



Centre Interuniversitaire sur le Risque,
les Politiques Économiques et l'Emploi

Cahier de recherche/Working Paper **15-18**

Evolutionary Sequential Monte Carlo Samplers for Change-point Models

Arnaud Dufays

Septembre/September 2015

Dufays : Département d'économie, 2216 Pavillon J.-A.-DeSève, Université Laval, Québec, Canada G1V 0A6; and CIRPÉE; website : <https://sites.google.com/site/websiteofarnauddufays/arnaud.dufays@ecn.ulaval.ca>

Abstract:

Sequential Monte Carlo (SMC) methods are widely used for non-linear filtering purposes. Nevertheless the SMC scope encompasses wider applications such as estimating static model parameters so much that it is becoming a serious alternative to Markov-Chain Monte-Carlo (MCMC) methods. Not only SMC algorithms draw posterior distributions of static or dynamic parameters but additionally provide an estimate of the marginal likelihood. The tempered and time (TNT) algorithm, developed in the paper, combines (off-line) tempered SMC inference with on-line SMC inference for drawing realizations from many sequential posterior distributions without experiencing a particle degeneracy problem. Furthermore, it introduces a new MCMC rejuvenation step that is generic, automated and well-suited for multi-modal distributions. As this update relies on the wide heuristic optimization literature, numerous extensions are already available. The algorithm is notably appropriate for estimating Change-point models. As an example, we compare Change-point GARCH models through their marginal likelihoods over time.

Keywords: Bayesian inference, Sequential Monte Carlo, Annealed Importance sampling, Change-point models, Differential Evolution, GARCH models

JEL Classification: C11, C15, C22, C58

1 Introduction

Sequential Monte Carlo (SMC) algorithm is a simulation-based procedure used in Bayesian framework for drawing distributions. Its core idea relies on an iterated application of the importance sampling technique to a sequence of distributions converging to the distribution of interest¹. For many years, on-line inference was the most relevant applications of SMC algorithms. Indeed, one powerful advantage of sequential filtering consists in being able to update the distributions of the model parameters in light of new coming data (hence the term on-line) allowing for important time saving compared to off-line methods such as the popular Markov-Chain Monte-Carlo (MCMC) procedure that requires a new estimation based on all the data at each new observation entering in the system. Other SMC features making it very promising are an intuitive implementation based on the importance sampling technique (Geweke (1989), Smith and Gelfand (1992) and Gordon, Salmond, and Smith (1993)) and a direct computation of the marginal likelihood (i.e. the normalizing constant of the targeted distribution, see e.g. Chib, Nardari, and Shephard (2002)).

Recently, the SMC algorithms have been applied to inference of static parameters, field in which the MCMC algorithm excels. Neal (1998) provides a relevant improvement in this direction by building a SMC algorithm, named annealed importance sampling (AIS), that sequentially evolves from the prior distribution to the posterior distribution using a tempered function, which basically consists in gradually introducing the likelihood information into the sequence of distributions by means of an increasing function. To preclude particles degeneracies, he uses MCMC kernels at each SMC iteration. Few years later, Chopin (2002) proposes an Iterated Batch Importance Sampling (IBIS) SMC algorithm, a special case of the Re-sample Move (RM) algorithm of Gilks and Berzuini (2001), which sequentially evolves over time and adapts the posterior distribution using the previous approximate distribution. Again, an MCMC move (and a re-sampling step) is used for diversifying the particles. The SMC sampler (see Del Moral, Doucet, and Jasra (2006)) unifies, among others, these SMC algorithms in a theoretical framework. It is shown that the methods of Neal (1998) and Gilks and Berzuini (2001) arise as special cases with a specific choice of the 'backward kernel function' introduced in their paper. These researches have been followed by empirical works (see Jasra, Stephens, and Holmes (2007), Jasra, Stephens, Doucet, and Tsagaris (2011) and

¹whereas the sequence does not have to evolve over the time domain

Jeremiah, Sisson, Marshall, Mehrotra, and Sharma (2011)) where it is demonstrated that the SMC mixing properties often dominate MCMC methods based on a single Markov-chain. Nowadays papers are devoted to build self-adapting SMC samplers by automatically tuning the MCMC kernels (e.g. Fearnhead and Taylor (2013)), by marginalizing the state vector (in a state space specification) using the particle MCMC framework (e.g. Fulop and Li (2013) and Chopin, Jacob, and Papaspiliopoulos (2013)), to construct efficient SMC samplers for parallel computations (see Durham and Geweke (2012)) or to simulate from complex multimodal posterior distributions (e.g. Herbst and Schorfheide (2012)).

In this paper, we document a generic SMC inference for Change-point models that can additionally be updated through time. For example, in a model comparison context the standard methodology consists in repeating estimations of the parameters given an evolving number of observations. In circumstances where the Bayesian parameter estimation is highly demanding as it is usually the case for complex models and where the number of available observations is huge, this iterative methodology can be too intensive. Change-point (CP) Generalized Autoregressive Conditional Heteroskedastic (GARCH) processes may require several hours for one inference (e.g. Bauwens, Dufays, and Rombouts (2013)). A recursive forecast exercise on many observations is therefore out of reach. Our first contribution is a new SMC algorithm, called the tempered and time (TNT), which exhibits the AIS, the IBIS and the RM samplers as special cases. It innovates by switching over tempered and time domains for estimating posterior distributions. For instance, it firstly iterates from the prior to the posterior distributions by means of a sequence of tempered posterior distributions. It then updates in the time dimension the slightly different posterior distributions by sequentially adding new observations, each SMC step providing all the forecast summary statistics relevant for comparing models. The TNT algorithm combines the tempered approach of Neal (1998) with the IBIS algorithm (IBIS) of Chopin (2002) if the model parameters are static or with the RM method of Gilks and Berzuini (2001) if their support evolves with the SMC updates. Since all these methods are built on the same SMC steps (re-weighting, re-sampling and re-juvenating) and the same SMC theory, the combination is achieved without efforts.

The proposed methodology exhibits several advantages over SMC algorithms that directly iterate on the time domain (Gilks and Berzuini (2001) and Chopin (2002)). In fact, these

algorithms may experience high particle discrepancies. Although the problem is more acute for models where the parameter space evolves through time, it remains an issue for models with static parameters at the very first SMC steps. To quote Chopin (2002) (p 546) :

Note that the particle system may degenerate strongly in the very early stages, when the evolving target distribution changes the most[...].

The combination of tempered and time SMC algorithms allows for limiting this particle discrepancy observed at the early stage since the first posterior distribution of interest is estimated by taking into account more than a few observations. One advantage of using a sequence of tempered distributions to converge to the posterior distribution consists in the number of SMC steps that can be used. Compared to SMC algorithms that directly iterate on time domain where the sequence of distributions is obviously defined by the number of data, the tempered approach allows for choosing this sequence of distribution and for targeting the posterior distribution of interest by using as many bridging distributions as needed.

Many SMC algorithms rely on MCMC kernels to rejuvenate the particles. The TNT sampler makes no exception. We contribute by proposing new generic MCMC kernels based on the heuristic optimization literature. These kernels are well appropriated in the SMC context as they build their updates on particles interactions. We start by emphasizing that the Differential Evolution Adaptive Metropolis (DREAM, see Vrugt, ter Braak, Diks, Robinson, Hyman, and Higdon (2009)), the walk move (see Christen and Fox (2010)) and the stretch one (see Foreman-Mackey, Hogg, Lang, and Goodman (2013)) separately introduced in the statistic literature as generic Metropolis-Hastings proposals are in fact standard mutation rules of the Differential Evolution (DE) optimization. From this observation, we propose seven new MCMC updates based on the heuristic literature and emphasize that many other extensions are possible. The proposed MCMC kernel is adapted for continuous parameters. Consequently, discrete parameters such as the break parameters of Change-point models cannot directly be inferred from our algorithm. To solve this issue, we transform the break parameters into continuous ones which make them identifiable up to a discrete value. To illustrate the potential of the TNT sampler, we compare CP-GARCH models differing by their number of regimes on the S&P 500 daily percentage returns.

The paper is organized as follows. Section 2 presents the SMC algorithm as well as its theoretical derivation. Section 3 introduces the different Metropolis-Hastings proposals which compose what we call the Evolutionary MCMC. We then detail a simulation exercise on the CP-GARCH process in section 4. Eventually we study the CP-GARCH performance on the S&P 500 daily percentage returns in section 5. Section 6 concludes.

2 Off-line and On-line inferences

We first theoretically and practically introduce the tempered and time (TNT) framework. To ease the discussion, let consider a standard state space model:

$$y_t = f(\theta, s_t, \omega_t) \quad (1)$$

$$s_t = g(\theta, s_{t-1}, v_t) \quad (2)$$

where s_t is a random variable driven by a Markov chain and the functions $f(-)$ and $g(-)$ are deterministic given their arguments. The observation y_t belongs to the set $y_{1:T} = \{y_1, \dots, y_T\}$ with T denoting the sample size and is assumed to be independent conditional to the state s_t and θ with distribution $f(y_t|\theta, s_t)$. The innovations ω_t and v_t are mutually independent and stand for the noise of the observation/state equations. The model parameters included in θ do not evolve over time (i.e. they are static). Let denote the set of parameters at time t by $x_t = \{\theta, s_{1:t}\}$ defined on the measurable space E_t .

