

Article

# Evolutionary Sequential Monte Carlo Samplers for Change-Point Models

Arnaud Dufays

Department of economics, Laval University, 2216 Pavillon J.-A.-DeSève, QC G1V 0A6, Canada; arnaud.dufays@ecn.ulaval.ca; Tel.: +1-418-6562131

Academic Editors: Roberto Casarin, Francesco Ravazzolo, Herman K. van Dijk and Nalan Basturk  
Received: 24 August 2015; Accepted: 28 January 2016; Published: 8 March 2016

**Abstract:** Sequential Monte Carlo (SMC) methods are widely used for non-linear filtering purposes. However, 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 do SMC algorithms draw posterior distributions of static or dynamic parameters but additionally they provide an estimate of the marginal likelihood. The tempered and time (TNT) algorithm, developed in this 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 readily available. The algorithm is notably appropriate for estimating change-point models. As an example, we compare several change-point GARCH models through their marginal log-likelihoods over time.

**Keywords:** bayesian inference; sequential monte carlo; annealed importance sampling; change-point models; differential evolution; GARCH models

**JEL:** C11, C15, C22, C58.

---

## 1. Introduction

Sequential Monte Carlo (SMC) algorithm is a simulation-based procedure used in Bayesian frameworks 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 interest <sup>1</sup>. 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 an 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 ([1–3]) and a direct computation of the marginal likelihood (*i.e.*, the normalizing constant of the targeted distribution, see, e.g., [4]).

Recently, the SMC algorithms have been applied to infer static parameters, field in which the MCMC algorithm excels. Neal [5] provides a relevant improvement in this direction by building a SMC algorithm, named annealed importance sampling (AIS), that sequentially evolves from the

---

<sup>1</sup> whereas the sequence does not have to necessarily evolve over the time domain

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 an MCMC kernel at each SMC iteration. Few years later, [6] proposes an Iterated Batch Importance Sampling (IBIS) SMC algorithm, a special case of the Re-sample Move (RM) algorithm of [7], 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 [8]) unifies, among others, these SMC algorithms in a theoretical framework. It is shown that the methods of [5,7] 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 [9–11]) where it is demonstrated that the SMC mixing properties often dominate the MCMC approach based on a single Markov-chain. Nowadays papers are devoted to build self-adapting SMC samplers by automatically tuning the MCMC kernel (e.g., [12]), by marginalizing the state vector (in a state space specification) using the particle MCMC framework (e.g., [13,14]), to construct efficient SMC samplers for parallel computations (see [15]) or to simulate from complex multi-modal posterior distributions (e.g., [16]).

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 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., [17]). A recursive forecast exercise on many observations is therefore out of reach. Our first contribution is a new SMC algorithm, called 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 [5] with the IBIS algorithm of [6] if the model parameters are static or with the RM method of [7] 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 ([6,7]). 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 [6] (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 to limit 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 the 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 is no exception. We contribute by proposing several 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 the particles interactions. We start by emphasizing that the Differential Evolution Adaptive Metropolis (DREAM, see [18]), the walk move (see [19]) and the stretch one (see [20]) 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 several 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 us 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  $f(\cdot)$  and  $g(\cdot)$  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  $\theta$  with distribution  $f(y_t|\theta, s_t)$ . The innovations  $\omega_t$  and  $v_t$  are mutually independent and stand for the noise of the observation/state equations. The model parameters included in  $\theta$  do not evolve over time (*i.e.*, they are static). Let us 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 sampling<sup>2</sup> (AIS, see [5]) with the Re-sample Move (RM) SMC inference of [7]<sup>3</sup>. To build the TNT algorithm, we rely on the theoretical paper of [8] 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

<sup>2</sup> enhanced in the sense that the AIS incorporates a re-sampling step

<sup>3</sup> The IBIS algorithm being a particular case

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 the 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  $\pi_p$  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 random variables  $\{x_n\}_{n=0}^p$  exhibit the same support  $E$  which is also shared by  $x_\tau$ . Furthermore, the random variable  $x_\tau$  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  $\eta_n$  combined with an MCMC kernel are used to approximate the next posterior distribution  $\pi_{n+1}$ , the difficulty lying in specifying the sequential proposal distribution. Del Moral, Doucet, and Jasra [8] theoretically develop a coherent framework for choosing 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  $\gamma_n(x_n) = \gamma(y_{1:\tau}|x_n)^{\phi(n)} f(x_n)$ ,  $Z_n = \int_E \gamma(y_{1:\tau}|x_n)^{\phi(n)} f(x_n) dx_n$  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  $\pi_k$  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  $\pi_{n-1}$  will be close to  $\pi_n$ . We therefore define the backward kernel by detailed balance argument as

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

It gives the following weights :

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

The normalizing constant  $Z_n$  is approximated as

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

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  $\phi(n)$ , the MCMC kernels with the invariant distributions  $\pi_n(\cdot)$ , a number of particles  $M$ , of iterations  $p$ , of MCMC steps  $J$ . Adjusting these parameters can be difficult. Some guidance are given in [16] 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 [10]. 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 us 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 us 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 is equal to

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

which is exactly the weight equation of the IBIS algorithm (see [6], step 1, p. 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 the state space model Equations (1) and (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 us define the following distribution :

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

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

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

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

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

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

To deal with the time-varying dimension of  $x_t$ , we augment the support of the artificial sequence of distributions by several 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 the distributions  $\tilde{K}_k(x_k, x_k^*|x_{k-1})$ , which are composed by a user-defined distribution  $\tilde{q}_k(x_k^*|x_{k-1})$  and an MCMC kernel exhibiting  $\pi_k(x_k)$  as the invariant distribution.

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

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

By setting the distributions  $q_t(x_{t-1}|x_t^*) = \delta_{\{x_{t-1}=\theta^*, s_{1:t-1}^*\}}$ , where  $\delta_i$  denotes the probability measure concentrated at  $i$ , and  $\tilde{q}_t(x_t^*|x_{t-1}) = v_t(s_t^*|x_{t-1})\delta_{\{\theta^*, s_{1:t-1}^*=x_{t-1}\}}$ , we recover the weight equation of [7] (see Equation (20), p. 135)

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

Like in [7], only the distribution  $v_t(\cdot)$  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 the weight Equations ((5) and (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 ( $n \leq p$ ) :

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

- 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  $\kappa$  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 an 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 [8]) 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 Equation (10)). In the 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 the degeneracy problem. The choice of the tempered function  $\phi(n)$  is therefore relevant. In the spirit of a black-box algorithm as the IBIS one is, the Section 2.4 shows how the TNT algorithm automatically adapts the tempered function 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 time<sup>4</sup>. 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.,

---

<sup>4</sup> Theorem 1 in [6] 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

the ESS falls below a user-defined value  $\kappa_1 < \kappa$ ). The adaptation 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  $\kappa$ . Controlling for the degeneracy issue is therefore automated and a minimal number of effective sample size is ensured at each SMC iteration.

#### 2.4. Adaptation of the Tempered Function

Previous works on the SMC sampler usually provide a tempered function  $\phi(n)$  obtained by several empirical trials<sup>5</sup>, making these functions model-dependent. Jasra *et al.* [10] 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  $\text{ESS}_{n-1}$  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 *et al.* [21] and Beskos *et al.* [22] 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 it determines the posterior support exploration, making its choice very relevant. Chopin [6] 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 the MCMC kernel is the Metropolis-Hastings with an independent proposal distribution whose summary statistics are derived from the particles of the previous SMC step and from 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 adaptative Metropolis algorithm of random walk (RW) type that is generic, fully automated, suited for multi-modal distributions and that dominates most of the other RW alternatives in terms of sampling efficiencies. The algorithm is inspired from the heuristic Differential Evolution (DE) optimization literature (for a review, see [23]).

