View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Universidad Carlos III de Madrid e-Archivo

Noname manuscript No.
(will be inserted by the editor)

A simple scheme for the parallelisation of particle filters and its application to the tracking of complex stochastic systems

Dan Crisan · Joaquín Míguez ·
Gonzalo Ríos

arXiv:1407.8071v2 [stat.CO] 9 Oct 2015

Received: date / Accepted: date

Abstract Considerable effort has been devoted to the design of schemes for the parallel, or distributed, implementation of particle filters. The approaches vary from the totally heuristic to the mathematically well-principled. However, the former are largely based on (often loose) approximations that prevent the claim of any rigorous guarantees of convergence, whereas the latter involve considerable overhead to ensure the proper interaction of particles, which impairs the efficiency of the intended parallelisation. In this paper we investigate the use of possibly the simplest scheme for the parallelisation of the standard particle filter, that consists in splitting the computational budget into M fully independent particle filters with N particles each, and then obtaining the desired estimators by averaging over the M independent outcomes of the filters. This approach minimises the parallelisation overhead yet displays highly desirable theoretical properties. Under very mild assumptions, we analyse the mean square error (MSE) of the estimators of 1-dimensional statistics of the optimal filtering distribution and show explicitly the effect of parallelisation scheme on the convergence rate. Specifically, we study the decomposition of the MSE into variance and bias components, to show that the former decays as $\frac{1}{MN}$, i.e., linearly with the total number of particles, while the latter converges towards 0 as $\frac{1}{N^2}$. Parallelisation, therefore, has the obvious advantage of dividing the running times while preserving the (asymptotic) performance of the particle filter. Following this lead, we propose a time-error index to compare schemes with different degrees of parallelisation. Finally, we provide two numerical examples. The first one deals with the tracking of a Lorenz 63 chaotic system with dynamical noise and partial (noisy) observations, while

Dan Crisan
Department of Mathematics, Imperial College London (UK).
E-mail: d.crisan@imperial.ac.uk

Joaquín Míguez
School of Mathematical Sciences, Queen Mary University of London (UK).
E-mail: j.miguez@qmul.ac.uk

Gonzalo Ríos
Department of Signal Theory & Communications, Universidad Carlos III de Madrid (Spain).
E-mail: griosm@tsc.uc3m.es

the second example involves a dynamical network of modified FitzHugh-Nagumo (FH-N) stochastic nodes. The latter is a large dimensional system ($\approx 3,000$ state variables in our computer experiments) designed to numerically reproduce typical electrical phenomena observed in the atria of the human heart. In both examples, we show how the proposed parallelisation scheme attains the same approximation accuracy as a centralised particle filter with only a small fraction of the running time, using a standard multicore computer.

Keywords Particle filtering · Parallelisation · Convergence analysis · Stochastic FitzHugh-Nagumo · Excitable media

1 Introduction

Over the past decade there has been a continued interest in the design of schemes for the implementation of particle filtering algorithms using parallel or distributed hardware of various types, including general purpose devices such as multi-core CPUs or graphical processing units (GPUs) [18], and application-tailored devices such as field programmable gate arrays (FPGAs) [3].

A particle filter is a recursive algorithm for the approximation of the sequence of posterior probability distributions that arise from a stochastic dynamical system in state-space form (see, e.g., [12,28,8,4,2,22] and references therein for a general view of the field). A typical particle filter includes three steps that are repeated sequentially:

- Monte Carlo sampling in the space of the state variables,
- computation of weights for the generated samples and, finally,
- resampling according to the weights.

While at first sight the algorithm may look straightforward to parallelise (sampling and weighting can be carried out concurrently without any constraint), the resampling step involves the interaction of the whole set of Monte Carlo samples. Several authors have proposed schemes for 'splitting' the resampling step into simpler tasks that can be carried out concurrently. The approaches are diverse and range from the heuristic [14,24,19] to the mathematically well-principled [31, 30] (see also [3] together with [26,17]). However, the former are largely based on (often loose) approximations that prevent the claim of any rigorous guarantees of convergence, whereas the latter involve non-negligible overhead to ensure the proper interaction of particles. The double bootstrap filter as described in [30], for example, performs resampling at two levels (involving individual particles and sets of particles, respectively) and only one of the two admits direct parallelisation. The distributed resampling scheme of [3,26,17] is similar, as it resamples small subsets of particles in parallel, but the second level of resampling is substituted by an exchange of particles among different subsets which are typically assigned to different processing elements. The idea of exchanging particles is generalized in the α sequential Monte Carlo (α-SMC) methodology [31], where the resampling step is parameterised by designing a sequence of maps of interactions among particles. The higher-level resampling step of [30], the particle exchange of [3] or the parameterised interaction of [31] imply a computational overhead, i.e., there are extra computations that have to be performed in exchange for parallelising the original resampling task.