We are interested in estimating many posterior distributions starting from $\pi(x_\tau|y_{1:\tau})$, where $\tau \ll T$, until T . The SMC algorithm approximates these posterior distributions with a large (dependent) collection of M weighted random samples $\{W_t^i, x_t^i\}_{i=1}^M$ where $W_t^i > 0$ and $\sum_{i=1}^M W_t^i = 1$ such that as $M \rightarrow \infty$, the empirical distribution converges to the posterior distribution of interest, meaning that for any $\pi(x_t|y_{1:t})$ -integrable function $g : E_t \rightarrow \mathfrak{R}$

:

$$\sum_{i=1}^M W_t^i g(x_t^i) \rightarrow \int_{E_t} g(x_t) \pi(x_t|y_{1:t}) dx_t \quad \text{almost surely.}$$

The TNT method combines an enhanced Annealed Importance sampling² (AIS, see Neal

²enhanced in the sense that the AIS incorporates a re-sampling step

(1998)) with the Re-sample Move (RM) SMC inference of Gilks and Berzuini (2001)³. To build the TNT algorithm, we rely on the theoretical paper of Del Moral, Doucet, and Jasra (2006) that unifies the two SMC methods into one SMC framework called 'SMC sampler'. The TNT algorithm first estimates an initial posterior distribution, namely $\pi(x_\tau|y_{1:\tau})$, by an enhanced AIS (E-AIS) algorithm and then switches from the tempered domain to the time domain and sequentially updates the approximated distributions from $\pi(x_\tau|y_{1:\tau})$ to $\pi(x_T|y_{1:T})$ by adding one by one the new observations. We now begin by mathematically deriving the validity of the SMC algorithms under the two different domains and by showing that they are particular cases of the SMC sampler. The practical algorithm steps are given afterward (see subsection 2.3).

2.1 E-AIS : the tempered domain

The first phase, carried out by an E-AIS, creates a sequence of probability measures $\{\pi_n\}_{n=0}^p$ that are defined on measurable spaces $\{E_n, \xi_n\}$, where $E_n = E_{n+1} = E \ni x_\tau$, $n \in \{0, 1, \dots, p\}$ is a counter and does not refer to 'real time', p denotes the number of posterior distribution estimations and π_p coincides with the first posterior distribution of interest $\pi(x_\tau|y_{1:\tau})$. The sequence distribution, used as bridge distributions, is defined as $\pi_n(x_n|y_{1:\tau}) = \gamma(y_{1:\tau}|x_n)^{\phi(n)} f(x_n) / Z_n$ where $Z_n = \int_E \gamma(y_{1:\tau}|x_n)^{\phi(n)} f(x_n) dx_n$ denotes the normalizing constant, $\gamma(y_{1:\tau}|x_n)$ and $f(x_n)$ respectively are the likelihood function and the prior density of the model. Through an increasing function $\phi(n)$ respecting the bound conditions $\phi(0) = 0$ and $\phi(p) = 1$, the E-AIS artificially builds a sequence of distributions that converges to the posterior distribution of interest.

Remark 1: The E-AIS makes up a sequence of random variables $\{x_n\}_{n=0}^p$ that exhibit the same support E also shared by x_τ . The random variable x_τ coincides with x_p since $\phi(p) = 1$.

The E-AIS is merely a sequential importance sampling technique where the draws of a proposal distribution η_n with MCMC kernel combinations are used to approximate the next posterior distribution π_{n+1} , the difficulty lying in specifying the sequential proposal distribution. Del Moral, Doucet, and Jasra (2006) theoretically develop a coherent framework for

³The IBIS algorithm being a particular case.

helping to choose a generic sequence of proposal distributions.

In the SMC sampler framework, we augment the support of the posterior distribution ensuring that the targeted posterior distribution marginally arises :

$$\begin{aligned}\tilde{\pi}_n(x_{1:n}) &= \pi_n(x_n) \prod_{k=2}^n L_k(x_{k-1}|x_k), \\ &= \frac{\gamma_n(x_n)}{Z_n} \prod_{k=2}^n L_k(x_{k-1}|x_k),\end{aligned}$$

where $\gamma_n(x_n) = \gamma(y_{1:T}|x_n)^{\phi(n)} f(x_n)$, $Z_n = \int_E \gamma(y_{1:T}|x_n)^{\phi(n)} f(x_n) dx_n$ is the normalizing constant, and $L_k(x_{k-1}|x_k)$ is a backward MCMC kernel such that $\int_E L_k(x_{k-1}|x_k) dx_{k-1} = 1$.

By defining a sequence of proposal distributions as

$$\eta_n(x_{1:n}) = f(x_1) \prod_{k=2}^n K_k(x_k|x_{k-1}),$$

where $K_k(x_k|x_{k-1})$ is an MCMC kernel with stationary distribution π_k such that it verifies $\pi_k(x_k) = \int_E K_k(x_k|x_{k-1}) \pi_k(x_{k-1}) dx_{k-1}$, we derive a recursive equation of the importance weight :

$$\begin{aligned}w_n(x_{1:n}) &= \frac{\gamma_n(x_n) \prod_{k=2}^n L_k(x_{k-1}|x_k)}{Z_n f(x_1) \prod_{k=2}^n K_k(x_k|x_{k-1})}, \\ &= w_{n-1}(x_{1:n-1}) \frac{Z_{n-1} \gamma_n(x_n) L_n(x_{n-1}|x_n)}{Z_n \gamma_{n-1}(x_{n-1}) K_n(x_n|x_{n-1})}.\end{aligned}$$

For a smooth increasing tempered function $\phi(n)$, we can argue that π_{n-1} will be close to π_n .

We therefore define the backward kernel by detailed balance argument as

$$L_n(x_{n-1}|x_n) = \frac{\pi_n(x_{n-1})}{\pi_n(x_n)} K_n(x_n|x_{n-1}). \quad (3)$$

It gives the following weights :

$$\begin{aligned}w_n(x_{1:n}) &= w_{n-1}(x_{1:n-1}) \frac{Z_{n-1} \gamma_n(x_{n-1})}{Z_n \gamma_{n-1}(x_{n-1})}, \\ &\propto w_{n-1}(x_{1:n-1}) \gamma(y_{1:T}|x_{n-1})^{\phi_n - \phi_{n-1}}.\end{aligned} \quad (4)$$

The normalizing constant Z_n is approximated as

$$\frac{Z_n}{Z_{n-1}} \approx \sum_{i=1}^M W_{n-1}^i \frac{\gamma_n(x_{n-1})}{\gamma_{n-1}(x_{n-1})},$$

where $W_{n-1}^i = w_{n-1}(x_{1:n-1}^i) / \sum_{j=1}^M w_{n-1}(x_{1:n-1}^j)$, i.e. the normalized weight.

The E-AIS requires to tune many parameters : an increasing function $\phi(n)$, MCMC kernels of invariant distribution $\pi_n(\cdot)$, a number of particles M , of iterations p , of MCMC steps J . Tuning these parameters can be difficult. Some guidance are given in Herbst and Schorfheide (2012) for DSGE models. For example, they propose a quadratic tempered function $\phi(n)$. It slowly increases for small values of n and the step becomes larger and larger as n tends to p . In this paper, the TNT algorithm generically adapts the different user-defined parameters and belongs to the class of adaptive SMC algorithms. It automatically adjusts the tempered function with respect to an efficiency measure as it was proposed by Jasra, Stephens, Doucet, and Tsagaris (2011). By doing so, we preclude the difficult choice of the function $\phi(n)$ and the number of iteration p . The number of MCMC steps J will be controlled by the acceptance rate exhibited by the MCMC kernels. The choice of MCMC kernels and the number of particles are discussed later (see section 3).

2.2 The Re-sample Move algorithm : the time domain

Once we have a set of particles that approximates the first posterior distribution of interest $\pi(x_\tau|y_{1:\tau})$, a second phase takes place. Firstly, let assume that the support of x_t does not evolve over time (i.e. $x_t \in E \ \forall \ t$). In this context, the SMC sampler framework shortly reviewed here for the tempered domain still applies. Let define the following distributions :

$$\begin{aligned} \pi_t(x_t) &= \pi(x_t|y_{1:t}), \\ \tilde{\pi}_t(x_{1:t}) &= \pi_t(x_t) \prod_{k=2}^t L_k(x_{k-1}|x_k), \\ \eta_t(x_{1:t}) &= f(x_1) \prod_{k=2}^t K_k(x_k|x_{k-1}), \\ L_k(x_{k-1}|x_k) &= \frac{\pi_k(x_{k-1})}{\pi_k(x_k)} K_k(x_k|x_{k-1}). \end{aligned}$$

Then the weight equation of the SMC sampler are equal to :

$$w_t(x_{1:t}) = w_{t-1}(x_{1:t-1}) \frac{\pi_t(x_{t-1})}{\pi_{t-1}(x_{t-1})}, \quad (5)$$

which is exactly the weight equation of the IBIS algorithm (see Chopin (2002), step 1, page 543).

Let now consider the more difficult case where a subset of the support of x_t evolves with t such as $x_t = \{x_{t-1}, s_t\} = \{\theta, s_{1:t-1}, s_t\}$ (see state space model equations (1),(2)) meaning that $\forall t \in [1, T], x_t \in E_t$ and $E_{t-1} \subset E_t$. The previous method cannot directly be applied (due to the backward kernel) but with another choice of the kernel functions, the SMC sampler also operates. Let define the following distribution :

$$\pi_t(x_t) = \pi(x_t|y_{1:t}), \quad (6)$$

$$\tilde{\pi}_t(x_{1:t}, x_{2:t}^*) = \pi_t(x_t) \prod_{k=2}^t L_k(x_{k-1}, x_k^*|x_k), \quad (7)$$

$$\eta_t(x_{1:t}, x_{2:t}^*) = f(x_1) \prod_{k=2}^t \tilde{K}_k(x_k, x_k^*|x_{k-1}), \quad (8)$$

$$\tilde{K}_k(x_k, x_k^*|x_{k-1}) = \tilde{q}_k(x_k^*|x_{k-1})K_k(x_k|x_k^*), \quad (9)$$

$$L_k(x_{k-1}, x_k^*|x_k) = q_k(x_{k-1}|x_k^*)K_k(x_k^*|x_k), \quad (10)$$

$$K_k(x_t^*|x_t) = \frac{\pi_k(x_k^*)K_k(x_k|x_k^*)}{\pi_k(x_k)} \quad \text{by detailed balance argument.} \quad (11)$$

To deal with the time-varying dimension of x_t , we augment the support of the artificial sequence of distributions by new random variables (see $x_{2:t}^*$ in equation (7)) while ensuring that the posterior distribution of interest $\pi_t(x_t)$ marginally arises. Sampling from the proposal distribution $\eta_t(x_{1:t}, x_{2:t}^*)$ is achieved by drawing from the prior distribution and then by sequentially sampling from distributions $\tilde{K}_k(x_k, x_k^*|x_{k-1})$, which is composed by a user-defined distribution $\tilde{q}_k(x_k^*|x_{k-1})$ and an MCMC kernel exhibiting $\pi_k(x_k)$ as invariant distribution.