---

<sup>5</sup> a piecewise cooling linear function for [8] and a quadratic function for [16]

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 [24]. 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  $\mathbb{R}^d$ , 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

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

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

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

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  $x_t^{\text{best}}$  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 an MCMC context as it has been built up to explore and find the global optimum 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)–(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

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

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

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

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

These three update rules (20)–(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 [18] 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  $\eta_x$  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 a multivariate Normal posterior distribution as demonstrated in [25].<sup>6</sup> Since the posterior distribution is rarely a

---

<sup>6</sup> in the sense of  $M \rightarrow \infty$

Normal one, we prefer adapting  $F(\delta, d)$  from one SMC iteration to another so that the scale parameter is fixed during the entire MCMC moves of each SMC step. The adapting procedure is detailed below. Importantly, [18] 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 [19] and can be thought as the MCMC equivalence of the mutation (18).<sup>7</sup> 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

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

As in their paper, we set the density to  $g_W(z) \propto 1/\sqrt{1+z}$  if  $z \in [\frac{-a_W}{1+a_W}, a_W]$  (with  $a \in \mathbb{R}^+$ ) 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 [19] in the sense that the random parameter  $\delta$  can be greater than one and also because only one realization from  $Z_W$  is generated in order to update an entire new vector  $x_t^j$ . The latter modification is motivated by the success of the novel MCMC algorithm based on deterministic proposals (see the Transformation-based MCMC of [26]) 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 [19] and improved by [20]. The probability of accepting the proposal  $\tilde{x}_t$  is

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

when the density  $g_S(\cdot)$  of  $Z_S$  verifies  $z g_S(z) = g_S(z^{-1})$ . We adopt the same density function as in their paper which is given by  $g_S(z) = 1/\sqrt{z}$  for  $z \in [1/a_S, a_S]$ ,  $a_S \in \mathbb{R}^+$  and zero otherwise.

---

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

The corresponding cumulative density function, its inverse, the expectation and the variance are analytically tractable and are 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 [20]).

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, [27] suggest using a trigonometric mutation equation based on three random parameters  $x_t^{r_1}, x_t^{r_2}, x_t^{r_3}$  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

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

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

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

where  $Z_{\text{Dir}} = 1$  with probability 0.5 and -1 otherwise. Note that due to the random variable  $Z_{\text{Dir}}$ , 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 [28], updates the parameters by combining the attractiveness and the distance of the particles. For our purpose, we define the FF update as

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

where  $F_{\text{FF}}$  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

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

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

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

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

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

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

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

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

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

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, [18] 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 [29]), 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.

**Table 1.** Average of the autocorrelation times over the five dimensions for multiple update moves and different distributions.

	Stretch Move	Walk Move	DREAM
<b>Move</b>	<b>5-Dimension Normal Distribution</b>		
	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	–
	<b>5-Dimension Normal Distribution</b>		
	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	–
	<b>5-Dimension Student Distribution</b>		
	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	–

As the posterior distribution can take many different shapes, a specific MCMC kernel which may work in bags of situations can fail for some ill posterior distributions (see for example the anisotropic density in [20] or the twisted gaussian distribution in [18]). An appealing automatic approach is to use several kernels which can behave differently depending on the posterior distribution. To do so, 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)–(22), (25)–(31)) according to a multinomial probability  $p_{\text{kernel}}$ . 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, by (23) or by (24) depending on the selected mutation rule.<sup>8</sup> 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  $p_{\text{kernel}}$  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  $p_{\text{kernel}}$  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 parallelization between

<sup>8</sup> Although only the chosen mixture enters in the MH acceptance ratio, the MCMC algorithm is still valid. For further explanations, see [30], section Transition Mixtures.

the subsets. Secondly and more importantly, the SMC theory derived in Section 2 does not allow for particle interactions. 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 the MCMC kernels given by the proposal distribution (20) or (25). Then, it yields a standard SMC sampler with particle weights given by the 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.

#### *Adaptation of the Scale Parameters $F(\delta, d)$ , $a_W$ and $a_S$*

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  $F(\delta, d)$ ,  $a_W$  and  $a_S$ . Atchadé and Rosenthal [31] 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 :

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

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  $\alpha_{n-1}$  stands for the acceptance rate of the MCMC kernel of the  $n - 1$  SMC step and  $\alpha_{\text{targeted}}$  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  $\alpha_{\text{targeted}}$  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 [31] 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 step. The validity of this adaptation can be theoretically justified by [22].

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 ([32]) or the Kalman filter ([33]) operate, the state variables 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 [13,14] rely on the particle MCMC framework of [34] 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 [13] 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 the 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  $\kappa$  is recommended to be high as the evolutionary MCMC updates crucially depend on the diversification of the particles. For that reason we set it to 0.75 M. The second threshold  $\kappa_1$  that triggers a new run of the simulated annealing algorithm is chosen as 0.1 M 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 iterations is set to  $J = 90$ . This number should 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.

**Table 2.** Tuned parameters for the TNT algorithm.

Parameters	TNT Algorithm
Nb. Particles $M$	2.000
Threshold ESS $\kappa$	0.75 M
Sec. threshold ESS $\kappa_1$	0.1 M
Acc. rate $\alpha_{\text{targeted}}$	1/3
Nb. MCMC $J$	90

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

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

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

where  $\tau_1 = 0$ ,  $\tau_K = T$  and  $\tau_i$  with  $i \in [2, K]$  denotes the observation when the 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 parameters 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 of the CP-GARCH models is available on the author’s website.

**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  $\lambda$  is expressed as  $\text{Exp}(\lambda)$ (with density function :  $f(\tau|\lambda) = \lambda e^{-\lambda\tau}$ ) 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  $f(\lambda|a, b) = \frac{b^a}{\Gamma(a)} \lambda^{a-1} e^{-b\lambda}$ ).

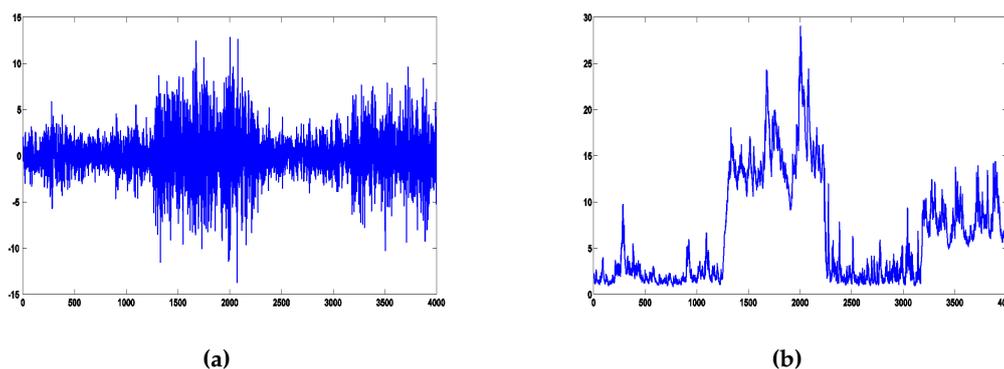
<b>Mean Parameter</b> $\forall i \in [1, K]$ :		
$\mu_i \sim N(0, 1)$		
<b>GARCH Parameters</b> $\forall i \in [1, K]$ :		
$\omega_i \sim U[0, 1]$	$\alpha_i   \beta_i \sim U[0, 1 - \beta_i]$	$\beta_i \sim U[0.2, 1]$
<b>Break Parameters</b> $\forall i \in [2, K - 1]$ :		
$d_1 = \tau_1 \sim \text{Exp}(\lambda)$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

In this section, we test our algorithm on a simulated series and a financial time series. In addition to that, we found relevant to compare our results with the algorithm of [35] which allows for an online detection of the breaks in the GARCH parameters. A proper comparison of the two approaches is detailed in Appendix C.

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.

**Table 4.** Data generating process of the Change-point-Generalized Autoregressive Conditional Heteroskedastic (CP-GARCH) model.