In this paper we investigate the use of possibly the simplest scheme for the parallelisation of the standard particle filter, that consists in splitting the computational budget into M fully independent particle filters with N particles each, and then obtaining the desired estimators by averaging over the M independent outcomes of the filters. This approach minimises the parallelisation overhead, since there is no interaction at all among the filters, yet displays desirable theoretical properties. Under mild assumptions, we analyse the mean square error (MSE) of the estimators of 1-dimensional statistics of the optimal filtering distribution and show explicitly the effect of parallelisation scheme on the convergence rate. Specifically, we study the decomposition of the MSE into variance and bias components, to show that the former decays as $\frac{1}{MN}$, i.e., linearly with the total number of particles, while the latter converges towards 0 as $\frac{1}{N^2}$. Independent parallelisation, therefore, has the obvious advantage of reducing the running time while preserving the (asymptotic) performance of a centralised filter. To compare different parallelisation schemes, we introduce a time-error index that brings together the time complexity (asymptotic order of the running time) and the estimation accuracy (asymptotic error rates) into a single quantitative figure of merit that can be used to compare schemes with different degrees of parallelisation. As a side result, we also show that the expected value of the random probability measure output by each independent filter converges in total variation distance to the true posterior with rate of order $1/N$ (and we note that the average measure over the M filters is just a sample-mean estimate of this expected measure).

The analysis of the particle filters in this paper is based on very mild assumptions on the stochastic dynamic model and classic induction arguments. We do not address uniform convergence over time or impose further assumptions on the models because our aim is to illuminate the relationship between the standard (centralised) particle filter and its parallelised versions in the simplest possible framework. A more sophisticated analysis can obviously be carried out by imposing additional assumptions on the dynamic models, yet the underlying argument for the comparison of parallel schemes would be identical. Additional results concerning the conditions that need to be satisfied to attain uniform convergence, the relationship with a α sequential Monte Carlo (α-SMC) of [31] and a central limit theorem can be found in [17]. A study of ensembles of independent estimators in a machine learning framework, which shares the approach in this paper to some extent, is presented in [29]. While we have focused here on particle filters for discrete-time state-space models, the analysis can be similarly done for continuous-time systems and, indeed, the basic results needed for that case (the analysis of the approximation) can be found in [16].

The rest of the paper is organised as follows. In Section 2 we present basic background material and notations to be used through the rest of the paper. The theoretical results are introduced in Section 3, including the analysis of the bias, MSE and total variation distance for various particle approximations. Numerical results for two examples are also presented and discussed, including the tracking of a stochastic Lorenz 63 system, in Section 4, and a stochastic-network model of an excitable medium, in Section 5. The latter is a high-dimensional model (with $\sim$3,000 state variables in our simulations) that consists of a network of modified stochastic FitzHugh-Nagumo nodes, and displays some of the dynamical features observed during atrial fibrillation phenomena in the human heart [20]. Finally, Section 6 contains some concluding remarks.

2 Background

2.1 Notation and preliminaries

We first introduce some common notations to be used through the paper, broadly classified by topics. Below, $\mathbb{R}$ denotes the real line, while for an integer $d \geq 1$, $\mathbb{R}^d = \overbrace{\mathbb{R} \times \ldots \times \mathbb{R}}^{d \text{ times}}$

- Functions.
  - The supremum norm of a real function $f : \mathbb{R}^d \to \mathbb{R}$ is denoted as $\|f\|_\infty = \sup_{x \in \mathbb{R}^d} |f(x)|$.
  - $B(\mathbb{R}^d)$ is the set of bounded real functions over $\mathbb{R}^d$, i.e., $f \in B(\mathbb{R}^d)$ if, and only if, $\|f\|_\infty < \infty$.
- Measures and integrals. Let $S \subseteq \mathbb{R}^d$ be a subset of $\mathbb{R}^d$.
  - $\mathcal{B}(S)$ is the σ-algebra of Borel subsets of S.
  - $\mathcal{P}(S)$ is the set of probability measures over the measurable space $(\mathcal{B}(S), S)$.
  - $(f, \mu) \triangleq \int f(x)\mu(dx)$ is the integral of a real function $f : S \to \mathbb{R}$ with respect to (w.r.t.) a measure $\mu \in \mathcal{P}(S)$.
  - Given a probability measure $\mu \in \mathcal{P}(S)$, a Borel set $A \in \mathcal{B}(S)$ and the indicator function
  $$I_A(x) = \begin{cases} 1, \text{ if } x \in A \\ 0, \text{ otherwise} \end{cases},$$
  $\mu(A) = (I_A, \mu) = \int I_A(x)\mu(dx)$ is the probability of A.
- Sequences, vectors and random variables (r.v.).
  - We use a subscript notation for sequences, namely $x_{t_1:t_2} \triangleq \{x_{t_1}, \ldots, x_{t_2}\}$.
  - For an element $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ of an Euclidean space, its norm is denoted as $\|x\| = \sqrt{x_1^2 + \ldots + x_d^2}$.
  - The L_p norm of a real r.v. Z, with $p \geq 1$, is written as $\|Z\|_p \triangleq E[|Z|^p]^{1/p}$, where $E[\cdot]$ denotes expectation w.r.t. the distribution of Z.