Under this framework, the weight equation of the SMC sampler becomes :

$$w_t(x_{1:t}, x_{2:t}^*) = w_{t-1}(x_{1:t-1}, x_{2:t-1}^*) \frac{\pi_t(x_t^*)q_t(x_{t-1}|x_t^*)}{\pi_{t-1}(x_{t-1})\tilde{q}_t(x_t^*|x_{t-1})}. \quad (12)$$

By setting the distributions $q_t(x_{t-1}|x_t^*) = \delta_{\{x_{t-1}=\theta^*, s_{1:t-1}^*\}}$, where δ_i denotes the probability measure concentrated at i , and $\tilde{q}_t(x_t^*|x_{t-1}) = \nu_t(s_t^*|x_{t-1})\delta_{\{\theta^*, s_{1:t-1}^*=x_{t-1}\}}$, we recover the weight equation of Gilks and Berzuini (2001) (see eq. (20), page 135) :

$$w_t(x_{1:t}, x_{2:t}^*) = w_{t-1}(x_{1:t-1}, x_{2:t-1}^*) \frac{\pi_t(x_{t-1}, s_t^*)}{\pi_{t-1}(x_{t-1})\nu_t(s_t^*|x_{t-1})}. \quad (13)$$

Like in Gilks and Berzuini (2001), only the distribution $\nu_t(\cdot)$ has to be specified. For example, it can be set either to the prior distribution or the full conditional posterior distribution (if the latter exhibits a closed form).

Remark 2: The division $\frac{\pi(x_t|y_{1:t})}{\pi(x_{t-1}|y_{1:t-1})}$ appearing in weight equations ((5),(13)) can be reduced to $\frac{Z_{t-1}}{Z_t}\gamma(y_t|x_t, y_{1:t-1})f(s_t|x_{t-1})$ which highly limits the computational cost of the weights.

2.3 The TNT algorithm

The algorithm initializes the M particles using the prior distributions, sets each initial weight $\{W_0^i\}_{i=1}^M$ to $W_0^i = \frac{1}{M}$ and then iterates from $n = 1, \dots, p, p+1, \dots, p+(T-\tau)+1$ as follows

- **Correction step:** $\forall i \in [1, M]$, Re-weight each particle with respect to the n th posterior distribution

- If in tempered domain ($n \leq p$) :

$$\tilde{w}_n^i = \gamma(y_{1:\tau}|x_{n-1}^i)^{\phi(n)-\phi(n-1)} \quad (14)$$

- If in time domain ($n > p$) and the parameter space does not evolve over time (i.e. $E_{n-1} = E_n$) :

$$\tilde{w}_n^i = \frac{\gamma(y_{1:\tau+n-p}|x_{n-1}^i)}{\gamma(y_{1:\tau+n-p-1}|x_{n-1}^i)} = \gamma(y_{\tau+n-p}|x_{n-1}^i, y_{1:\tau+n-p-1}) \quad (\text{see remark 2}) \quad (15)$$

- If in time domain ($n > p$) and the parameter space increases (i.e. $E_{n-1} \subset E_n$) :

Set $x_n^i = \{x_{n-1}^i, s_n^i\}$ with $s_n^i \sim \nu_n(\cdot|x_{n-1}^i)$.

$$\tilde{w}_n^i = \frac{\gamma(y_{1:\tau+n-p}|x_n^i)f(x_n^i)}{\gamma(y_{1:\tau+n-p-1}|x_{n-1}^i)f(x_{n-1}^i)\nu_n(s_n^i|x_{n-1}^i)} \quad (16)$$

Compute the unnormalized weights : $\tilde{W}_n^i = \tilde{w}_n^i W_{n-1}^i$.

Normalize the weights : $W_n^i = \frac{\tilde{W}_n^i}{\sum_{j=1}^M \tilde{W}_n^j}$

- **Re-sampling step:** Compute the Effective Sample Size (ESS) as

$$ESS = \frac{1}{\sum_{i=1}^M (W_n^i)^2}$$

If $ESS < \kappa$ where κ is a user-defined threshold then re-sample the particles and reset the weight uniformly.

- **Mutation step:** $\forall i \in [1, M]$, run J steps of an MCMC kernel with invariant distribution $\pi_n(x_n|y_{1:\tau})$ for $n \leq p$ and $\pi(x_{\tau+n-p}|y_{1:\tau+n-p})$ for $n > p$.

Remark 3: According to the algorithm derivation, note that the mutation step is not required at each SMC iteration.

When the parameter space does not change over time (i.e. tempered or time domains with $E_{n-1} = E_n$), the algorithm reduces to the SMC sampler with a specific choice of the backward kernel (see equation (3), more discussions in Del Moral, Doucet, and Jasra (2006)) that implies that $\pi_{n-1}(\cdot)$ must be close to $\pi_n(\cdot)$ for non-degenerating estimations. The backward kernel is introduced for avoiding the choice of an importance distribution at each iteration of the SMC sampler. This specific choice of the backward kernel does not work for model where the parameter space increases with the sequence of posterior distributions (hence the use of a second weighting scheme when $n > p$, see equation (12)) but the algorithm also reduces to a SMC sampler with another backward kernel choice (see (10)). In empirical applications, we first estimate an off-line posterior distribution with fixed parameters and then by just switching the weight equation, we sequentially update the posterior distributions by adding new observations. This two phases preclude the particle degeneration that may occur at the early stage of the SMC algorithms that directly iterate on time such as the IBIS and the RM algorithms. The tempered function $\phi(n)$ allows for converging to the first targeted posterior distribution as slowly as we want. Indeed, as we are not constraint by the time domain, we can sequentially iterate as much as needed to get rid of degeneracy problems. The choice of the tempered function $\phi(n)$ is therefore relevant. In the spirit of a black-box algorithm as is the IBIS one, the section 2.4 shows how the TNT algorithm automatically adapts it at each SMC iteration.

During the second phase (i.e. updating the posterior distribution through time), one may observe high particle discrepancies especially when the space of the parameters evolves over

time⁴. In that case, one can run an entire E-AIS on the data $y_{1:\tau+n-p}$ when a degeneracy issue is detected (i.e. the ESS falls below a user-defined value $\kappa_1 < \kappa$). The adaption of the tempered function (discussed in the next section) makes the E-AIS faster than usual since it reduces the number of iteration p at its minimum given the ESS threshold κ . Controlling for degeneracy issue is therefore automated and a minimal number of effective sample size is ensured at each SMC iteration.

2.4 Adaption of the tempered function

Previous works on SMC samplers usually provide a tempered function $\phi(n)$ obtained by empirical trials⁵, making these functions model-dependent. Jasra, Stephens, Doucet, and Tsagaris (2011) innovate by proposing a generic choice of $\phi(n)$ that only requires a few more codes. The E-AIS correction step (see equation (14)) of iteration n is modified as follows

1. Find $\bar{\phi}_n$ such that

$$\frac{1}{\sum_{i=1}^M (W_n^i)^2} = 0.95 \text{ESS}_{n-1},$$

where ESS_{n-1} refers to the Effective Sample Size of the previous SMC iteration, $W_n^i = \frac{\tilde{W}_n^i}{\sum_{j=1}^M \tilde{W}_n^j}$ is the normalized weights and the unnormalized \tilde{W}_n^i depends on $\bar{\phi}_n$ as

$$\tilde{w}_n^i = \frac{\gamma(y_{1:\tau}|x_{n-1}^i)^{\bar{\phi}_n}}{\gamma(y_{1:\tau}|x_{n-1}^i)^{\bar{\phi}_{n-1}}}.$$

2. Compute the normalized weights $\{W_n^i\}_{i=1}^M$ under the value of $\bar{\phi}_n$.

Roughly speaking, we find the value $\bar{\phi}_n$ that makes the ESS criterion close to the previous one in order to keep the artificial sequence of distributions very similar as required by the choice of the backward kernel 3.

Because the tempered function is adapted on the fly using the SMC history, the usual SMC asymptotic results do not apply. Del Moral, Doucet, and Jasra (2012) and Beskos, Jasra, and

⁴Theorem 1 in Chopin (2002) ensures that with a sufficiently large number of particles M , any relative precision of the importance sampling can be obtained if the number of observations already covered is large enough in the IBIS context.

⁵a piecewise cooling linear function for Del Moral, Doucet, and Jasra (2006) and a quadratic function for Herbst and Schorfheide (2012).

Thiery (2013) provide asymptotic results by assuming that the adapted tempered function converges to the optimal one (if it exists).

3 Choice of MCMC kernels

The MCMC kernel is the most computational demanding step of the algorithm and determines the posterior support exploration, making its choice very relevant. Chopin (2002) emphasizes that the IBIS algorithm is designed to be a true 'black box' (i.e. whose the sequential steps are not model-dependent), reducing the task of the practitioner to only supply the likelihood function and the prior densities. For this purpose, a natural choice of MCMC kernel is the Metropolis-Hastings with an independent proposal distribution whose summary statistics are derived using the particles of the previous SMC step and the weight of the current step. The IBIS algorithm uses an independent Normal proposal. It is worth noting that this 'black box' structure is still applicable in this framework that combines SMC iterations on tempered and time domains.

Nevertheless the independent Metropolis-Hastings kernel may perform poorly at the early stage of the algorithm if the posterior distribution is well behaved and at any time otherwise. We rather suggest using a new adaptive Metropolis algorithm of random walk (RW) type that is generic, fully automated, suited for multi-modal distributions and that dominates most of other RW alternatives in terms of sampling efficiencies. The algorithm is inspired from the heuristic Differential Evolution (DE) optimization literature (for a review, see Das and Suganthan (2011)).