	$\mu$	$\omega$	$\alpha$	$\beta$	$\tau$
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	—



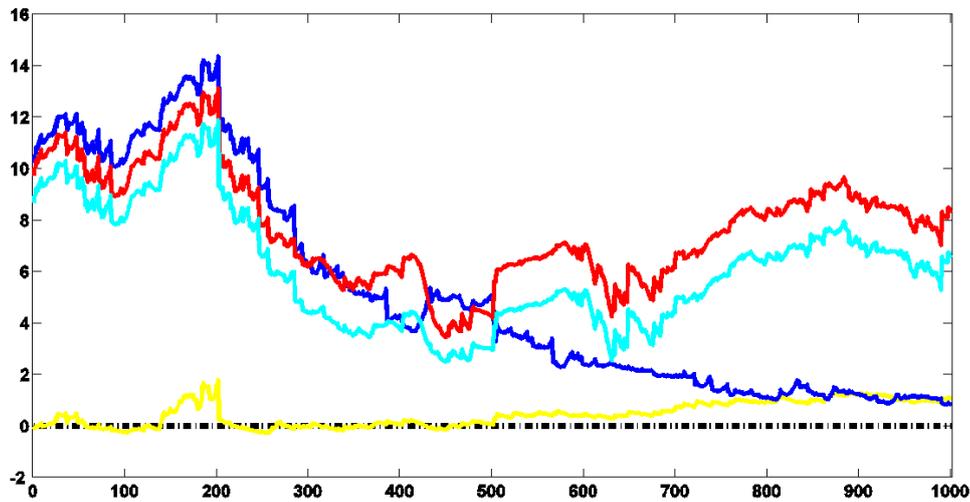
**Figure 1.** Simulated series from the data generating process (DGP) exhibited in Table 4 and its corresponding volatility over time. (a) Simulated series; (b) Volatility over time.

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 [36]). As the TNT algorithm is both an off-line and an on-line method, we start by estimating the posterior 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 Table 4, the model exhibiting three regimes should at least dominate over the first 170 observations and then the model with four regimes should gradually take the lead.

Figure 2 shows the log-Bayes factors (log-BFs) of CP-GARCH models with respect to the standard GARCH process (*i.e.*,  $K = 1$ ).<sup>9</sup> The best model over give or take the first 300 observations is the one exhibiting three regimes. Afterward, it is gradually dominated by the process with four

<sup>9</sup> We remind that the log-BF is computed as the difference of the MLL of two models. Following the informal rule of [37], if the logarithm of the Bayes factor exceeds 3, we have strong evidence in favor of the model with the highest value.

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.



**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.

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 the CP-GARCH models and correctly updates the distribution in the light of new observations.

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

	$\mu$	$\omega$	$\alpha$	$\beta$	$\tau$
Regime 1	-0.02 (0.04)	0.11 (0.04)	0.11 (0.03)	0.85 (0.04)	1253.7 (9.91)
Regime 2	0.15 (0.13)	0.67 (0.21)	0.05 (0.02)	0.91 (0.03)	2238.4 (17.42)
Regime 3	0.01 (0.04)	0.25 (0.08)	0.16 (0.03)	0.73 (0.05)	3169.4 (11.33)
Regime 4	0.06 (0.1)	0.61 (0.22)	0.07 (0.02)	0.85 (0.04)	

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

**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.

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

## 5. Empirical Application

As emphasized in the simulated exercise, the TNT algorithm allows to compare complex models through the 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.

**Table 7.** S&P 500 daily log-returns—marginal log-likelihoods (MLLs) of the CP-GARCH models given different number of regimes. The highest value is bolded.

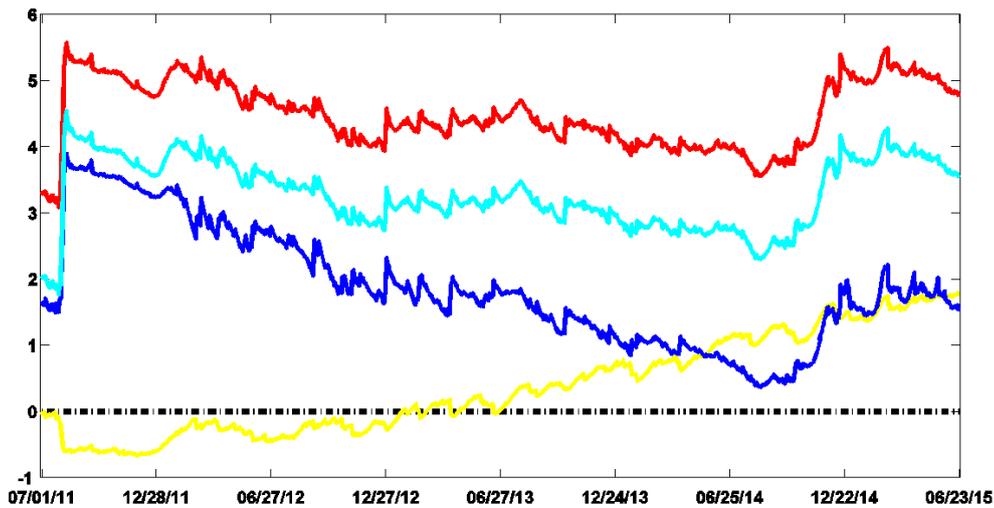
# Regimes	1	2	3	4	5
MLL	-5732.6	-5730.86	-5731.1	<b>-5727.87</b>	-5729.1

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 17 February 2007. This date sometimes refers to the beginning of the collapse of the financial system.

**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.

	$\mu$	$\omega$	$\alpha$	$\beta$	$\tau$
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)	

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 the 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 five regimes fits similarly the data but is over-parametrized compared to the same model with four regimes. The difference between the two models comes from the penalization of this over-parametrization through the prior distributions. For interested readers, Appendix B provides additional results on the CP-GARCH models such as the filtered volatility of the preferred process and a detailed comparison with other CP-GARCH models exhibiting breaks in the intercept  $\omega$  or with student innovations.



**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.

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 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.

**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 Sequential Monte Carlo (SMC) iterations. The highest probability is bolded.

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	<b>0.24</b>	0.12
100th	0.13	0.13	0.12	0.12	0.08	0.05	0.05	0.11	<b>0.15</b>	0.07
last	0.12	0.12	0.12	0.11	0.08	0.05	0.06	0.10	<b>0.17</b>	0.09
Mean	0.12	0.12	0.11	0.11	0.09	0.05	0.05	0.10	<b>0.16</b>	0.08

## 6. Conclusions

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 [5], the on-line IBIS algorithm of [6] and the RM method of [7] that all arise as special cases in the SMC sampler theory (see [8]). 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 the interactions of the particles. These MCMC updates are selected according to some probabilities that are adjusted over the SMC iterations. Furthermore, the scale parameter of these updates are also automated thanks to the method of [31]. 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 7 January 2011 to 24 June 2015. In fact, the MLL clearly indicates evidence in favor of a CP-GARCH model exhibiting four 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 [12,13] 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.

**Acknowledgments:** 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. He also thanks the referees of the Econometrics journal for their precious comments. 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.

**Author Contributions:** The author contributes entirely to the work presented in this paper.

**Conflicts of Interest:** The author declares no conflict of interest.

## Appendix

### 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  $N \times M$  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 :

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

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 :

$$\eta_n(x_n^{1:M}) = \left[ \prod_{i=1}^M f(x_1^i) \right] \prod_{k=2}^n K_k(x_k^1 | x_{k-1}^{1:M}) \prod_{q=2}^M K_k(x_k^q | x_{k-1}^{1:q-1}, x_{k-1}^{q:M})$$

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  $(N-1) \times M$  times. As proven in [18,38], 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

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

The sequential importance sampling procedure generates weights given by

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

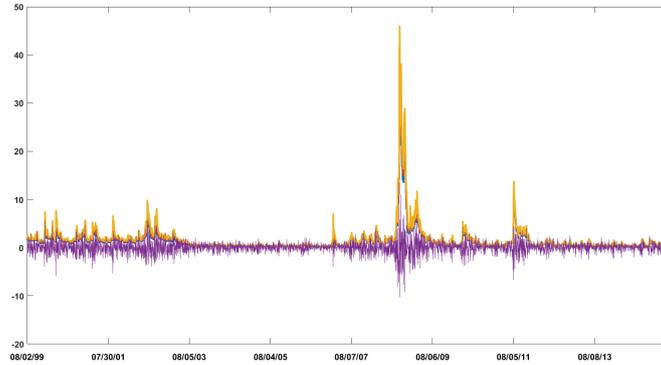
resulting in a product of independent weights exactly equal to the product of SMC sampler weights (see Equation (4)).