2.2 State-space Markov models in discrete time

Consider two random sequences, $\{X_t\}_{t\geq 0}$ and $\{Y_t\}_{t\geq 1}$, taking values in $\mathcal{X} \subseteq \mathbb{R}^{d_x}$ and $\mathbb{R}^{d_y}$, respectively. Let $\mathbb{P}_t$ be the joint probability measure for the collection of random variables $\{X_0, X_n, Y_n\}_{1\leq n\leq t}$.

We refer to the sequence $\{X_t\}_{t\geq 0}$ as the state (or signal) process and we assume that it is an inhomogeneous Markov chain governed by an initial probability measure $\tau_0 \in \mathcal{P}(\mathcal{X})$ and a sequence of Markov transition kernels $\tau_t : \mathcal{B}(\mathcal{X}) \times \mathcal{X} \to [0, 1]$. To be specific, we define

$$\tau_0(A) \triangleq \mathbb{P}_0 \{X_0 \in A\}, \tag{1}$$
$$\tau_t(A|x_{t-1}) \triangleq \mathbb{P}_t \{X_t \in A | X_{t-1} = x_{t-1}\}, \quad t \geq 1, \tag{2}$$

where $A \in \mathcal{B}(\mathcal{X})$ is a Borel set. The sequence $\{Y_t\}_{t\geq 1}$ is termed the observation process. Each r.v. Y_t is assumed to be conditionally independent of other observations given X_t, namely

$$\begin{aligned} &\mathbb{P}_t \{Y_t \in A | X_{0:t} = x_{0:t}, \{Y_k = y_k\}_{k\neq t}\} \\ &\qquad = \mathbb{P}_t \{Y_t \in A | X_t = x_t\} \end{aligned} \tag{3}$$

for any $A \in \mathcal{B}(\mathbb{R}^{d_y})$, and the conditional distribution of the r.v. Y_t given $X_t = x_t$ is fully described by the probability density function (pdf) $g_t(y_t|x_t) > 0$. We often use g_t as a function of x_t (i.e., as a likelihood) and we emphasise this by writing $g_t^y(x) \triangleq g_t(y|x)$. The prior τ_0, the kernels $\{\tau_t\}_{t\geq1}$, and the functions $\{g_t\}_{t\geq1}$, describe a stochastic Markov state-space model in discrete time.

The stochastic filtering problem consists in the computation of the posterior probability measure of the state X_t given the sequence of observations up to time t. Specifically, for a given observation record $\{y_t\}_{t\geq1}$, we seek the probability measures

$$\pi_t(A) \triangleq \mathbb{P}_t \left\{ X_t \in A | Y_{1:t} = y_{1:t} \right\}, \quad t = 0, 1, 2, ...$$

where $A \in \mathcal{B}(\mathcal{X})$. For many practical problems, the interest actually lies in the computation of statistics of π_t, e.g., the posterior mean or the posterior variance of X_t. Such statistics can be written as integrals of the form (f, π_t), for some function $f : \mathcal{X} \to \mathbb{R}$. Note that, for $t = 0$, we recover the prior signal measure, i.e., $\pi_0 = \tau_0$.

An associated problem is the computation of the one-step-ahead predictive measure

$$\xi_t(A) \triangleq \mathbb{P}_t \left\{ X_t \in A | Y_{1:t-1} = y_{1:t-1} \right\}, \quad t = 1, 2, ...$$

This measure can be explicitly written in terms of the kernel τ_t and the filter π_{t-1}. Indeed, for any integrable function $f : \mathcal{X} \to \mathbb{R}$, we readily obtain (see, e.g., [2, Chapter 10])

$$\begin{aligned} (f, \xi_t) &= \int \int f(x) \tau_t(dx|x') \pi_{t-1}(dx') \\ &= ((f, \tau_t), \pi_{t-1}), \end{aligned} \tag{4}$$

and we write $\xi_t = \tau_t \pi_t$ as shorthand.