The DE algorithms have been designed to solve optimization problems without requiring derivatives of the objective function. The algorithms are initiated by randomly generating a set of parameter values. Afterward, relying on a mutation rule and a cross-over (CR) probability, these parameters are updated in order to explore the space and to converge to the global optimum. The mutation equation is usually linear with respect to the parameters and the CR probability determines the number of parameters that changes at each iteration. The first DE algorithm dates back to Storn and Price (1997). Nowadays, numerous alternatives based on this principle have been designed and many of them display a different mutation rule. Considering a set of parameters $\{x_t^i\}_{i=1}^M$ lying in \mathfrak{R}^d , the standard algorithm operates by sequentially updating each parameter given the other ones. For a specific parameter x_t^j , the

mutation equation to obtain a new value \tilde{x}_t are typically chosen among the following ones

$$\tilde{x}_t = x_t^j + F\left(\sum_{g=1}^2 x_t^{r_1(g)} - \sum_{h=1}^2 x_t^{r_2(h)}\right), \quad (17)$$

$$\tilde{x}_t = x_t^j + F(x_t^j - x_t^{\text{best}}) + F(x_t^{r_1} - x_t^{r_2}), \quad (18)$$

$$\tilde{x}_t = x_t^{\text{best}} + F\left(\sum_{g=1}^2 x_t^{r_1(g)} - \sum_{h=1}^2 x_t^{r_2(h)}\right), \quad (19)$$

where $i \neq r_1(g), r_2(h)$; $r_1(\cdot)$ and $r_2(\cdot)$ stand for random integers uniformly distributed on the support $[1, M]_{-i}$ and it is required that $r_1(g) \neq r_2(h)$ when $g = h$, F is a fixed parameter and x_t^{best} denotes the parameter related to the highest objective function in the swamp. Then, for each element of the new vector \tilde{x}_t , the CR step consists in replacing its value with the one of x_t^j according to a fixed probability.

The DE algorithm is appealing in MCMC frameworks as it has been built up to explore and find global optima of complex objective functions. However, the DE method has to be adapted if one wants to draw realizations from a complex distribution. To employ the mutation equations (17), (18) and (19) into an MCMC algorithm, we need to insure that the detailed balance is preserved, that the Markov-chain is ergodic with a unique stationary distribution and that this distribution is the targeted one. To do so, we slightly modify the mutation equations as follows

$$\tilde{x}_t = x_t^j + F(\delta, d)\left(\sum_{g=1}^{\delta} x_t^{r_1(g)} - \sum_{h=1}^{\delta} x_t^{r_2(h)}\right) + \zeta, \quad (20)$$

$$\tilde{x}_t = x_t^j + Z_W\left(x_t^j - \frac{\sum_{g=1}^{\delta} x_t^{r_1(g)}}{\delta}\right), \quad (21)$$

$$\tilde{x}_t = \frac{\sum_{g=1}^{\delta} x_t^{r_1(g)}}{\delta} + Z_{\text{Stretch}}\left(x_t^j - \frac{\sum_{g=1}^{\delta} x_t^{r_1(g)}}{\delta}\right), \quad (22)$$

in which $\zeta \sim N(0, \eta_x^2 I)$; $\delta \sim U[1, 3]$, $F(\delta, d)$ is a fixed parameter, Z_{RW} and Z_{Stretch} are random variables driven by two different distributions defined below.

These three update rules (20), (21) and (22) are valid in an MCMC context and have been separately proposed in the literature. The first equation (20) refers to the Differential Evolution Adaptive Metropolis (DREAM) proposal distribution of Vrugt, ter Braak, Diks, Robinson, Hyman, and Higdon (2009) and is the MCMC analog of the DE mutation (17). In their paper, it is shown that the proposal distribution is symmetric and so that the acceptance ratio is independent of the proposal density. Also, they fix η_x to a very small value (such

as $1e-4$) and $F(\delta, d)$ to $2.38/\sqrt{2\delta d}$ because it constitutes the asymptotic optimal choice for Normal posterior distributions as demonstrated in Roberts and Rosenthal (2001).⁶ Since the posterior distribution is rarely Normal, we prefer adapting it from one SMC iteration to another so as the scale parameter is fixed during the entire MCMC moves of each SMC step. The adapting procedure is detailed below. Importantly, Vrugt, ter Braak, Diks, Robinson, Hyman, and Higdon (2009) provide empirical evidence that the DREAM equation dominates most of the other RW alternatives (including the optimal scaling and the adaptive ones) in terms of sampling efficiencies.

The second equation (21) is an adapted version of the walk move of Christen and Fox (2010) and can be thought as the MCMC equivalence of the mutation (18).⁷ When the density $g_W(\cdot)$ of Z_W verifies $g_W(-z/(1+z)) = (1+z)g_W(z)$, it can be shown that the proposal parameter \tilde{x}_t is accepted with a probability given by

$$\min\left\{\frac{|1 + Z_W|^{d-1} \pi(\tilde{x}_t | y_{1:t})}{\pi(x_t^j | y_{1:t})}, 1\right\}. \quad (23)$$

As in their paper, we set the density to $g_W(z) \propto 1/\sqrt{1+z}$ if $z \in [-\frac{a_W}{1+a_W}, a_W]$ and zero otherwise. The cumulative density function, its inverse and the first two moments of the distributions are given by

$$\begin{aligned} F_{Z_W}(x) &= 1 - \frac{(a_W + 1)^{1/2} - (x + 1)^{1/2}}{(a_W + 1)^{1/2} - (a_W + 1)^{-1/2}}, \\ F_{Z_W}^{-1}(u) &= -1 + [(a_W + 1)^{-1/2} + u((a_W + 1)^{1/2} - (a_W + 1)^{-1/2})]^2, \\ E(Z_W) &= \frac{a_W^2}{3(a_W + 1)}, \\ V(Z_W) &= a_W^2 \frac{4a_W^2 + 15a_W + 15}{45(a_W + 1)^2}. \end{aligned}$$

In the seminal paper of the walk move, the parameter a_W is set to 2. However, we rather suggest solving the equation $V(Z_W) = 2.38/\sqrt{2\delta d}$ in order to obtain the optimal value of a_W . Note that equation (21) is slightly different from the standard walk move of Christen and Fox (2010) in the sense that the random parameter δ can be greater than one and that only one realization of Z_W is generated in order to update an entire new vector x_t^j . The

⁶in the sense of $M \rightarrow \infty$

⁷The best particle is replaced by an average over the particles $\frac{\sum_{g=1}^{\delta} x_t^{r1(g)}}{\delta}$ which further diversifies the proposed parameters.

latter modification is motivated by the success of the novel MCMC based on deterministic proposals (see the Transformation-based MCMC of Dutta and Bhattacharya (2014)) and by the DREAM update which also exhibits one (fixed) parameter $F(\delta, d)$ to propose the entire new vector.

Lastly, the third equation (22) corresponds to the stretch move proposed in Christen and Fox (2010) and improved by Foreman-Mackey, Hogg, Lang, and Goodman (2013). The probability of accepting the proposal \tilde{x}_t is

$$\min\left\{\frac{|Z_S|^{d-1}\pi(\tilde{x}_t|y_{1:t})}{\pi(x_t^j|y_{1:t})}, 1\right\}, \quad (24)$$

when the density $g_S(\cdot)$ of Z_S verifies $zg_S(z) = g_S(z^{-1})$. We adopt the same density function as in their paper which is given by $g_S(z) = 1/\sqrt{z}$ for $z \in [1/a_S, a_S]$, $a_S \in \mathfrak{R}^+$ and zero otherwise. The corresponding cumulative density function, its inverse, the expectation and the variance are analytically tractable and given by

$$\begin{aligned} F_{Z_S}(x) &= \frac{\sqrt{a_S x} - 1}{a - 1}, \\ F_{Z_S}^{-1}(u) &= \frac{(u(a_S - 1) + 1)^2}{a_S}, \\ E(Z_S) &= \frac{a_S + a_S^{-1} + 1}{3}, \\ V(Z_S) &= \frac{(a_S - 1)^2(4a_S^2 + 7a_S + 4)}{45a_S}. \end{aligned}$$

In the standard stretch move, the parameter a_S is set to 2.5. Like the DREAM algorithm, the stretch move has been proven to be a powerful generic MCMC approach to generate complex posterior distributions. The method is becoming very popular in astrophysics (see references in Foreman-Mackey, Hogg, Lang, and Goodman (2013)).

Once it is recognized that all these updates are also involved in the DE optimization problems, incorporating many other techniques from the latter becomes straightforward. To highlight the potential, we extend the DREAM, the walk and the stretch moves by proposing new update equations that are derived from the trigonometric move, the standard DE mutation and the firefly optimization.

In the DE literature, Fan and Lampinen (2003) suggests using a trigonometric mutation equation based on three random parameters $x_t^{r1}, x_t^{r2}, x_t^{r3}$ and their corresponding posterior

density values $\gamma(y_{1:t}|x_t^{r_i})f(x_t^{r_i})$ with $i = 1, 2, 3$. From these quantities, the new parameter is given by

$$x_t^{\text{trigo}} = \sum_{i=1}^3 x_t^{r_i} / 3 + (p_2 - p_1)(x_t^{r_1} - x_t^{r_2}) + (p_3 - p_2)(x_t^{r_2} - x_t^{r_3}) + (p_1 - p_3)(x_t^{r_3} - x_t^{r_1}),$$

in which $p_i \propto \gamma(y_{1:t}|x_t^{r_i})f(x_t^{r_i})$ for $i \in [1, 3]$ are probabilities such that $\sum_{i=1}^3 p_i = 1$. Similarly, we can extend the DREAM, the walk and the stretch moves using the trigonometric parameter as follows

$$\text{DREAM : } \tilde{x}_t = x_t^j + Z_{\text{Dir}}F(\delta = 1, d)(x_t^{\text{trigo}} - x_t^j) + \zeta, \quad (25)$$

$$\text{Walk move : } \tilde{x}_t = x_t^j + Z_{\text{RW}}(x_t^j - x_t^{\text{trigo}}), \quad (26)$$

$$\text{Stretch move : } \tilde{x}_t = x_t^{\text{trigo}} + Z_{\text{Stretch}}(x_t^j - x_t^{\text{trigo}}), \quad (27)$$

where $Z_{\text{Dir}} = 1$ with probability 0.5 and -1 otherwise. Note that due to the random variable Z_{Dir} , the DREAM proposal (25) is still symmetric and therefore the acceptance ratio remains identical to the standard RW one.