### B. Additional Estimation Results

From a financial econometric point of view, volatility modeling is very important. In this appendix, we go deeper in our analysis of the CP-GARCH model. To begin with, Appendix B.1 provides more details on the CP-GARCH results obtained in the empirical section. We then propose two extensions of the process by allowing for partial breaks in the intercept in Appendix B.2 and by relaxing the Normal assumption of the innovation (see Appendix B.3).

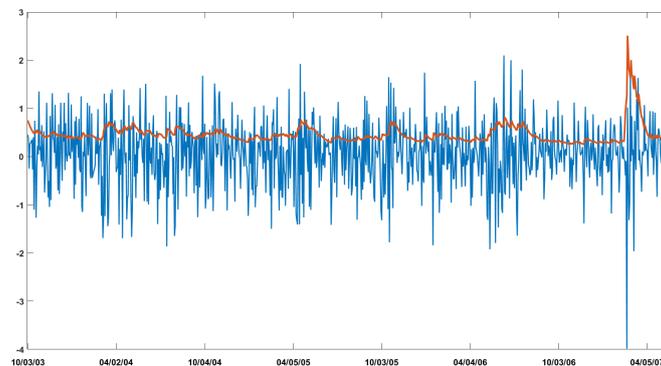
### B.1. Filtered Variance of the CP-GARCH Model

Figure B1 displays the S&P 500 daily log-returns and different quantiles of the filtered variance posterior distribution given by the best model. The volatility tracks very well the magnitude of the returns. Interestingly, since the different quantiles are almost undistinguishable, we conclude that the uncertainty on the filtered variance is very small.



**Figure B1.** S&P 500 daily log-returns in relation with the 5% (blue), 50% (red) and the 95% (yellow) quantiles of the filtered volatility posterior distribution provided by the CP-GARCH model with 4 regimes.

In the results documented in the empirical exercise, the closeness of the last two break dates may appear suspicious. Figure B2 highlights the presence of a huge extreme value occurring in 2 February 2007. The process based on the GARCH parameters derived from the quiet period preceding the financial crisis (from 2003 to 2007) cannot produce such a negative return (that amounts to  $-3.98$ ). In fact, the parameters have been estimated on a period where the second extreme value reaches 2.1 which is almost half of the outlier magnitude. Consequently, a new regime is created just to handle this high return. In this specific case, the flexibility of the CP-GARCH process turns out to be negative as creating a regime for an outlier is unappealing and counter-productive. To avoid the detection of outliers, one can for instance increase the persistence in the transition probabilities  $\lambda$  since it will penalize very short regimes.



**Figure B2.** S&P 500 daily log-returns over the quiet period preceding the financial crisis in relation with the median of the filtered volatility posterior distribution provided by the CP-GARCH model with four regimes.

B.2. A CP-GARCH Process with Partial Breaks

Instead of considering breaks in all parameters, one can be interested in a more parsimonious model where only the intercept  $\omega$  evolves over time. In fact, as the latter parameter is related to the long-run volatility of a regime, the partial-break model can highlight if it is the volatility persistence that is changing over time or only the long-term variance.

We start by a simulation exercise. A series of 4000 observations exhibiting breaks only in the  $\omega$  parameter has been generated from the DGP documented in Table B1.

**Table B1.** Data generating process of the CP-GARCH model exhibiting breaks in  $\omega$ . The long run variance of each regime is given in the row  $\frac{\omega}{1-\alpha-\beta}$ .

	$\mu$	$\omega$	$\alpha$	$\beta$	$\tau$	$\frac{\omega}{1-\alpha-\beta}$
Regime 1	0	0.1	0.1	0.85	1210	2
Regime 2	0	0.3	—	—	2060	6
Regime 3	0	0.05	—	—	3030	1
Regime 4	0	0.4	—	—	—	8

We carry out a In-sample simulation by fixing  $\tau = 4000$  (no online estimation) and so by estimating only the posterior distribution of the model parameters given all the observations. To begin with, Table B2 delivers the MLLs for the partial CP-GARCH models exhibiting different number of regimes. In this specific case, the criterion selects the true number of regimes.

**Table B2.** Simulated series from the DGP Table B1—MLLs of the partial CP-GARCH models given different number of regimes. The highest value is bolded.

# Regimes	1	2	3	4	5
MLL	-7746.57	-7748.83	-7749.35	<b>-7738.38</b>	-7738.68

Table B3 provides summary statistics of the targeted distribution. The break dates are well detected and the highest standard deviation is related to the first break which is in agreement with the DGP. In fact, the difference in the long run variance is the smallest when moving from the first to the second regime. We also observe that the persistence parameters as well as the intercepts are quite accurately estimated.

**Table B3.** Simulated series from the DGP Table B1—posterior means of the parameters of the partial CP-GARCH model with four regimes and their corresponding standard deviations.

	$\mu$	$\omega$	$\alpha$	$\beta$	$\tau$
Regime 1	0.03 (0.04)	0.16 (0.02)	0.11 (0.02)	0.81 (0.04)	1197.78 (58.29)
Regime 2	-0.05 (0.09)	0.52 (0.17)	— —	— —	2062.17 (22.83)
Regime 3	0.03 (0.04)	0.08 (0.02)	— —	— —	3033.17 (22.55)
Regime 4	0.04 (0.1)	0.7 (0.20)	— —	— —	

We now turn to the empirical series and study the S&P 500 daily log returns. As before, we select the best model according to the MLL. Table B4 documents the criterion for the CP-GARCH and the partial CP-GARCH processes given several numbers of breaks. Interestingly, if only the intercept varies over time, the preferred model solely exhibits one break. Additionally, the model where all the parameters vary is preferred which gives evidence in favor of some breaks in the volatility persistence. The break of the partial CP-GARCH model corresponds to the end of the dot com bubble since the posterior mean of the break date is 2 February 2003 with a standard deviation of 103 days.

**Table B4.** S&P 500 daily log-returns — MLLs of the CP-GARCH models given different number of regimes for two distinct specifications. The partial CP-GARCH model denotes a CP-GARCH process where only the intercept is allowed to exhibit breaks. The standard CP-GARCH model exhibits breaks in all its parameters. The highest value is bolded.

# Regimes	1	2	3	4	5
Partial CP-GARCH Model					
MLL	-5733.71	<b>-5732.81</b>	-5737.97	-5735.85	-5734.47
Standard CP-GARCH Model					
MLL	-5732.6	-5730.86	-5731.1	<b>-5727.87</b>	-5729.1

### B.3. A CP-GARCH Process with Student Innovations

As it is well known that financial returns exhibit fat tails, we now study the t-CP-GARCH model, *i.e.*, a CP-GARCH process with *t*-student innovations. To begin with, a series of 4000 observations from the DGP displayed in Table B5 has been simulated. Figure B3 shows a simulated series and the corresponding conditional volatility over time.<sup>10</sup>

For each regime *i*, the degree of freedom  $dof_i$  is an additional parameter to be estimated. We constraint the parameter  $dof_i \in [2, 100]$  and use the non-linear transformation  $d\tilde{of}_i = \log(\frac{dof_i-2}{100-dof_i})$  in order to map it on the real line. For each transformed parameter, we choose as a prior distribution a Normal distribution with mean equal to 0 and a variance amounting to 2. As in the previous exercises, we fix  $\tau = 4000$  and simulate the final posterior distribution. Table B6 documents the MLLs for the t-CP-GARCH process with different numbers of regimes. The criterion selects the right specification.