The filter at time t, π_t, can be obtained from the predictive measure, ξ_t, and the likelihood, $g_t^{y_t}$, by way of the so-called projective product [2], or Boltzman-Gibbs transformation [8], $\pi_t = g_t^{y_t} \star \xi_t$, defined as

$$(f, g_t^{y_t} \star \xi_t) \triangleq \frac{(f g_t^{y_t}, \xi_t)}{(g_t^{y_t}, \xi_t)}$$

for any integrable function $f : \mathbb{R}^{d_x} \to \mathbb{R}$, which, combined with (4), yields the recursive formula

$$\pi_t = g_t^{y_t} \star \tau_t \pi_{t-1}. \tag{5}$$

It is also useful to keep track of the sequence of non-normalised measures $\{\rho_t\}_{t\geq0}$, where

$$\rho_0 = \pi_0, \quad \rho_t = g_t^{y_t} \cdot \tau_t \rho_{t-1} \tag{6}$$

and, for any integrable function $f : \mathcal{X} \to \mathbb{R}$ and any measure $\alpha \in \mathcal{P}(\mathcal{X})$, we define

$$(f, g_t^{y_t} \cdot \alpha) \triangleq (f g_t^{y_t}, \alpha). \tag{7}$$

We remark that ρ_t is *not* a probability measure, but an unnormalised version of π_t, namely

$$(f, \pi_t) = \frac{(f, \rho_t)}{(\mathbf{1}, \rho_t)},$$

where $\mathbf{1}(x) = 1$ is the constant unit function.

2.3 Standard particle filter

Assume that a sequence of observations $Y_{1:T} = y_{1:T}$, for some $T < \infty$, is given. Then, the sequences of measures $\{\pi_t\}_{t\geq 1}$, $\{\xi_t\}_{t\geq 1}$ and $\{\rho_t\}_{t\geq 0}$ can be numerically approximated using particle filtering. Particle filters are numerical methods based on the recursive relationships (5) and (7). The simplest algorithm, often called 'standard particle filter' or 'bootstrap filter' [15] (see also [11]), can be described as follows.

Algorithm 1 *Bootstrap filter.*

```
1. Initialisation. At time $t = 0$, draw $N$ i.i.d. samples, $x_0^{(i)}$, $i = 1, \ldots, N$, from the distribution $\tau_0$.
2. Recursive step. Let $\{x_{t-1}^{(i)}\}_{1\leq i\leq N}$ be the particles (samples) generated at time $t-1$. At time $t$, proceed with the two steps below.
   (a) For $i = 1, ..., N$, draw a sample $\bar{x}_t^{(i)}$ from the probability distribution $\tau_t(\cdot|x_{t-1}^{(i)})$ and compute the normalised weight
       $$w_t^{(i)} = \frac{g_t^{y_t}(\bar{x}_t^{(i)})}{\sum_{k=1}^N g_t^{y_t}(\bar{x}_t^{(k)})}. \quad (8)$$
   (b) For $i = 1, ..., N$, let $x_t^{(i)} = \bar{x}_t^{(k)}$ with probability $w_t^{(k)}$, $k \in \{1, ..., N\}$.
```

Step 2.(b) is referred to as resampling or selection. In the form stated here, it reduces to the so-called multinomial resampling algorithm [13, 10] but the convergence of the filter can be easily proved for various other schemes (see, e.g., the treatment of the resampling step in [6]).

Using the sets $\{\bar{x}_t^{(i)}\}_{1\leq i\leq N}$ and $\{x_t^{(i)}\}_{1\leq i\leq N}$, we construct random approximations of ξ_t, ρ_t and π_t, namely

$$\xi_t^N = \frac{1}{N}\sum_{i=1}^N \delta_{\bar{x}_t^{(i)}}, \quad \pi_t^N = \frac{1}{N}\sum_{i=1}^N \delta_{x_t^{(i)}}, \quad \text{and} \tag{9}$$

$$\rho_t^N = G_t^N \pi_t^N \tag{10}$$

where δ_x is the delta unit-measure located at $x \in \mathbb{R}^{d_x}$ and[1]

$$G_t^N = \frac{1}{N^t}\prod_{k=1}^t \left(\sum_{j=1}^N g_k^{y_k}(\bar{x}_k^{(j)})\right). \tag{11}$$

[1] Note that G_t^N is an estimate of the normalising constant for π_t (namely, the integral $(\mathbf{1}, \rho_t)$) which can be shown to be unbiased under mild assumptions [8]. In Bayesian model selection this constant is termed "model evidence" while in parameter estimation problems it is often referred to as the likelihood (of the unknown parameters) [1].