The last two extensions are adaptations only for the stretch and the walk moves (as for the DREAM one, it does not change the initial proposal distribution). The next proposal comes from another heuristic optimization technique. The firefly (FF) algorithm, initially introduced in Yang (2009), updates the parameters by combining the attractiveness and the distance of the particles. For our purpose, we define the FF update as

$$x_t^{\text{FF}} = x_t^{r_1} + F_{FF}(x_t^{r_1} - x_t^{r_2}),$$

where F_{FF} is a chosen constant and r_1, r_2 are taken without replacement in the $M - 1$ remaining particles. The two new moves based on the FF equation are given by

$$\begin{aligned} \text{Walk move : } \tilde{x}_t &= x_t^j + Z_{\text{RW}}(x_t^j - x_t^{\text{FF}}), \\ &= x_t^j + Z_{\text{RW}}(x_t^j - x_t^{r_1}) + Z_{\text{RW}}F_{FF}(x_t^{r_1} - x_t^{r_2}), \end{aligned} \quad (28)$$

$$\begin{aligned} \text{Stretch move : } \tilde{x}_t &= x_t^{\text{FF}} + Z_{\text{Stretch}}(x_t^j - x_t^{\text{FF}}), \\ &= x_t^{r_1} + Z_{\text{Stretch}}(x_t^j - x_t^{r_1}) + (1 + Z_{\text{Stretch}})F_{FF}(x_t^{r_1} - x_t^{r_2}). \end{aligned} \quad (29)$$

We set F_{FF} of the walk move to $\frac{2.38}{E(Z_{\text{RW}})\sqrt{2d}}$ and the constant F_{FF} of the stretch move is fixed to $\frac{E(Z_{\text{Stretch}})}{E(Z_{\text{Stretch}})+1}$.

Regarding the last new updates, one can notice that the standard Differential Evolution mutation can also be used to improve the proposal distribution of the stretch and the walk moves. In particular, we consider the move of the DE optimization given by

$$x_t^{\text{DE}} = x_t^{r_1} + F_{DE}(x_t^{r_2} - x_t^{r_3}),$$

in which F_{DE} is a fixed constant and r_1, r_2, r_3 are taken without replacement in the $M - 1$ remaining particles. Inserting this update into the stretch and the walk moves delivers new proposal distributions as follows

$$\begin{aligned} \text{Walk move : } \tilde{x}_t &= x_t^j + Z_{\text{RW}}(x_t^j - x_t^{\text{DE}}), \\ &= x_t^j + Z_{\text{RW}}(x_t^j - x_t^{r_1}) + Z_{\text{RW}}F_{DE}(x_t^{r_2} - x_t^{r_3}), \end{aligned} \tag{30}$$

$$\begin{aligned} \text{Stretch move : } \tilde{x}_t &= x_t^{\text{DE}} + Z_{\text{Stretch}}(x_t^j - x_t^{\text{DE}}), \\ &= x_t^{r_1} + Z_{\text{Stretch}}(x_t^j - x_t^{r_1}) + (1 + Z_{\text{Stretch}})F_{DE}[(x_t^{r_2} - x_t^{r_3})]. \end{aligned} \tag{31}$$

Similarly to the Firefly proposal, we fix F_{DE} of the walk move to $\frac{2.38}{E(Z_{\text{RW}})\sqrt{2d}}$ and the constant F_{DE} of the stretch move is set to $\frac{E(Z_{\text{Stretch}})}{E(Z_{\text{Stretch}})+1}$.

The standard DREAM, the walk and the stretch moves are typically used in an MCMC context. However, when the parameter dimension d is large, many parallel chains must be run because, as all these updates are based on linear transformations, they can only generate subspaces spanned by their current positions. To remedy this issue in the MCMC scheme, Vrugt, ter Braak, Diks, Robinson, Hyman, and Higdon (2009) have introduced the CR probability. Once the proposal parameter has been generated, each element is randomly kept or set back to the previous value according to some fixed probability p_{CR} . Eventually, the standard MH acceptance step takes place. In contrast, these multiple chains arise naturally in SMC frameworks since the rejuvenate step consists in updating all the particles by some MCMC iterations. However, the CR probability has the additional advantage of generating many other moves of the parameters. For this reason, we also include the CR step into our MCMC kernel.

In order to test all the new move strategies, Table 1 documents the average autocorrelation times over the multivariate random realizations (computed by batch means, see Geyer (1992)), obtained from each update rule. The dimension of each distribution from which the realizations are sampled is set to 5 and we consider Normal distributions with low and high

correlations as well as a student distribution with a degree of freedom equal to 5. From this short analysis, the DREAM update is the most efficient in terms of mixing. We also observe that the additional moves perform better than the standard ones for the walk and the stretch moves.

Move	Stretch move	Walk move	DREAM
5-dimension Normal distribution with correlation of 0.5 and variances set to unity			
Standard	84.99	106.19	13.79
Trigo	56.62	58.59	20.36
FF	52.21	51.75	–
DE	44.85	66.81	–
5-dimension Normal distribution with correlation of 0.999 and variances set to unity			
Standard	92.94	96.63	34.93
Trigo	63.14	82.83	23.91
FF	59.31	38.54	–
DE	70.07	61.45	–
5-dimension Student distribution with correlation of 0.999, $df = 5$ and variances set to unity			
Standard	104.59	75.91	23.11
Trigo	54.66	65.38	19.83
FF	38.17	35.21	–
DE	37.01	35.53	–

Table 1: Average of the Autocorrelation times over the 5 dimensions for multiple update moves and different distributions.

As the performance of a specific Metropolis update generally depends on the distribution that has to be drawn, we suggest to incorporate all the generic moves in combination with

a fixed CR probability into the MCMC rejuvenation step of the SMC algorithm. In practice, at each MCMC iteration, the proposal distribution is chosen among the different update equations ((20),(21),(22),(25),(26),(27),(28),(29),(30),(31)) according to a multinomial probability p_{kernel} . Then, some of the new elements of the updated vector are set back to their current MCMC value according to the CR probability. The proposal is then accepted with probability that is defined either by the standard RW metropolis ratio or by (23) or (24) depending on the selected mutation rule.⁸ By assessing the efficiency of each update equation with the Mahalanobis distance, one can monitor which proposal leads to the best exploration of the support and can appropriately and automatically adjust the probability p_{kernel} at the end of the rejuvenation step. More precisely, once a proposed parameter is accepted, we add the Mahalanobis distance between the previous and the accepted parameters to the distance already achieved by the selected move. At the end of each rejuvenation step, the probabilities p_{kernel} are reset proportionally to the distance performances of all the moves.

Two relevant issues should be discussed. First, the MCMC kernel makes interacting the particles, which rules out the desirable parallel property of the SMC. To keep this advantage, we apply the kernel on subsets of particles instead of on all the particles and we perform paralelization between the subsets. Secondly and more importantly, the SMC theory derived in Section 2 does not allow for particle interaction. Proposition 1 ensures that the TNT sampler also works under a DREAM-type MCMC kernel.

Proposition 1. Consider a SMC sampler with a given number of particles M and MCMC kernels allowing for interacting particles via the proposal distribution (20) or (25). Then, it yields a standard SMC sampler with particle weights given by equation (4).

Proof. See Appendix A.

Adapting the proof for the walk and the stretch moves is straightforward as the stationary distribution of the Markov-chain also factorizes into a product of the targeted distribution.

Adaption of the scale parameters $F(\delta, d)$, a_W and a_S

⁸Although only the chosen mixture enters in the MH acceptance ratio, the MCMC algorithm is still valid. For further explanations, see Geweke (2005), section Transition Mixtures.

Since the chosen backward MCMC kernel in the algorithm derivation implies that the consecutive distributions approximated by the TNT sampler are very similar, we can analyze the mixing properties of the previous MCMC kernel to adapt the scale parameters $\gamma(\delta, d)$, a_W and a_S . Atchadé and Rosenthal (2005) present a simple recursive algorithm in order to achieve a specified acceptance rate in an MCMC framework. Considering one scale parameter (either $F(\delta, d)$, a_W or a_S) generically denoted by c_{n-1} , at the end of the $n - 1$ SMC step, we adapt the parameter as follows :

$$c_{n-1} = p(c_{n-2} + \frac{\alpha_{n-1} - \alpha_{\text{targeted}}}{(n-1)^{0.6}}) \quad (32)$$

where the function $p(\cdot)$ is such that $p(c) = c$ if $c \in [A_0, +\infty]$ and $p(c) = A_0$ if $c < A_0$, the parameter α_{n-1} stands for the acceptance rate of the MCMC kernel of the $n - 1$ SMC step and α_{targeted} is a user-defined acceptance rate. The function $p(\cdot)$ prevents from negative values of the recursive equation and if the optimal scale parameter lies in the compact set $[A_0, +\infty]$, the equation will converge to it (in an MCMC context). In the empirical exercise, we fix the variable A_0 to 1e-8 for the DREAM-type move and to 1.01 for the other updates. The rate α_{targeted} is set to $\frac{1}{3}$ implying that every three MCMC iterations, all the particles have been approximately rejuvenated. It is worth emphasizing that the denominator $(n - 1)^{0.6}$ has been chosen as proposed in Atchadé and Rosenthal (2005) but its value, which ensures the ergodicity property in an MCMC context, is not relevant in our SMC framework since at each rejuvenation step, the scale parameter c_n is fixed for the entire MCMC move. The validity of this adaption can be theoretically justified by Beskos, Jasra, and Thiery (2013).

When the parameter space evolves over time, the MCMC kernel can become model dependent since sampling the state vector using a filtering method is often the most efficient technique in terms of mixing. In special cases where the forward-backward algorithm (Rabiner (1989)) or the Kalman filter (Kalman (1960)) operate, the state can be filtered out. By doing so, we come back to the framework with static parameter space. For non linear state space model, recent works of Chopin, Jacob, and Papaspiliopoulos (2013) and Fulop and Li (2013) rely on the particle MCMC framework of Andrieu, Doucet, and Holenstein (2010) for integrating out the state vector. We believe that switching from the tempered domain to the time one as well as employing the Evolutionary MCMC kernel presented above could even more increase the efficiency of these sophisticated SMC samplers. For example, the particle discrepancies

of the early stage inherent to the IBIS algorithm is present in all the empirical simulations of Fulop and Li (2013) whereas with the TNT sampler, we can ensure a minimum ESS value during the entire procedure.