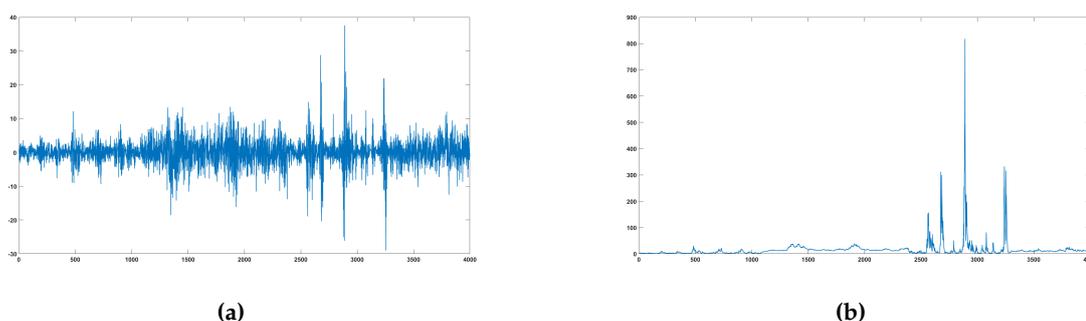
**Table B5.** Data generating process of the t-CP-GARCH model. The degree of freedom is denoted by dof.

	$\mu$	$\omega$	$\alpha$	$\beta$	$\tau$	
Regime 1	0	0.1	0.1	0.85	15	1080
Regime 2	0	0.3	0.03	0.95	40	2390
Regime 3	0	0.25	0.2	0.70	5	3280
Regime 4	0	0.4	0.05	0.9	10	—

<sup>10</sup> taking into account the degree of freedom of the student distribution.

**Table B6.** Simulated series from the DGP Table B5—MLLs of the t-CP-GARCH models given different number of regimes. The highest value is bolded.

# Regimes	1	2	3	4	5
MLL	-10132.12	-10124.63	-10106.30	<b>-10095.56</b>	-10098.63



**Figure B3.** Simulated series from the DGP Table B5 and its corresponding volatility over time. (a) Simulated series; (b) Volatility over time.

Table B7 gives the standard summary statistics of the posterior distribution. We observe that all the parameters are quite accurately estimated. Obviously, as long as the student distribution gets closer to the Normal one, the likelihood function becomes flatter with respect to the degree of freedom. Consequently, it makes difficult to obtain a precise estimation of it.

**Table B7.** Simulated series from the DGP Table B5—posterior means of the parameters of the t-CP-GARCH model with four regimes and their corresponding standard deviations.

	$\mu$	$\omega$	$\alpha$	$\beta$	$\tau$	dof
Regime 1	0.03 (0.05)	0.16 (0.05)	0.16 (0.03)	0.8 (0.03)	1094.74 (31.65)	37.25 (22.39)
Regime 2	0.06 (0.11)	0.27 (0.12)	0.04 (0.01)	0.94 (0.01)	2373.9 (14.1)	51.42 (22.74)
Regime 3	0.17 (0.07)	0.25 (0.08)	0.2 (0.03)	0.69 (0.04)	3306.04 (22.71)	5.14 (0.89)
Regime 4	0.13 (0.13)	0.62 (0.28)	0.03 (0.02)	0.91 (0.04)		42.78 (24.12)

We now apply the t-CP-GARCH process to the empirical series. Table B8 provides the MLLs for many different numbers of regimes. There is a clear evidence in favour of the t-CP-GARCH process compared to the standard CP-GARCH one. Moreover, the best model does not exhibit any break. It emphasizes that the Normal innovations is clearly rejected and that the breaks were mainly detected to capture extreme values.

**Table B8.** S&P 500 daily log-returns—MLLs of the CP-GARCH models given different number of regimes for two distinct specifications. The t-CP-GARCH model denotes a CP-GARCH process with t-student innovations. The innovations of the standard CP-GARCH model are driven by a Normal distribution. The highest value is bolded.

# Regimes	1	2	3	4	5
t-CP-GARCH model					
MLL	<b>-5673.98</b>	-5675.91	-5678.11	-5682.85	-5686.78
Standard CP-GARCH model					
MLL	-5732.6	-5730.86	-5731.1	<b>-5727.87</b>	-5729.1

### C. Comparison with the Online SMC Algorithm

He and Maheu [35] provide an online algorithm able to detect structural breaks on the fly in a process which exhibits the path dependence issue. The paper strongly relies on [39] which develop an auxiliary particle filter (hereafter APF, see [40]) to estimate the static parameters. The sequence of targeted distributions in a standard APF evolves over the time domain and consists in the posterior distributions given a growing number of observations. As a consequence, the number of iteration equals to the sample size.

We propose a formal comparison of the two approaches. For the sake of completeness, the next section briefly reviews the method of [35] and discusses the important differences with the TNT sampler. Eventually, in Section C.2, we apply the algorithm on the two financial time series of the paper to empirically compare the two approaches.

#### C.1. The APF Algorithm for CP-GARCH Models

Following [35], the CP-GARCH model is specified as follows<sup>11</sup>

$$\begin{aligned} y_t &= \mu_{s_t} + \epsilon_t \quad \text{with} \quad \epsilon_t | y_{1:t-1} \sim N(0, \sigma_t^2), \\ \sigma_t^2 &= \omega_{s_t} + \alpha_{s_t} \epsilon_{t-1}^2 + \beta_{s_t} \sigma_{t-1}^2, \end{aligned}$$

where  $s_t$  is a discrete latent variable taking values in  $[1, K]$  that is driven by a Markov-chain. The  $K$  by  $K$  transition matrix is given by

$$P = \begin{pmatrix} p_1 & 1-p_1 & 0 & \dots & 0 \\ 0 & p_2 & 1-p_2 & \dots & 0 \\ & & \dots & & \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix}$$

As already mentioned, the CP-GARCH model exhibits the path dependence issue which makes the standard inference relying on the forward-backward filter not appropriate. To solve the problem, [35] use an APF algorithm (see [40]) in combination with an artificial sequence for the static parameters (see [39]). The static parameters are assumed to evolve over time according to a shrinkage random walk process. As this process requires unbounded support for all the model

<sup>11</sup> To compare with the specification of the paper, we have included the mean parameters  $\{\mu_i\}_{i=1}^K$ .

parameters, they map the bounded parameters on the real line using non-linear transformations. To be precise, if a parameter  $r$  belongs to  $[a,b]$  (with  $0 < a < b$ ), they apply the mapping  $\tilde{r} = \log \frac{r-a}{b-r} \in \mathfrak{R}$ . For notational convenience, we gather all the model parameters into the set  $\theta = \{\mu_1, \omega_1, \alpha_1, \beta_1, p_1, \dots, \mu_K, \omega_K, \alpha_K, \beta_K, p_K\}$  and their associated mapping into  $\tilde{\theta}$ . The key idea of [35] consists in including the lagged conditional variances into the set of the SMC particles (and the previous error terms as we have included mean parameters  $\{\mu_i\}_{i=1}^K$  in the CP-GARCH specification). Let define the filtration  $\mathcal{F}_{t-1} = \{\tilde{\theta}_{t-1}^i, s_{t-1}^i, \sigma_{t-1}^i, \epsilon_{t-1}^i, W_{t-1}^i\}_{i=1}^N$  generated by the SMC at time  $t - 1$ . By Bayes' theorem, we have

$$\pi(\tilde{\theta}_t, s_t | y_{1:t}, \mathcal{F}_{t-1}) \propto f(y_t | y_{1:t-1}, \mathcal{F}_{t-1}, s_t, \tilde{\theta}_t) f(s_t | y_{1:t-1}, \mathcal{F}_{t-1}, \tilde{\theta}_t) f(\tilde{\theta}_t | y_{1:t-1}, \mathcal{F}_{t-1}) \quad (C1)$$

In Equation (C1), the static parameters  $\theta$  are now time-varying and the distribution  $f(\tilde{\theta}_t | y_{1:t-1}, \mathcal{F}_{t-1})$  is approximated by a mixture as follows

$$f(\tilde{\theta}_t | y_{1:t-1}, \mathcal{F}_{t-1}) \approx \sum_{i=1}^N W_{t-1}^i N(\tilde{\theta}_{\mu,t-1}, (1 - a^2)V_{t-1}) \quad \text{with } \tilde{\theta}_{\mu,t-1} = a\tilde{\theta}_{t-1} + (1 - a)\tilde{\theta}_{t-1}$$

where  $\tilde{\theta}_{t-1} = \sum_{i=1}^N W_{t-1}^i \tilde{\theta}_{t-1}^i$ ,  $V_{t-1} = \sum_{i=1}^N W_{t-1}^i (\tilde{\theta}_{t-1}^i - \tilde{\theta}_{t-1})(\tilde{\theta}_{t-1}^i - \tilde{\theta}_{t-1})'$ ,  $a = (3\delta - 1)/(2\delta)$  and  $\delta$  is a discount factor belonging to  $[0, 1]$ , typically set around 0.95-0.99. Note also that as the filtration includes the previous conditional variances and error terms, the path dependence issue has been solved since the density function  $f(y_t | y_{1:t-1}, \mathcal{F}_{t-1}, s_t, \tilde{\theta}_t)$  is a computable Normal density. The APF algorithm of [35] is briefly detailed in Algorithm 1.