For any integrable function f on the state space, it is straightforward to approximate the integrals (f, ξ_t), (f, π_t) and (f, ρ_t) as

$$
\begin{aligned}
(f, \xi_t) &\approx (f, \xi_t^N) = \frac{1}{N} \sum_{i=1}^N f(\bar{x}_t^{(i)}), \\
(f, \pi_t) &\approx (f, \pi_t^N) = \frac{1}{N} \sum_{i=1}^N f(x_t^{(i)}), \quad \text{and} \\
(f, \rho_t) &\approx (f, \rho_t^N) = G_t^N (f, \pi_t^N),
\end{aligned}
$$

respectively.

The convergence of particle filters has been analysed in a number of different ways. Here we use simple results for the convergence of the L_p norms ($p \geq 1$) of the approximation errors. For the approximation of integrals w.r.t. ξ_t and π_t we have the following standard result.

Lemma 1 *Assume that the sequence of observations $Y_{1:T} = y_{1:T}$ is fixed (with $T < \infty$), $g_t^{y_t} \in B(\mathcal{X})$ and $g_t^{y_t} > 0$ (in particular, $(g_t^{y_t}, \xi_t) > 0$) for every $t = 1, 2, ..., T$. Then for any $f \in B(\mathcal{X})$, any $p \geq 1$ and every $t = 1, \ldots, T$,*

$$
\left\| (f, \xi_t^N) - (f, \xi_t) \right\|_p \leq \frac{\bar{c}_t \|f\|_\infty}{\sqrt{N}} \quad \textit{and} \tag{12}
$$

$$
\left\| (f, \pi_t^N) - (f, \pi_t) \right\|_p \leq \frac{c_t \|f\|_\infty}{\sqrt{N}}, \tag{13}
$$

where $\bar{c}_t$ and c_t are finite constants independent of N, $\|f\|_\infty = \sup_{x \in \mathcal{X}} |f(x)| < \infty$ and the expectations are taken over the distributions of the measure-valued random variables ξ_t^N and π_t^N, respectively.

Proof: This result is a special case of, e.g., Lemma 1 in [25]. □

Remark 1 *The constants $\bar{c}_t$ and c_t can be easily shown to increase with t. It is possible to find error rates independent of t by imposing additional assumptions on the state-space model (related to the stability of the optimal filter) [9, 8].*

3 Ensembles of independent particle filters

3.1 Overview

Assume we run M independent bootstrap filters (i.e., M independent instances of Algorithm 1), with N particles each, for the same sequence of observations $\{y_t\}_{0<t\leq T}$. Each filter yields a random approximation $\pi_t^{i,N}$, $i = 1, ..., M$, from which we compute the average $\pi_t^{M\times N} = \frac{1}{M} \sum_{i=1}^M \pi_t^{i,N}$ and adopt the mean square error (MSE) for bounded real test functions, $E\left[\left((f, \pi_t^{M\times N}) - (f, \pi_t) \right)^2 \right]$, $f \in B(\mathcal{X})$, as a performance metric. Since the underlying state-space model is the same

for all filters and they are run in a completely independent manner, the measured-valued random variables $\pi_t^{i,N}$, $i = 1, ..., M$, are i.i.d. and it is straightforward to show (via Lemma 1) that

$$E\left[\left((f, \pi_t^{M\times N}) - (f, \pi_t)\right)^2\right] \leq \frac{c_t^2\|f\|_\infty^2}{MN}, \tag{14}$$

for some constant t independent of N and M. However, the inequality (14) falls short of characterising the performance of the parallelisation scheme because it does not illuminate the effect of the choice of N. In the extreme case of $N = 1$, for example, $\pi_t^{M\times N}$ reduces to the outcome of a sequential importance sampling algorithm, with no resampling, which is known to degenerate quickly in practice. Instead of (14), we seek a bound for the approximation error that provides some indication on the trade-off between the number of independent filters, M, and the number of particles per filter, N.

For this purpose we derive bounds for the approximation error $\left((f, \pi_t^{M\times N}) - (f, \pi_t)\right)^2$ based on the classical decomposition of the MSE in variance and bias terms. First, we obtain preliminary results that are needed for the analysis of the average measure $\pi_t^{M\times N}$. In particular, we prove that the random unnormalised measure ρ_t^N produced by the bootstrap filter (Algorithm 1) is unbiased and attains L_p error rates proportional to $\frac{1}{\sqrt{N}}$, i.e., the same as ξ_t^N and π_t^N. We use these results to derive an upper bound for the bias of π_t^N which is proportional to $\frac{1}{N}$. The latter, in turn, enables us to deduce an upper bound for the MSE of the ensemble approximation $\pi_t^{M\times N}$ consisting of two additive terms that depend explicitly on M and N. Specifically, we show that the variance component of the MSE decays linearly with the total number of particles, $K = MN$, while the bias term decreases quadratically with the number of particles per filter, i.e., with N^2.