4 Simulations

We first illustrate the TNT algorithm through a simulation exercise before presenting results on empirical data. As the TNT algorithm is now completely defined, we start by spelling out the values set for the different parameters to be tuned. The threshold κ is recommended to be high as the evolutionary MCMC updates crucially depend on the diversification of the particles to operate. For that reason we set it to $0.75M$. The second threshold κ_1 that triggers a new run of the simulated annealing algorithm is chosen as $0.1M$ and the number of particles is set to $M = 2000$. We fix the acceptance rate of the MCMC move to $1/3$ and the number of MCMC step is set to $J = 90$. The number of MCMC iterations amounts to 90 in order to insure that each particle has moved away from its current position as it approximately implies 30 accepted draws. For all the simulations of the paper, Table 2 summaries these choices.

Parameters	TNT algorithm
Nb. Particles M	2.000
Threshold ESS κ	$0.75M$
Sec. threshold ESS κ_1	$0.1M$
Acc. rate α_{targeted}	$1/3$
Nb. MCMC J	90

Table 2: Tuned parameters for the TNT algorithm.

Our benchmark model for testing the algorithm is a Change-point Generalized Autoregressive Conditional Heteroskedastick (GARCH) process that is defined as follows

$$y_t = \mu_i + \epsilon_t \quad \text{with} \quad \epsilon_t | y_{1:t-1} \sim N(0, \sigma_t^2), \quad (33)$$

$$\sigma_t^2 = \omega_i + \alpha_i \epsilon_{t-1}^2 + \beta_i \sigma_{t-1}^2 \quad \text{for} \quad t \in [\tau_{i-1} + 1, \tau_i] \quad \text{and} \quad t > 1, \quad (34)$$

where $\tau_1 = 0$, $\tau_K = T$ and τ_i with $i \in [2, K]$ denotes the observation when break i occurs. The number K of break points are fixed before the estimation and occur sequentially (i.e.

$\tau_{i-1} < \tau_i < \tau_{i+1} \forall i \in [2, K-1]$). Stationarity conditions are imposed within each regime by assuming $|\alpha_i + \beta_i| < 1$. The Table 3 documents the prior distributions of the model parameters. We innovate by assuming that the regime durations $d_1 = \tau_1$ and $d_i = \tau_i - \tau_{i-1} \forall i \in [2, K-1]$ are continuous and are driven by exponential distributions. The duration parameters are therefore identifiable up to a discrete value since they indicate at which observation the process switches from one set of parameter to another. However it brings an obvious advantage as it makes possible to use the Metropolis update developed in Section 3 for the duration parameters too. Consequently, we are able to update in one block all the model parameters. The TNT algorithm for CP-GARCH models is available on the author's website.

Mean parameter $\forall i \in [1, K]$:		
$\mu_i \sim N(0,1)$		
GARCH parameters $\forall i \in [1, K]$:		
$\omega_i \sim U[0, 1]$	$\alpha_i \beta_i \sim U[0, 1 - \beta_i]$	$\beta_i \sim U[0.2, 1]$
Break parameters $\forall i \in [2, K-1]$:		
$d_1 = \tau_1 \sim \text{Exp}(\lambda)$	$d_i = \tau_i - \tau_{i-1} \sim \text{Exp}(\lambda)$	$\lambda \sim \text{Gamma}(1, 1/T)$

Table 3: Prior Distributions of the CP parameters. The distribution $N(a, b)$ denotes the Normal distribution with expectation a and variance b and $U[a, b]$ stands for the Uniform distribution with lower bound a and upper bound b . The exponential distribution with parameter λ is expressed as $\text{Exp}(\lambda)$ (with density function : $f(\tau|\lambda) = \lambda^\tau e^{-\lambda\tau}$) and the Gamma distribution is denoted by $\text{Gamma}(a, b)$ in which a is the shape parameter and b the scale one (with density function $f(\lambda|a, b) = \frac{b^a}{\Gamma(a)} \lambda^{a-1} e^{-b\lambda}$).

We generate 4000 observations from the data generating process (DGP) of Table 4. The DGP exhibits four breaks in the volatility dynamic and tries to mimic the turbulent and quiet periods observed in a financial index. Figure 1 shows a simulated series and the corresponding volatility over time.

We use the marginal log-likelihood (MLL) for selecting the number of regimes by estimating several CP-GARCH models differing by their number of regimes (see Chib (1998)). As the TNT algorithm is both an off-line and an on-line method, we start by estimating the pos-

	μ	α	β	σ^2	τ
Regime 1	0	0.1	0.1	0.85	1250
Regime 2	0	0.3	0.03	0.95	2230
Regime 3	0	0.25	0.2	0.70	3170
Regime 4	0	0.4	0.05	0.9	—

Table 4: Data generating process of the CP-GARCH model.

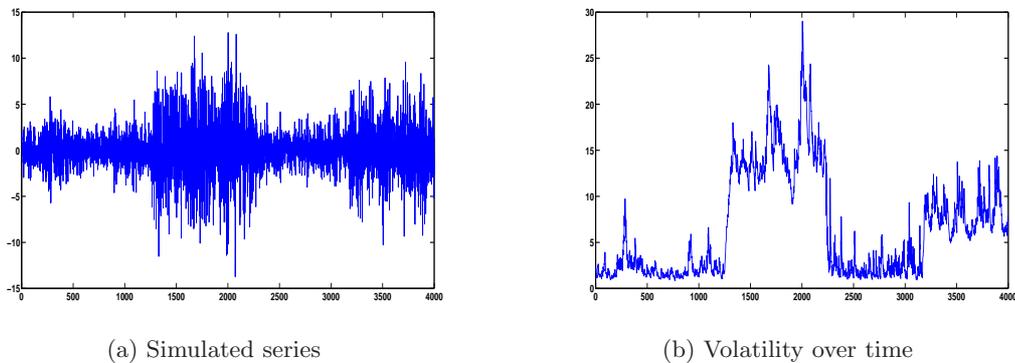


Figure 1: Simulated series from the DGP 4 and its corresponding volatility over time.

terior distribution with 3000 observations (i.e. $\tau = 3000$) and then we add one by one the remaining observations. For each model, we obtain 1001 estimated posterior distributions (from $\pi(x_\tau|y_{1:\tau})$ to $\pi(x_T|y_{1:T})$) and their respective 1001 MLLs. By so doing, the evolution of the best model over time can be observed. A sharp decrease in the MLL value means that the model cannot easily capture the new observation. According to the DGP 4, the model exhibiting 3 regimes should at least dominate over the first 170 observations and then the model with 4 regimes should gradually take the lead.

Figure 3 shows the log-Bayes factors (log-BFs) of CP-GARCH models with respect to the standard GARCH process (i.e. $K = 1$).⁹ The best model over give or take the first 300 observations is the one exhibiting three regimes. Afterward, it is gradually dominated by the

⁹We remind that the log-BF is computed as the difference of the MLL of two models. Following the informal rule of Kass and Raftery (1995), if the logarithm of the Bayes factor exceeds 3, we have strong evidence in favor of the model with the highest value.

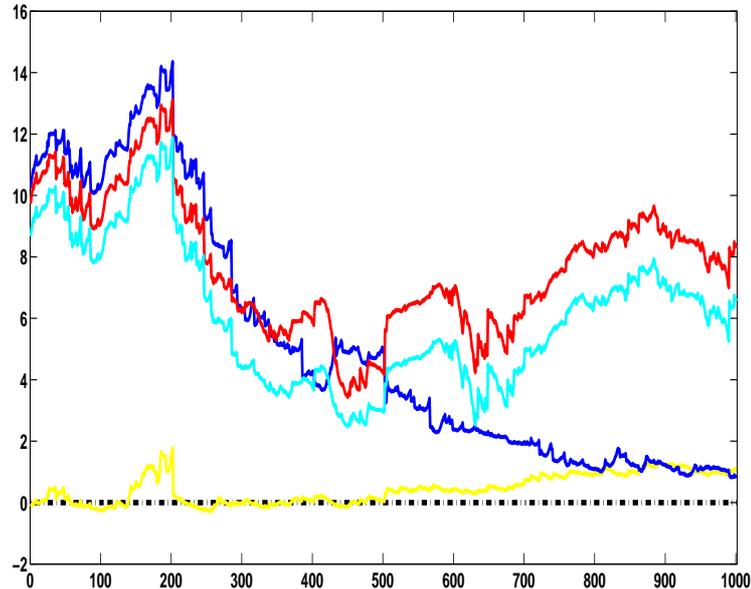


Figure 2: Log-BF over time of the CP-GARCH models in relation to the GARCH one. The log-BF of the CP-GARCH model with two, three, four and five regimes are depicted in yellow, blue, red and cyan respectively. A positive value provides evidence in favor of the considered model compared to the GARCH one.

process with four regimes (in red). The on-line algorithm has been able to detect the coming break and according to the MLLs, around 150 observations are needed to identify it.

Table 5 documents the posterior means of the parameters of the model exhibiting the highest MLL at the end of the simulation (i.e. with a number of regimes equal to 4) as well as their standard deviations. We observe that the values are close to the true ones which indicates an accurate estimation of the model. The breaks are also precisely inferred. At least for this particular DGP, the TNT algorithm is able to draw the posterior distribution of CP-GARCH models and correctly updates the distribution in the light of new observations.

Eventually, one can have a look to the varying probabilities associated with each Evolutionary update function. These probabilities are computed at each SMC iteration and are proportional to the Mahalanobis distances of the accepted draws. Table 6 documents the values for several SMC iterations. We observe that the probabilities highly vary over the SMC iterations.