The APF algorithm exposed in Algorithm 1 is distinct in many ways from the TNT sampler. We believe that none of the two approaches dominate and that the algorithmic choice should depend on the needs of the user. In fact, if she is interested in smooth estimates of the states and in an exact posterior simulation (besides the monte carlo error), then she should use the TNT sampler. On the contrary, if she needs a simulation of all the posterior distributions (given only the very first observation to the entire sample) and if she wants a fast estimation, then she should definitely choose the APF approach. Below we discuss in deep details the noticeable differences of the methods.

- The complexity of the APF is  $O(NT)$  which is by far faster than the TNT sampler. However, in order to avoid particles' discrepancies, the number of particles needs to be very high. He and Maheu [35] recommend  $N = 300.000$ .
- The algorithm of [35] provides an online detection but does not deliver smooth estimates of the states while the TNT sampler gives the smooth probabilities of the states and an online detection when the algorithm switches to the time domain.
- The APF algorithm relies on an approximation of the distribution  $f(\tilde{\theta}_t | y_{1:t-1}, \mathcal{F}_{t-1})$  and on an artificial process for the static parameters which depends on a user-defined parameter  $\delta$ . On the contrary, the TNT sampler is not based on any approximations and can directly estimate static parameters.
- The TNT sampler updates the particles according to an ergodic MCMC kernel which converges to the targeted distribution. By contrast, the APF algorithm is sensible to outliers (as discussed in [40]) and if the number of particles is too low, the APF method may not well approximate the final posterior distribution.
- The algorithm of [35] exhibits the appealing advantage of estimating the number of regimes. In fact, one can fix the number of regimes  $K$  to a very high value and counts the number of detected regimes in the output. Oppositely, we have to fix the number of regimes before estimating the CP-GARCH model. However, it is more a model issue than an algorithmic one as a CP-GARCH model with an undetermined number of regimes (which relies on hierarchical Dirichlet processes, see [41]) can be estimated by the TNT sampler.

- As mentioned in [35], the Algorithm 1 requires a prior on the transition probability that is close to one for obtaining sensible results since the estimation of the states is sensitive to the outliers. In the same spirit, the other priors need to be informative in order to improve the detection of new regimes (and to limit the number of particles). The TNT sampler allows to freely choose the prior distributions of the model parameters.
- As long as the probability of being in a state is not equal to zero, the algorithm of [35] can get back to an already visited regime. This feature is highlighted in the simulation exercise below.
- Since the TNT sampler relies on an MCMC kernel to update the particles, the detections of breaks (exacerbated in the second phase when observations are introduced one by one) highly depend on the mixing of the MCMC kernel and therefore on the number  $J$  of MCMC iterations. On the contrary, the online detection of the APF algorithm will depend on the number of particles and how the prior distributions are diffuse. Therefore, for online estimations, the two algorithms exhibit pros and cons which makes their performance depending on the problem at hand.

---

**Algorithm 1** APF algorithm for the CP-GARCH(1,1) model
 

---

```

for  $i = 1$  to  $N$  do
  Sample  $s_1^i \sim f(s_1)$ 
  Sample  $\tilde{\theta}_1^i \sim f(\tilde{\theta}_1)$  where  $f(\tilde{\theta}_1)$  denotes the prior density of  $\tilde{\theta}$ 
  Compute the un-normalized weight :  $w_1^i = f(y_1 | s_1^i, \tilde{\theta}_1^i)$ 
  Save the conditional standard error  $\sigma_1^i$  and the error term  $\epsilon_1^i$ 
end for
For  $i=1, \dots, N$ , compute normalized auxiliary weights :  $W_1^i = w_1^i / \sum_{r=1}^N w_1^r$ 
for  $t = 2$  to  $T$  do
  Prediction step :
  for  $i = 1$  to  $N$  do
    Compute  $\tilde{\theta}_{\mu,t}^i$  and the mode  $s_{\mu,t}^i$  of  $f(s_t | s_{t-1}^i, \tilde{\theta}_{t-1}^i)$ 
    Compute the un-normalized auxiliary weight :  $g_t^i = f(y_t | y_{1:t-1}, s_{\mu,t}^i, \tilde{\theta}_{\mu,t}^i, \sigma_{t-1}^i) W_{t-1}^i$ 
  end for
  For  $i=1, \dots, N$ , compute normalized auxiliary weights :  $G_t^i = g_t^i / \sum_{r=1}^N g_t^r$ 
  Resample step on the auxiliary weights :
  For  $i=1, \dots, N$ , assign a particle to each  $r^i$  by stratified sampling
  (see [42])
  Update step :
  for  $i = 1$  to  $N$  do
    Sample  $\tilde{\theta}_t^i \sim N(\tilde{\theta}_{\mu,t-1}^i, (1 - a^2)V_{t-1})$ 
    Sample  $s_t^i \sim f(s_t | s_{t-1}^i, \tilde{\theta}_t^i)$ 
    Compute un-normalized weight :  $w_t^i = \frac{f(y_t | y_{1:t-1}, s_t^i, \tilde{\theta}_t^i, \sigma_{t-1}^i)}{f(y_t | y_{1:t-1}, s_{\mu,t}^i, \tilde{\theta}_{\mu,t}^i, \sigma_{t-1}^i)}$ 
    Save the conditional standard error  $\sigma_t^i$  and the error term  $\epsilon_t^i$ 
  end for
  For  $i=1, \dots, N$ , compute normalized auxiliary weights :  $W_t^i = w_t^i / \sum_{r=1}^N w_t^r$ 
end for

```

---

## C.2. Comparison of the He and Maheu Algorithm with the TNT Sampler

We estimate the CP-GARCH model with the algorithm of [35]. To begin with, Table C1 summarizes the prior distributions of [35]. In fact, the algorithm requires transition probabilities close to one as well as quite informative priors for the other parameters to correctly perform. Regarding the tuning parameters, we follow the recommendations of [35] and fix  $\delta$  to 0.99,  $N$  to

300.000 and the number  $K$  of regimes to 6.

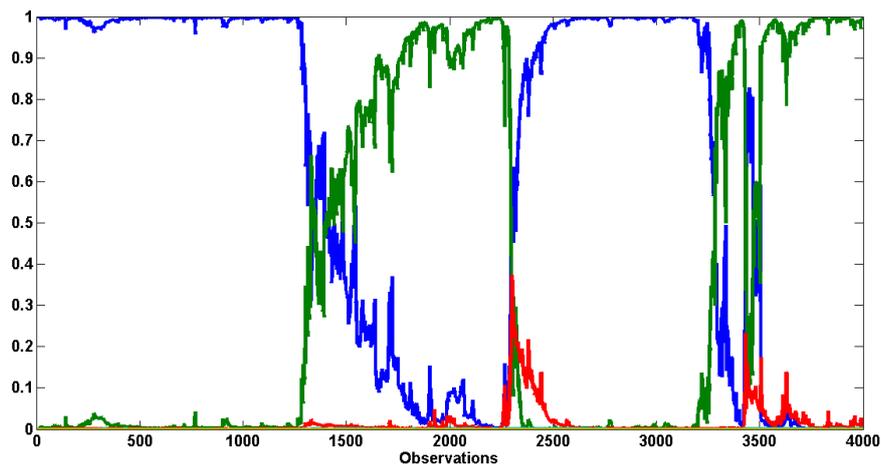
**Table C1.** Prior distributions of the CP-GARCH parameters. The distribution  $N(a, b)$  denotes the Normal distribution with expectation  $a$  and variance  $b$  and  $\text{Gam}(a, b)$  stands for the gamma distribution with  $a$  being the shape parameter and  $b$  the scale one. The beta distribution is expressed as  $\text{Beta}(a, b)$  with shape parameters  $a$  and  $b$ .