All the results to be introduced in the rest of Section 3 hold true under the (mild) assumptions of Lemma 1, which we summarise below for convenience of presentation.

Assumption 1 *The sequence of observations $Y_{1:T} = y_{1:T}$ is arbitrary but fixed, with $T < \infty$.*

Assumption 2 *The likelihood functions are bounded and positive, i.e.,*

$$g_t^{y_t} \in B(\mathcal{X}) \quad \textit{and} \quad g_t^{y_t} > 0 \quad \textit{for every} \quad t = 1, 2, ..., T.$$

Remark 2 *Note that Assumptions 1 and 2 imply that*

- $(g_t^{y_t}, \alpha) > 0$, *for any* $\alpha \in \mathcal{P}(\mathcal{X})$, *and*
- $\prod_{k=1}^T g_t^{y_t} \leq \prod_{k=1}^T \|g_t^{y_t}\|_\infty < \infty$,

for every $t = 1, 2, ..., T$.

Remark 3 *We seek simple convergence results for a fixed time horizon $T < \infty$, similar to Lemma 1. Therefore, no further assumptions related to the stability of the optimal filter for the state-space model [9, 8] are needed. If such assumptions are imposed then stronger (time uniform) asymptotic convergence rates can be found. See [17] for some additional results that apply to the independent filters $\pi_t^{i,N}$ and the ensemble $\pi_t^{M\times N}$.*

3.2 Bias and error rates

Our analysis relies on some properties of the particle approximations of the non-normalised measures ρ_t, $t \geq 1$. We first show that the estimate ρ_t^N in Eq. (9) and (10) is unbiased.

Lemma 2 *If Assumptions 1 and 2 hold, then*

$$E\left[(f, \rho_t^N)\right] = (f, \rho_t)$$

for any $f \in B(\mathcal{X})$ and every $t = 1, 2, ..., T$.

Proof: See Appendix A. □

Combining Lemma 2 with the standard result of Lemma 1 leads to an explicit convergence rate for the L_p norms of the approximation errors $(f, \rho_t^N) - (f, \rho_t)$. This is formally stated below.

Lemma 3 *If Assumptions 1 and 2 hold, then, for any $f \in B(\mathcal{X})$, any $p \geq 1$ and every $t = 1, 2, ..., T$, we have the inequality*

$$\|(f, \rho_t^N) - (f, \rho_t)\|_p \leq \frac{\tilde{c}_t \|f\|_\infty}{\sqrt{N}}, \tag{15}$$

where $\tilde{c}_t < \infty$ is a constant independent of N.

Proof: See Appendix B. □

Finally, Lemmas 2 and 3 together enable the calculation of explicit rates for the bias of the particle approximation of (f, π_t). This is the first contribution of this paper and a key result for the decomposition of the MSE $E\left[\left((f, \pi_t^{M\times N}), (f, \pi_t)\right)^2\right]$ into variance and bias terms. To be specific, we can prove the following theorem.

Theorem 1 *If $0 < (\mathbf{1}, \rho_t) < \infty$ for $t = 1, 2, ..., T$ and Assumptions 1 and 2 hold, then, for any $f \in B(\mathcal{X})$ and every $0 \leq t \leq T$, we obtain*

$$\left| E\left[(f, \pi_t^N) - (f, \pi_t)\right] \right| \leq \frac{\hat{c}_t \|f\|_\infty}{N},$$

where $\hat{c}_t < \infty$ is a constant independent of N.

Proof: Let us first note that $(f, \pi_t) = (f, \rho_t)/(\mathbf{1}, \rho_t)$ and

$$(f, \pi_t^N) = \frac{(f, \rho_t^N)}{G_t^N} \tag{16}$$

$$= \frac{(f, \rho_t^N)}{G_t^N (\mathbf{1}, \pi_t^N)} \tag{17}$$

$$= \frac{(f, \rho_t^N)}{(\mathbf{1}, \rho_t^N)}, \tag{18}$$

where (16) follows from the construction of ρ_t^N, (17) holds because $(\mathbf{1}, \pi_t^N) = 1$ and (18) is, again, a consequence of the definition of ρ_t^N. Therefore, the difference $(f, \pi_t^N) - (f, \pi_t)$ can be written as

$$(f, \pi_t^N) - (f, \pi_t) = \frac{(f, \rho_t^N)}{(\mathbf{1}, \rho_t^N)} - \frac{(f, \rho_t)}{(\mathbf{1}, \rho_t)}$$

and, since $(f, \rho_t) = E[(f, \rho_t^N)]$ (from Lemma 2), the bias can be expressed as

$$E\left[(f, \pi_t^N) - (f, \pi_t)\right] = E\left[\frac{(f, \rho_t^N)}{(\mathbf{1}, \rho_t^N)} - \frac{(f, \rho_t^N)}{(\mathbf{1}, \rho_t)}\right]. \tag{19}$$