	μ	ω	α	β	τ
Regime 1	-0.02	0.11	0.11	0.85	1253.7
	(0.04)	(0.04)	(0.03)	(0.04)	(9.91)
Regime 2	0.15	0.67	0.05	0.91	2238.4
	(0.13)	(0.21)	(0.02)	(0.03)	(17.42)
Regime 3	0.01	0.25	0.16	0.73	3169.4
	(0.04)	(0.08)	(0.03)	(0.05)	(11.33)
Regime 4	0.06	0.61	0.07	0.85	
	(0.1)	(0.22)	(0.02)	(0.04)	

Table 5: Posterior means of the parameters of the CP-GARCH model with four regimes and their corresponding standard deviations.

Moreover, the stretch move and the DREAM algorithm slightly dominate the walk update.

SMC iteration	Stretch Move				Walk Move				DREAM Move	
	Trigo	DE	Firefly	Standard	Trigo	DE	Firefly	Standard	Trigo	Standard
1th	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
10th	0.13	0.09	0.08	0.07	0.10	0.06	0.07	0.08	0.22	0.11
100th	0.14	0.11	0.13	0.13	0.11	0.04	0.06	0.16	0.09	0.04
last	0.11	0.12	0.11	0.11	0.08	0.05	0.06	0.10	0.17	0.09
Mean	0.13	0.12	0.12	0.11	0.10	0.05	0.06	0.14	0.12	0.06

Table 6: Probabilities (proportional to the Mahalabonis distance) of choosing a specific type of Metropolis-Hastings move. Mean stands for the average over the SMC iterations.

5 Empirical application

As emphasized in the simulated exercise, the TNT algorithm allows for comparing complex models through marginal likelihoods. We examine the performances of the CP-GARCH models over time on the S&P 500 daily percentage returns spanning from February 08, 1999 to June 24, 2015 (4000 observations). We estimate the models with a number of regimes varying from 1 to 5 using the TNT algorithm and we fix the value $\tau = 3000$ which controls the change from the tempered to the time domain.

To begin with, Table 7 documents the MLLs of the CP-GARCH models with different number of regimes when all the observations have been included. The best model exhibits four regimes.

# Regimes	1	2	3	4	5
MLL	-5732.6	-5730.86	-5731.1	-5727.87	-5729.1

Table 7: S&P 500 daily log-returns - MLLs of the CP-GARCH models given different number of regimes. The highest value is bolded.

Table 8 provides the posterior means and the standard deviations of the best CP-GARCH model. Not surprisingly, the break dates occur after the dot-com bubble and at the beginning of the financial crisis. To link the results with the crisis event, Freddie Mac company announced that it will no longer buy the most risky subprime mortgages and mortgage-related securities in February 17th, 2007. This date sometimes refers to the beginning of the collapse of the financial system.

We now turn to the recursive estimations of the CP-GARCH models. For the 1001 estimated posterior distributions (from $\pi(x_\tau|y_{1:\tau})$ to $\pi(x_T|y_{1:T})$), Figure 3 shows the log-BFs of CP-GARCH models with respect to the fixed parameter GARCH model (i.e. $K = 1$). The CP-GARCH model with four regimes dominates over the entire period. The process with 5 regimes fits similarly the data but is over-parametrized compared to the same model with 4 regimes. The difference between the two models comes from the penalization of this over-parametrization through the prior distributions.

To end this study, Table 9 delivers the probabilities associated with each Evolutionary update function for several SMC iterations. As in the simulated exercise, the stretch move and the

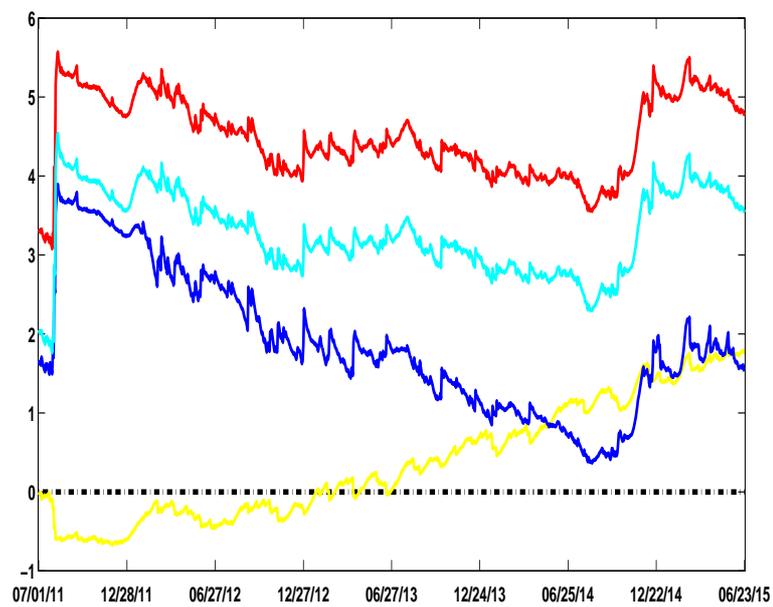


Figure 3: S&P 500 daily log-returns - log-BFs over time of the volatility models in relation to the GARCH one. The log-BFs of the CP-GARCH models with two, three, four and five regimes are depicted in yellow, blue, red and cyan respectively. A positive value provides evidence in favor of the considered model compared to the GARCH one.

	μ	ω	α	β	τ
Regime 1	-0.01 (0.05)	0.16 (0.06)	0.11 (0.03)	0.81 (0.05)	25 March 2003 (74.22)
Regime 2	0.06 (0.02)	0.02 (0.03)	0.05 (0.02)	0.91 (0.05)	14 February 2007 (62.33)
Regime 3	-0.33 (0.34)	0.66 (0.22)	0.13 (0.15)	0.52 (0.20)	09 March 2007 (32.63)
Regime 4	0.08 (0.02)	0.02 (0.01)	0.12 (0.01)	0.86 (0.01)	

Table 8: S&P 500 daily log-returns : Posterior means of the parameters of the CP-GARCH model with four regimes and their corresponding standard deviations.

DREAM algorithm slightly dominate the walk one. We also observe that the trigonometric move exhibits good mixing properties since its associated probabilities are high, especially for the DREAM-type update.

SMC iteration	Stretch Move				Walk Move				DREAM Move	
	Trigo	DE	Firefly	Standard	Trigo	DE	Firefly	Standard	Trigo	Standard
1th	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
10th	0.12	0.09	0.08	0.06	0.10	0.06	0.07	0.07	0.24	0.12
100th	0.13	0.13	0.12	0.12	0.08	0.05	0.05	0.11	0.15	0.07
last	0.12	0.12	0.12	0.11	0.08	0.05	0.06	0.10	0.17	0.09
Mean	0.12	0.12	0.11	0.11	0.09	0.05	0.05	0.10	0.16	0.08

Table 9: S&P 500 daily log-returns - Probabilities (proportional to the Mahalabonis distance) of choosing a specific type of Metropolis-Hastings move. Mean stands for the average over all the SMC iterations. The highest probability is bolded.

6 Conclusion

We develop an off-line and on-line SMC algorithm (called TNT) well-suited for situations where a relevant number of similar distributions has to be estimated. The method encompasses the off-line AIS of Neal (1998), the on-line IBIS algorithm of Chopin (2002) and the RM method of Gilks and Berzuini (2001) that all arise as special cases in the SMC sampler theory (see Del Moral, Doucet, and Jasra (2006)). The TNT algorithm benefits from the conjugacy of the tempered and the time domains to avoid particle degeneracies observed in the on-line methods. More importantly, we introduce a new adaptive MCMC kernel based on the Evolutionary optimization literature which consists in 10 different moves based on particles interactions. These MCMC updates are selected according to probabilities that are adjusted over the SMC iterations. Furthermore, the scale parameter of these updates are also automated thanks to the method of Atchadé and Rosenthal (2005). It makes the TNT algorithm fully generic and one needs only to plug the likelihood function, the prior distributions and the number of particles to use it.

The TNT sampler combines on-line and off-line estimations and is consequently suited for comparing complex models. Through a simulated exercise, the paper highlights that the algorithm is able to detect structural breaks of a CP-GARCH model on the fly. Eventually, an empirical application on the S&P 500 daily percentage log-returns shows that no break in the volatility of the GARCH model had arisen from January 7th, 2011 to June 24th, 2015. In fact, the MLL clearly indicates evidence in favor of a CP-GARCH model exhibiting 4 regimes in which the breaks occur at the end of the dot-com bubble as well as at the beginning of the financial crisis.

We believe that the TNT algorithm could be adapted to recent SMC algorithms such as Fearnhead and Taylor (2013) and Fulop and Li (2013) since they propose advanced SMC samplers based on the IBIS and the E-AIS samplers. Another avenue of research could be an application on Change-point stochastic volatility models. Indeed, the Evolutionary MCMC kernel is potentially able to update the volatility parameters of these models without filtering them.

7 Acknowledgement

The author would like to thank Rafael Wouters for his advices on an earlier version of the paper and is grateful to Nicolas Chopin who provided his comments that helped to improve the quality of the paper. Research has been supported by the National Bank of Belgium and by the contract "Investissement d'Avenir" ANR-11-IDEX-0003/Labex Ecodec/ANR-11-LABX-0047 granted by the Centre de Recherche en Economie et Statistique (CREST). Arnaud Dufays is also a CREST associate Research Fellow. The views expressed in this paper are the author's ones and do not necessarily reflect those of the National Bank of Belgium. The scientific responsibility is assumed by the author.