<b>Mean parameter</b> $\forall i \in [1, K]$ : $\mu_i \sim N(0, 0.1)$
<b>GARCH parameters</b> $\forall i \in [1, K]$ : $\omega_i \sim \text{Gam}[1, 0.2]$ $\alpha_i \sim \text{Beta}[1, 8]$ $\beta_i \sim \text{Beta}[4, 1]$
<b>Transition parameters</b> $\forall i \in [1, K - 1]$ : $\tilde{p}_i \sim N(10, 1)$

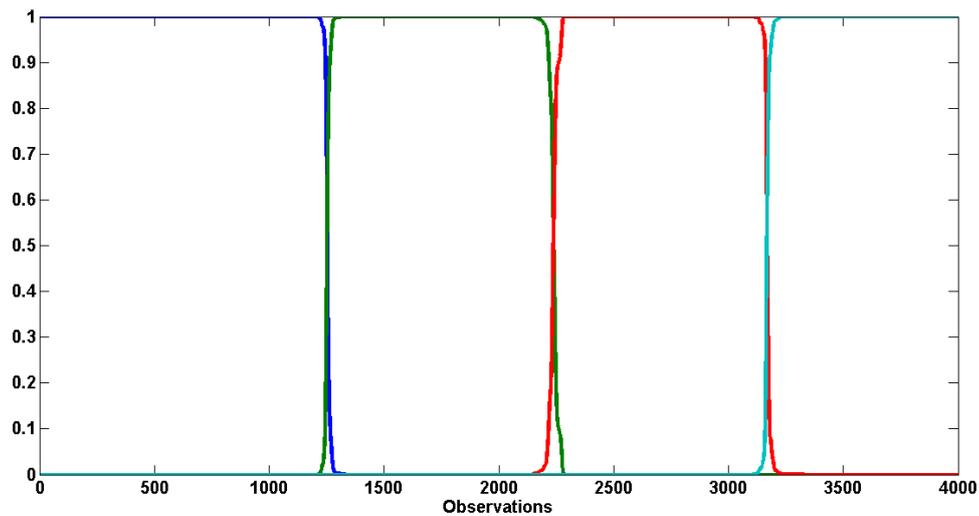
### C.2.1. Results on the Simulated Series

We start by estimating the model on the same simulated series as in the paper. The DGP is given in Table C2.

As the method cannot provide smooth probabilities of the states, Figure C1 displays the filtered distribution (*i.e.*,  $f(s_t | y_{1:t})$ ). All the breaks have been correctly detected. However, the number of detected regimes only amounts to two. In fact, the probability of being in regime 1 was not zero when the third break has been detected and since the third and the first regimes exhibit similar parameters (see DGP Table C2), the weights of the particles related to the first states were again influential. To compare with the TNT estimation, Figure C2 documents the smooth probabilities (*i.e.*,  $f(s_t | y_{1:T})$ ). The breaks are sharply identified as the estimation uses future observations to confirm the presence of a change.



**Figure C1.** Simulated series—filtered probabilities of the states given by the He and Maheu’s algorithm. The probabilities of the first state are displayed in blue, the second in green, the third in red and the last three regimes have virtually zero probabilities over the entire sample.



**Figure C2.** Simulated series—smooth probabilities of the states given by the TNT algorithm. The probabilities of the first state are displayed in blue, the second in green, the third in red and the last one in turquoise.

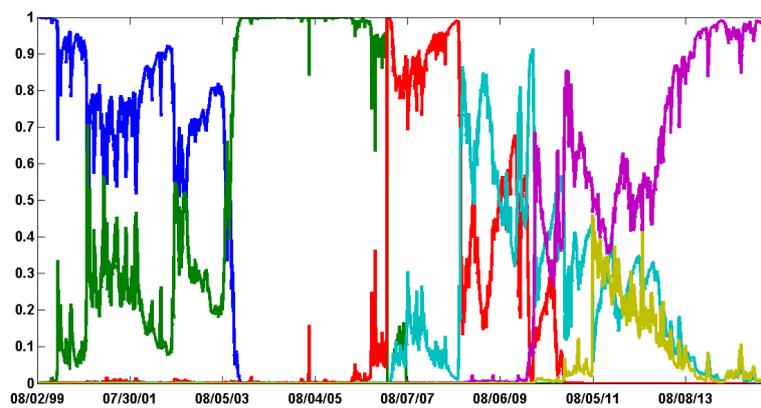
**Table C2.** Data generating process of the CP-GARCH model.

	$\mu$	$\omega$	$\alpha$	$\beta$	$\tau$
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	—

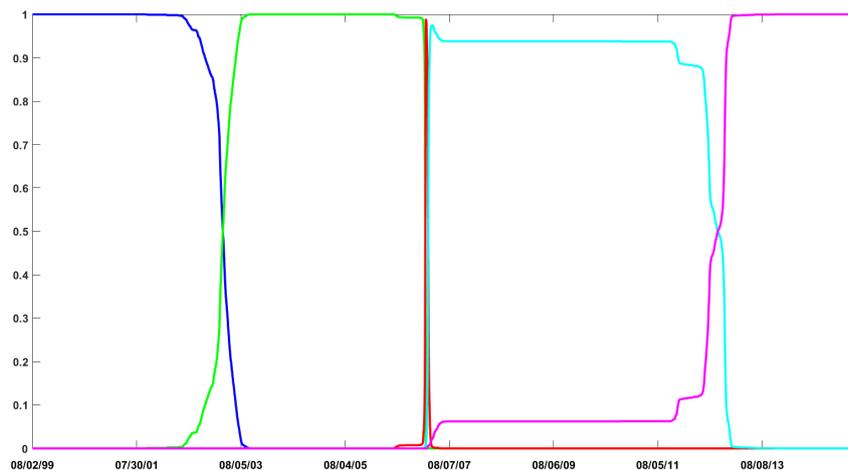
### C.2.2. Results on the S&P 500 Daily Percentage Returns

We now compare the two algorithms on the empirical time series used in the paper. Figure C3 documents the filtered probabilities of the states provided by the He and Maheu's algorithm. The graphic shows the detection of at least five regimes which means one additional break compared to the preferred model obtained by the TNT sampler. To compare the break detection, Figure C4 documents the smooth probabilities of the TNT sampler for the CP-GARCH model with five regimes (the second best model according to the MLL). The two approaches give very similar results. Interestingly, all the breaks obtained by the TNT sampler are also identified by the APF algorithm. Two differences are worth discussing.

1. The algorithm of [35] detects two breaks during the financial crisis while only one regime covers the period for the TNT sampler. Nevertheless the presence of two regimes in this spell is quite uncertain by judging the probabilities of the third regime (in red) which remain above 20% over the entire period.
2. Compared to the online detection of [35], the uncertainties around the break dates are very small in Figure C4 as the smooth probabilities take the future observations into account.



**Figure C3.** S&P 500 daily log-returns—filtered probabilities of the states given by the He and Maheu’s algorithm. The probabilities of the first state are given in blue, the second in green, the third in red and the fourth in turquoise, the fifth in purple and the last one in pale green.