Some elementary manipulations on (19) yield the equality

$$E\Big[(f, \pi_t^N) - (f, \pi_t)\Big] = E\left[(f, \pi_t^N)\frac{(\mathbf{1}, \rho_t) - (\mathbf{1}, \rho_t^N)}{(\mathbf{1}, \rho_t)}\right]. \tag{20}$$

If we realise that $E[(\mathbf{1}, \rho_t) - (\mathbf{1}, \rho_t^N)] = 0$ (again, a consequence of Lemma 2) and move the factor $(\mathbf{1}, \rho_t)^{-1}$ out of the expectation, then we easily rewrite Eq. (20) as

$$\begin{aligned}
&E\left[(f, \pi_t^N) - (f, \pi_t)\right] \\
&= \frac{1}{(\mathbf{1}, \rho_t)} E\left[(f, \pi_t^N)\Big((\mathbf{1}, \rho_t) - (\mathbf{1}, \rho_t^N)\Big)\right] \\
&\quad - \frac{(f, \pi_t)}{(\mathbf{1}, \rho_t)} E\left[(\mathbf{1}, \rho_t) - (\mathbf{1}, \rho_t^N)\right] \\
&= \frac{1}{(\mathbf{1}, \rho_t)} E\left[\Big((f, \pi_t^N) - (f, \pi_t)\Big)\Big((\mathbf{1}, \rho_t) - (\mathbf{1}, \rho_t^N)\Big)\right] \\
&\le \frac{1}{(\mathbf{1}, \rho_t)} \sqrt{E\left[\left((f, \pi_t^N) - (f, \pi_t)\right)^2\right]} \\
&\quad \times \sqrt{E\left[\left((\mathbf{1}, \rho_t) - (\mathbf{1}, \rho_t^N)\right)^2\right]} && (21) \\
&\le \frac{1}{(\mathbf{1}, \rho_t)} \left(\frac{c_t\|f\|_\infty}{N} \times \frac{\tilde{c}_t}{N}\right) = \frac{\hat{c}_t\|f\|_\infty}{N}, && (22)
\end{aligned}$$

where we have applied the Cauchy-Schwartz inequality to obtain (21), (22) follows from Lemmas 1 and 3 and the constant

$$\hat{c}_t = \frac{c_t\tilde{c}_t\|f\|_\infty}{(\mathbf{1}, \rho_t)} < \infty$$

is independent of N. □

For any $f \in B(\mathcal{X})$, let $\mathcal{E}_t^N(f)$ denote the approximation difference, i.e.,

$$\mathcal{E}_t^N(f) \triangleq (f, \pi_t^N) - (f, \pi_t).$$

This is a random variable whose second order moment yields the MSE of (f, π_t^N). It is straightforward to obtain a bound for the MSE from Lemma 1 and, by subsequently using Theorem 1, one also readily finds a similar bound for the variance of $\mathcal{E}_t^N(f)$, denoted $\mathsf{Var}[\mathcal{E}_t^N(f)]$. These results are explicitly stated by the corollary below.

Corollary 1 *If $0 < (\mathbf{1}, \rho_t) < \infty$ for $t = 1, 2, ..., T$ and Assumptions 1 and 2 hold, then, for any $f \in B(\mathcal{X})$ and any $0 \le t \le T$, we obtain*

$$E\left[\left(\mathcal{E}_t^N(f)\right)^2\right] \le \frac{c_t^2 \|f\|_\infty^2}{N} \quad \textit{and} \tag{23}$$

$$\mathsf{Var}\left[\mathcal{E}_t^N(f)\right] \le \frac{(c_t^v)^2 \|f\|_\infty^2}{N}, \tag{24}$$

where c_t and c_t^v are finite constants independent of N.

Proof: The inequality (23) for the MSE is a straightforward consequence of Lemma 1. Moreover, we can write the MSE in terms of the variance and the square of the bias, which yields

$$\begin{aligned} E\left[\left(\mathcal{E}_t^N(f)\right)^2\right] &= \mathsf{Var}\left[\mathcal{E}_t^N(f)\right] + E^2\left[\mathcal{E}_t^N\right] \\ &\le \frac{c_t^2 \|f\|_\infty^2}{N}. \end{aligned} \tag{25}$$

Since Theorem 1 ensures that $|E[\mathcal{E}_t^N]| \le \frac{\hat{c}_t \|f\|_\infty}{N}$, then the inequality (25) implies that there exists a constant $c_t^v < \infty$ such that (24) holds. □

3.3 Error rate for the ensemble approximation

Let us run M independent particle filters with the same (fixed) sequence of observations $Y_{1:T} = y_{1:T}$, $T < \infty$, and N particles each. The random measures output by the m-th filter are denoted $\xi_t^{m,N}$, $\pi_t^{m,N}$ and $\rho_t^{m,N}$, with $m = 1, 2, ..., M$. Obviously, all the theoretical properties established in Section 3.2, as well as the basic Lemma 1, hold for each one of the M independent filters.

