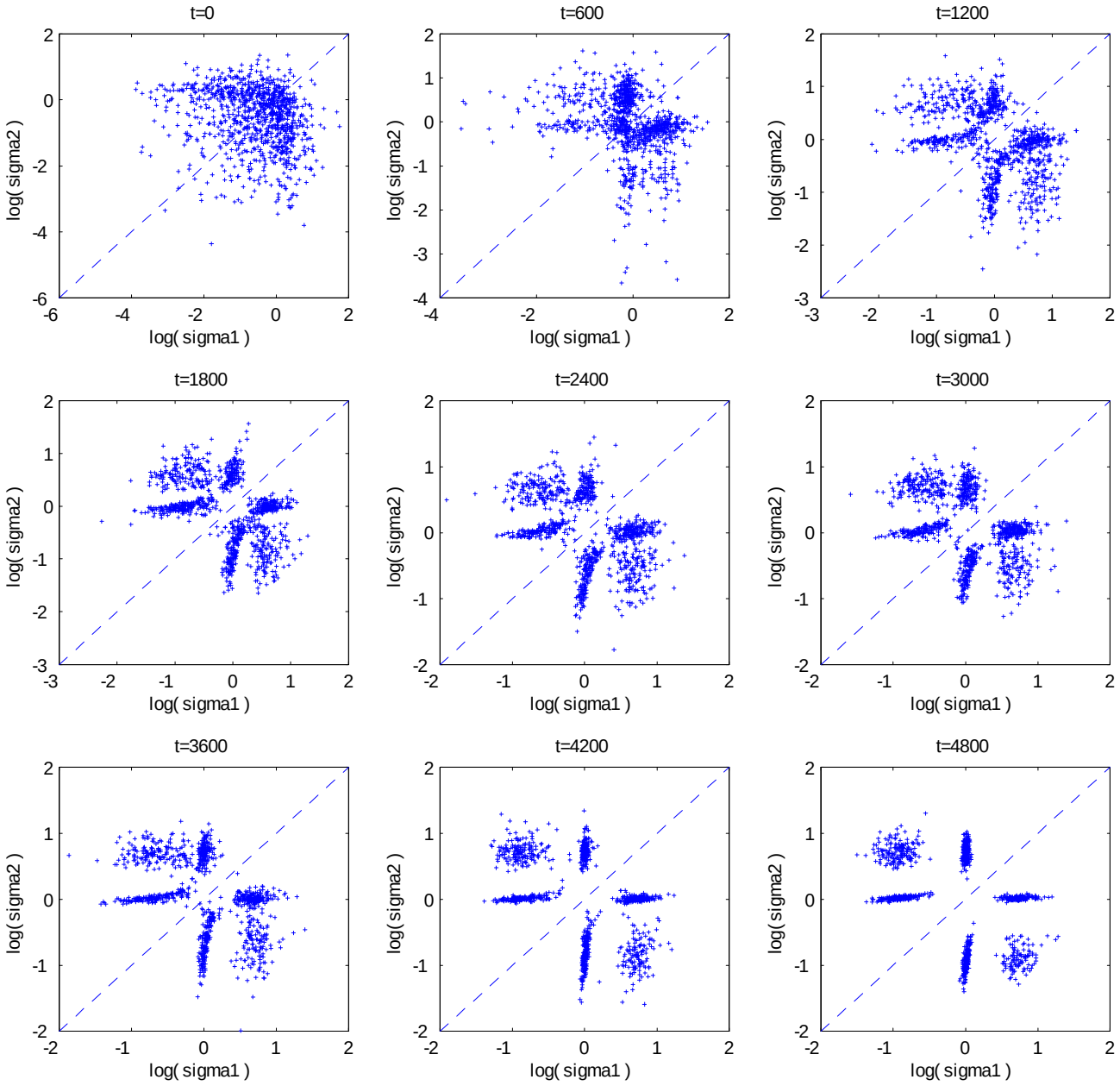
	t = March 31, 2009			t = March 31, 2010		
	Posterior mean	*NSE*	*RNE*	Posterior mean	*NSE*	*RNE*
$R = 5$	9.7045	0.0481	0.0009	0.0632	0.0015	0.0013
$R = 8$	9.7996	0.0379	0.0014	0.0704	0.0012	0.0024
$R = 13$	9.7445	0.0267	0.0027	0.0728	0.0009	0.0045
$R = 21$	9.7418	0.0198	0.0050	0.0748	0.0005	0.0137
$R = 34$	9.7639	0.0114	0.0145	0.0759	0.0004	0.0268
$R = 55$	9.7834	0.0074	0.0343	0.0770	0.0002	0.0643
$R = 89$	9.7851	0.0047	0.0857	0.0770	0.0001	0.1665
$R = 144$	9.7760	0.0037	0.1382	0.0769	0.0001	0.3524
EGARCH(2,3) $J = 64$; $N = 4096$; $J \cdot N = 262,144$						

Multimodal posterior distributions

- There are $J!$ permutations of the factors and $K!$ permutations of the normal components of ε_t.
  - $6 \times 2 = 12$ in the preferred EGARCH$(2, 3)$ model
  - 24×24 in the EGARCH$(4, 4)$ model for which the algorithm performed very smoothly.
- The prior distribution is also symmetric with respect to the factors and normal components of ε_t.
- This leads to reflections or "mirror images" in the posterior distribution.
- These permutations present a severe challenge for Markov chain Monte Carlo (MCMC) and are a standard test for such algorithms.
  - (Generic MCMC simply cannot handle the multimodality here.)

Performance of the algorithm

- Focus on the 3! permutations of parameters (p_s, μ_s, σ_s) of the normal mixture distribution
- Consider a parameter vector θ with 3 distinct values of the triplets (p_s, μ_s, σ_s)
  - There are six distinct ways these could be assigned to the three components of the normal mixture.
  - These permutations define six points θ_u $(u = 1, \ldots, 6)$.
- Thus the posterior distribution has six "mirror images" in a high-dimensional space.
- These mirror images will also be evident in any marginal distribution, including two-dimensional marginal distributions.


t=0
t=600
t=1200
t=1800
t=2400
t=3000
t=3600
t=4200
t=4800
log(sigma1)
log(sigma2)

"Speedup Factors"

- EGARCH(2,3) model
- SMC with $J = 16$, $N = 4,096$, $J \cdot N = 262,144$, $R = 55$
- MCMC random walk Metropolis, iterated until SMC numerical standard error is matched

	Marginal Likelihood	Posterior moments Type 1	Type 2
SMC particles	262,144	262,144	262,144
SMC time	1,796	1,796	1,796
MCMC iterations	2.772×10^9	1.087×10^6	$\longrightarrow \infty$
MCMC time	$\approxeq 7.18 \times 10^7$	28,167	$\longrightarrow \infty$
"Speedup factor"	$\approxeq 40,000$	15.68	$\longrightarrow \infty$

Conclusion

- The sequential posterior simulator is very closely related to simulated annealing methods for function optimization
- Essentially the same algorithm can be used to:
  - Conduct Bayesian inference
  - Conduct classical inference (extremum estimators)
  - Solve optimization problems (decision making)
    - Especially attractive for irregular functions, multiple local modes
  - Solve complex dynamic models
    - Can be used to determine existence and uniqueness
- In the next 10 - 20 years
  - Cheap parallel computing on graphical processing
  - Single-instruction multiple-data compatible algorithms will become vital to the computational infrastructure of statistics.