**Figure C4.** S&P 500 daily log-returns - smooth probabilities of the states given by the TNT algorithm for the CP-GARCH model exhibiting five regimes. The probabilities of the first state are displayed in blue, the second in green, the third in red, the fourth in turquoise and the fifth in purple.

## References

1. Geweke, J. Bayesian Inference in Econometric Models Using Monte Carlo Integration. *Econometrica* **1989**, *57*, 1317–1339.
2. Smith, A.F.M.; Gelfand, A.E. Bayesian Statistics without Tears: A Sampling-Resampling Perspective. *Am. Stat.* **1992**, *46*, 84–88.
3. Gordon, N.; Salmond, D.; Smith, A.F.M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F Radar Signal Process.* **1993**, *140*, 107–113.
4. Chib, S.; Nardari, F.; Shephard, N. Markov chain Monte Carlo methods for stochastic volatility models. *J. Econom.* **2002**, *108*, 281–316.

5. Neal, R.M. Annealed Importance Sampling. *Stat. Comput.* **1998**, *11*, 125–139.
6. Chopin, N. A Sequential Particle Filter Method for Static Models. *Biometrika* **2002**, *89*, 539–551.
7. Gilks, W.R.; Berzuini, C. Following a moving target—Monte Carlo inference for dynamic Bayesian models. *J. R. Stat. Soc. B* **2001**, *63*, 127–146.
8. Del Moral, P.; Doucet, A.; Jasra, A. Sequential Monte Carlo samplers. *J. R. Stat. Soc. B* **2006**, *68*, 411–436.
9. Jasra, A.; Stephens, D.A.; Holmes, C.C. On population-based simulation for static inference. *Stat. Comput.* **2007**, *17*, 263–279.
10. Jasra, A.; Stephens, D.A.; Doucet, A.; Tsagaris, T. Inference for Lévy-Driven Stochastic Volatility Models via Adaptive Sequential Monte Carlo. *Scand. J. Stat.* **2011**, *38*, 1–22.
11. Jeremiah, E.; Sisson, S.; Marshall, L.; Mehrotra, R.; Sharma, A. Bayesian calibration and uncertainty analysis of hydrological models: A comparison of adaptive Metropolis and sequential Monte Carlo samplers. *Water Resour. Res.* **2011**, *47*, doi:10.1029/2010WR010217.
12. Fearnhead, P.; Taylor, B. An adaptive Sequential Monte Carlo Sampler. *Bayesian Anal.* **2013**, *8*, 411–438.
13. Fulop, A.; Li, J. Efficient learning via simulation: A marginalized resample-move approach. *J. Econom.* **2013**, *176*, 146–161.
14. Chopin, N.; Jacob, P.E.; Papaspiliopoulos, O. SMC2: An efficient algorithm for sequential analysis of state space models. *J. R. Stat. Soc. B* **2013**, *75*, 397–426.
15. Durham, G.; Geweke, J. Adaptive Sequential Posterior Simulators for Massively Parallel Computing Environments. In *Bayesian Model Comparison (Advances in Econometrics)*; Jeliazkov, I., Poirier, D.J., Eds.; Emerald Group Publishing Limited: Bingley, UK, 2014; Volume 34, pp. 1–44.
16. Herbst, E.; Schorfheide, F. Sequential Monte Carlo Sampling for DSGE Models. *J. Appl. Econom.* **2014**, *29*, 1073–1098.
17. Bauwens, L.; Dufays, A.; Rombouts, J. Marginal Likelihood for Markov Switching and Change-point GARCH Models. *J. Econom.* **2013**, *178*, 508–522.
18. Vrugt, J.A.; ter Braak, C.J.F.; Diks, C.G.H.; Robinson, B.A.; Hyman, J.M.; Higdon, D. Accelerating Markov Chain Monte Carlo Simulation by Differential Evolution with Self-Adaptative Randomized Subspace Sampling. *Int. J. Nonlinear Sci. Numer. Simul.* **2009**, *10*, 271–288.
19. Christen, J.A.; Fox, C. A general purpose sampling algorithm for continuous distributions (the t-walk). *Bayesian Anal.* **2010**, *5*, 263–281.
20. Foreman-Mackey, D.; Hogg, D.W.; Lang, D.; Goodman, J. Emcee: The MCMC Hammer. *PASP* **2013**, *125*, 306–312.
21. Del Moral, P.; Doucet, A.; Jasra, A. On adaptive resampling strategies for sequential Monte Carlo methods. *Bernoulli* **2012**, *18*, 252–278.
22. Beskos, A.; Jasra, A.; Thiery, A. On the Convergence of Adaptive Sequential Monte Carlo Methods. 2013. Available online: <http://arxiv.org/pdf/1306.6462v2.pdf> (accessed on 16th February 2016).
23. Das, S.; Suganthan, P. Differential Evolution: A Survey of the State-of-the-Art. *IEEE Trans Evolut. Comput.* **2011**, *15*, 4–31.
24. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optimiz.* **1997**, *11*, 341–359.
25. Roberts, G.O.; Rosenthal, J.S. Optimal scaling for various Metropolis-Hastings algorithms. *Stat. Sci.* **2001**, *16*, 351–367.
26. Dutta, S.; Bhattacharya, S. Markov chain Monte Carlo based on deterministic transformations. *Stat. Methodol.* **2014**, *16*, 100–116.
27. Fan, H.-Y.; Lampinen, J. A Trigonometric Mutation Operation to Differential Evolution. *J. Glob. Optim.* **2003**, *27*, 105–129.
28. Yang, X.-S. Firefly Algorithms for Multimodal Optimization. In *Stochastic Algorithms: Foundations and Applications*; Watanabe, O., Zeugmann, T., Eds.; Springer: Berlin, Germany, 2009; pp. 169–178.
29. Geyer, C.J. Practical Markov Chain Monte Carlo. *Stat. Sci.* **1992**, *7*, 473–511.
30. Geweke, J. *Contemporary Bayesian Econometrics and Statistics*; Wiley Series in Probability and Statistics; John Wiley and Sons, Inc.: Hoboken, NJ, USA, 2005.
31. Atchadé, Y.; Rosenthal, J. On adaptive Markov chain Monte Carlo algorithms. *Bernoulli* **2005**, *11*, 815–828.
32. Rabiner, L.R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **1989**, *77*, 257–286.

33. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *Trans. ASME J. Basic Eng.* **1960**, *82*, 35–45.
34. Andrieu, C.; Doucet, A.; Holenstein, R. Particle Markov Chain Monte Carlo Methods. *J. R. Stat. Soc. B* **2010**, *72*, 269–342.
35. He, Z.; Maheu, J. Real Time Detection of Structural Breaks in GARCH Models. *Comput. Stat. Data Anal.* **2010**, *54*, 2628–2640.
36. Chib, S. Estimation and comparison of multiple change-point models. *J. Econom.* **1998**, *86*, 221–241.
37. Kass, R.; Raftery, A. Bayes Factors. *J. Am. Stat. Assoc.* **1995**, *90*, 773–795.
38. Bauwens, L.; Dufays, A.; De Backer, B. Estimating and forecasting structural breaks in financial time series. *J. Empir. Financ.* **2011**, doi:10.1016/j.jempfin.2014.06.008.
39. Liu, J.; West, M. Combined Parameter and State Estimation in Simulation-Based Filtering. In *Sequential Monte Carlo Methods in Practice*; Doucet, A., de Freitas, N., Gordon, N., Eds.; Springer: New York, USA, 2001; pp. 197–224.
40. Pitt, M.K.; Shephard, N. Filtering via Simulation: Auxiliary Particle Filters. *J. Am. Stat. Assoc.* **1999**, *94*, 590–599.
41. Ko, S.; Chong, T.; Ghosh, P. Dirichlet Process Hidden Markov Multiple Change-point Model. *Bayesian Anal.* **2015**, *2*, 275–296.
42. Carpenter, J.; Clifford, P.; Fearnhead, P. Improved particle filter for nonlinear problems. *IEE Proc. Radar Sonar Navig.* **1999**, *146*, 2–7.



© 2016 by the author; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons by Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).