A Proof of Proposition 1

Using the notation $x_{1:n}^{1:M} = \{x_1^1, \dots, x_1^M, x_2^1, \dots, x_n^M\}$ which stands for NxM random variables and assuming that $x_i^j \in E \ \forall i, j$ as in the E-AIS method (tempered domain) or the IBIS one (time domain), we consider the augmented posterior distribution :

$$\tilde{\pi}_n(x_{1:n}^{1:M}) = \left[\prod_{i=1}^M \pi_n(x_n^i) \right] \prod_{k=2}^n L_k(x_{k-1}^1 | x_k^{1:M}) \prod_{q=2}^M L_k(x_{k-1}^q | x_{k-1}^{1:q-1}, x_k^{q:M})$$

If the backward kernels $L_k(\cdot|\cdot)$ denote proper distributions, the product of the distribution of interest marginally arises :

$$\begin{aligned} \tilde{\pi}_n(x_n^{1:M}) &= \int \left[\prod_{i=1}^M \pi_n(x_n^i) \right] \prod_{k=2}^n L_k(x_{k-1}^1 | x_k^{1:M}) \prod_{q=2}^M L_k(x_{k-1}^q | x_{k-1}^{1:q-1}, x_k^{q:M}) dx_{1:n-1}^{1:M} \\ &= \left[\prod_{i=1}^M \pi_n(x_n^i) \right]. \end{aligned}$$

The SMC sampler with DREAM MCMC kernels leads to a proposal distribution of the form :

$$\eta_n(x_n^{1:M}) = \left[\prod_{i=1}^M f(x_n^i) \right] \prod_{k=2}^n K_k(x_k^1 | x_{k-1}^{1:M}) \prod_{q=2}^M K_k(x_k^q | x_{k-1}^{1:q-1}, x_{k-1}^{q:M})$$

where $K_k(\cdot|\cdot)$ denotes the DREAM subkernel with invariant distribution $\pi_k(\cdot)$. Sampling one draw from this proposal distribution is achieved by firstly drawing M realizations from the

prior distribution and then applying the DREAM algorithm (N-1)xM times. As proven in Vrugt, ter Braak, Diks, Robinson, Hyman, and Higdon (2009) and Bauwens, Dufays, and De Backer (2011), the DREAM algorithm leads to the detailed balance equation :

$$\begin{aligned} & \left[\prod_{i=1}^M \pi_k(x_{k-1}^i) \right] K_k(x_k^1 | x_{k-1}^{1:M}) \prod_{q=2}^M K_k(x_k^q | x_{k-1}^{1:q-1}, x_{k-1}^{q:M}) \\ = & \left[\prod_{i=1}^M \pi_k(x_k^i) \right] K_k(x_{k-1}^1 | x_k^{1:M}) \prod_{q=2}^M K_k(x_{k-1}^q | x_{k-1}^{1:q-1}, x_k^{q:M}) \end{aligned}$$

Using this relation, we specify the backward kernel as

$$\begin{aligned} & L_k(x_{k-1}^1 | x_k^{1:M}) \prod_{q=2}^M L_k(x_{k-1}^q | x_{k-1}^{1:q-1}, x_k^{q:M}) \\ = & \frac{\left[\prod_{i=1}^M \pi_k(x_{k-1}^i) \right] K_k(x_k^1 | x_{k-1}^{1:M}) \prod_{q=2}^M K_k(x_k^q | x_{k-1}^{1:q-1}, x_k^{q:M})}{\left[\prod_{i=1}^M \pi_k(x_k^i) \right]} \end{aligned}$$

The sequential importance sampling procedure generates weights given by

$$\begin{aligned} \tilde{w}_n(x_{1:n}^{1:M}) & \equiv \frac{\tilde{\pi}_n(x_{1:n}^{1:M})}{\eta_n(x_{1:n}^{1:M})} = \tilde{w}_{n-1}(x_{1:n-1}^{1:M}) \frac{\left[\prod_{i=1}^M \pi_n(x_{n-1}^i) \right]}{\left[\prod_{i=1}^M \pi_{n-1}(x_{n-1}^i) \right]} \\ & = \prod_{i=1}^M w_n(x_{1:n}^i), \end{aligned}$$

resulting in a product of independent weights exactly equal to the product of SMC sampler weights (see equation (4)).

References

- ANDRIEU, C., A. DOUCET, AND R. HOLENSTEIN (2010): “Particle Markov Chain Monte Carlo Methods,” *Journal of the Royal Statistical Society B*, 72, 269–342.
- ATCHADÉ, Y., AND J. ROSENTHAL (2005): “On adaptive Markov chain Monte Carlo algorithms,” *Bernoulli*, 11(5), 815–828.
- BAUWENS, L., A. DUFAYS, AND B. DE BACKER (2011): “Estimating and forecasting structural breaks in financial time series,” *Journal of Empirical Finance*, Forthcoming, DOI: 10.1016/j.jempfin.2014.06.008.

- BAUWENS, L., A. DUFAYS, AND J. ROMBOUTS (2013): “Marginal Likelihood for Markov Switching and Change-point GARCH Models,” *Journal of Econometrics*, 178 (3), 508–522.
- BESKOS, A., A. JASRA, AND A. THIERY (2013): “On the Convergence of Adaptive Sequential Monte Carlo Methods,” Available at <http://arxiv.org/pdf/1306.6462v2.pdf>.
- CHIB, S. (1998): “Estimation and comparison of multiple change-point models,” *Journal of Econometrics*, 86, 221–241.
- CHIB, S., F. NARDARI, AND N. SHEPHARD (2002): “Markov chain Monte Carlo methods for stochastic volatility models,” *Journal of Econometrics*, 108(2), 281–316.
- CHOPIN, N. (2002): “A Sequential Particle Filter Method for Static Models,” *Biometrika*, 89, 539–551.
- CHOPIN, N., P. E. JACOB, AND O. PAPASPILIOPOULOS (2013): “SMC2: an efficient algorithm for sequential analysis of state space models,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75, 397–426.
- CHRISTEN, J. A., AND C. FOX (2010): “A general purpose sampling algorithm for continuous distributions (the t-walk),” *Bayesian Anal.*, 5(2), 263–281.
- DAS, S., AND P. SUGANTHAN (2011): “Differential Evolution: A Survey of the State-of-the-Art,” *Evolutionary Computation, IEEE Transactions on*, 15(1), 4–31.
- DEL MORAL, P., A. DOUCET, AND A. JASRA (2006): “Sequential Monte Carlo samplers,” *The Royal Statistical Society: Series B(Statistical Methodology)*, 68, 411–436.
- (2012): “On adaptive resampling strategies for sequential Monte Carlo methods,” *Bernoulli*, 18(1), 252–278.
- DURHAM, G., AND J. GEWEKE (2012): “Adaptive Sequential Posterior Simulators for Massively Parallel Computing Environments,” <http://www.censoc.uts.edu.au/pdfs/gewekepapers/gpu2full.pdf>.
- DUTTA, S., AND S. BHATTACHARYA (2014): “Markov chain Monte Carlo based on deterministic transformations,” *Statistical Methodology*, 16(Complete), 100–116.

- FAN, H.-Y., AND J. LAMPINEN (2003): “A Trigonometric Mutation Operation to Differential Evolution,” *J. of Global Optimization*, 27(1), 105–129.
- FEARNHEAD, P., AND B. TAYLOR (2013): “An adaptive Sequential Monte Carlo Sampler,” *Bayesian Analysis*, 8 (2), 411–438.
- FOREMAN-MACKEY, D., D. W. HOGG, D. LANG, AND J. GOODMAN (2013): “emcee: The MCMC Hammer,” *PASP*, 125, 306–312.
- FULOP, A., AND J. LI (2013): “Efficient learning via simulation: A marginalized resample-move approach,” *Journal of Econometrics*, 176(2), 146 – 161.
- GEWEKE, J. (1989): “Bayesian Inference in Econometric Models Using Monte Carlo Integration,” *Econometrica*, 57, 1317–1339.
- (2005): *Contemporary Bayesian Econometrics and Statistics*, Wiley Series in Probability and Statistics. Wiley.
- GEYER, C. J. (1992): “Practical Markov Chain Monte Carlo,” *Statistical Science*, 7(4), 473–511.
- GILKS, W. R., AND C. BERZUINI (2001): “Following a moving target - Monte Carlo inference for dynamic Bayesian models,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63, 127–146.
- GORDON, N., D. SALMOND, AND A. F. M. SMITH (1993): “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *Radar and Signal Processing, IEE Proceedings F*, 140(2), 107–113.
- HERBST, E., AND F. SCHORFHEIDE (2012): “Sequential Monte carlo sampling for DSGE models,” *Working Paper No12-27, Federal reserve Bank of Philadelphia*.
- JASRA, A., D. A. STEPHENS, A. DOUCET, AND T. TSAGARIS (2011): “Inference for Lévy-Driven Stochastic Volatility Models via Adaptive Sequential Monte Carlo,” *Scandinavian Journal of Statistics*, 38, 1–22.
- JASRA, A., D. A. STEPHENS, AND C. C. HOLMES (2007): “On population-based simulation for static inference,” *Statistics and Computing*, 17(3), 263–279.

- JEREMIAH, E., S. SISSON, L. MARSHALL, R. MEHROTRA, AND A. SHARMA (2011): “Bayesian calibration and uncertainty analysis of hydrological models: A comparison of adaptive Metropolis and sequential Monte Carlo samplers,” *Water Resources Research*, 47(7), n/a–n/a.
- KALMAN, R. E. (1960): “A New Approach to Linear Filtering and Prediction Problems,” *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D), 35–45.
- KASS, R., AND A. RAFTERY (1995): “Bayes Factors,” *Journal of the American Statistical Association*, 90, 773–795.
- NEAL, R. M. (1998): “Annealed Importance Sampling,” *Statistics and Computing*, 11, 125–139.
- RABINER, L. R. (1989): “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, pp. 257–286.
- ROBERTS, G. O., AND J. S. ROSENTHAL (2001): “Optimal scaling for various Metropolis-Hastings algorithms,” *Statistical Science*, 16, 351–367.
- SMITH, A. F. M., AND A. E. GELFAND (1992): “Bayesian Statistics without Tears: A Sampling-Resampling Perspective,” *The American Statistician*, 46, 84–88.
- STORN, R., AND K. PRICE (1997): “Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces,” *Journal of Global Optimization*, 11(4), 341–359.
- VRUGT, J. A., C. J. F. TER BRAAK, C. G. H. DIKS, B. A. ROBINSON, J. M. HYMAN, AND D. HIGDON (2009): “Accelerating Markov Chain Monte Carlo Simulation by Differential Evolution with Self-Adaptative Randomized Subspace Sampling,” *International Journal of Nonlinear Sciences and Numerical Simulations*, 10, 271–288.
- YANG, X.-S. (2009): “Firefly Algorithms for Multimodal Optimization,” in *Stochastic Algorithms: Foundations and Applications*, ed. by O. Watanabe, and T. Zeugmann, vol. 5792 of *Lecture Notes in Computer Science*, pp. 169–178. Springer Berlin Heidelberg.