Definition 1 *The ensemble approximation of π_t with M independent filters is the discrete random measure $\pi_t^{M\times N}$ constructed as the average*

$$\pi_t^{M\times N} = \frac{1}{M}\sum_{m=1}^{M} \pi_t^{m,N}.$$

It is apparent that similar ensemble approximations can be given for ξ_t and ρ_t. Moreover, the statistical independence of the particle filters yields the following corollary as a straightforward consequence of Theorem 1 and Corollary 1.

Corollary 2 *If $0 < (\mathbf{1}, \rho_t) < \infty$ for $t = 1, 2, ..., T$ and Assumptions 1 and 2 hold, then, for any $f \in B(\mathcal{X})$ and any $0 \le t \le T$, the inequality*

$$E\left[\left((f, \pi_t^{M\times N}) - (f, \pi_t)\right)^2\right] \le \frac{(c_t^v)^2 \|f\|_\infty^2}{MN} + \frac{\hat{c}_t^2 \|f\|_\infty^2}{N^2} \tag{26}$$

holds for some constants c_t^v and $\hat{c}_t$ independent of N and M.

Proof: Let us denote

$$\mathcal{E}_t^{M\times N}(f) = (f, \pi_t^{M\times N}) - (f, \pi_t) \quad \text{and}$$
$$\mathcal{E}_t^{m,N}(f) = (f, \pi_t^{m,N}) - (f, \pi_t)$$

for $m = 1, 2, ..., M$. Since $\pi_t^{M\times N}$ is a linear combination of i.i.d. random measures, we easily obtain that

$$\begin{aligned}\left|E\left[\mathcal{E}_t^{M\times N}(f)\right]\right|^2 &= \left|\frac{1}{M}\sum_{m=1}^{M} E\left[\mathcal{E}_t^{m,N}(f)\right]\right|^2 \\ &= \left|E\left[\mathcal{E}_t^{m,N}(f)\right]\right|^2 \\ &\leq \frac{\hat{c}_t\|f\|_\infty}{N}, \quad \text{for any } m \leq M, \end{aligned} \tag{27}$$

where the inequality follows from Theorem 1. Moreover, again because of the independence of the random measures, we readily calculate a bound for the variance of $\mathcal{E}_t^{M\times N}(f)$,

$$\mathsf{Var}\left[\mathcal{E}_t^{M\times N}(f)\right] = \frac{1}{M}\mathsf{Var}\left[\mathcal{E}_t^{m,N}(f)\right] \leq \frac{(c_t^v)^2\|f\|_\infty^2}{MN}, \tag{28}$$

where the inequality follows from Corollary 1. Since $E[(\mathcal{E}_t^{M\times N})^2] = \mathsf{Var}[\mathcal{E}_t^{M\times N}] + \left|E[\mathcal{E}_t^{M\times N}]\right|^2$, combining (28) and (27) yields (26) and concludes the proof. □

The inequality in Corollary 2 is the main theoretical result in this paper and it admits several interpretations. Assume that some "computational budget" is given, i.e., that we have the resources to generate and update over time a fixed number K of particles. If we run Algorithm 1 with K particles, then, according to Lemma 1, the MSE $E[(\mathcal{E}_t^K)^2]$ vanishes asymptotically with rate $\mathcal{O}(1/K)$. The same result is obtained via Corollary 2: a centralised particle filter with K particles is the degenerate case of the ensemble approximation in Definition 1 with $M = 1$ and $N = K$, hence Corollary 2 also states that the MSE decreases with rate $\mathcal{O}(1/K)$. However, Corollary 2 also yields the same MSE rate when a proper ensemble is constructed, with some $M > 1$ and $N = K/M$, which is the property that turns out relevant for parallelisation.

Remark 4 *Some key features of the proposed scheme become apparent from Corollary 2:*

1. *If the total number of particles K is fixed, then the ensemble approximation $\pi_t^{M\times N}$ with $K = MN$ and the standard (bootstrap filter) approximation with K particles, π_t^K, yield the same MSE rate $\mathcal{O}(1/K)$. Since the ensemble approximation is obtained by averaging over M independent particle filters, the interactions among particles are strictly constrained to be local (within the M subsets of size N assigned to the independent particle filters). This implies that if the M particle filters are run in parallel, there is no parallelisation overhead in terms of interaction among the parallel processing units.*
2. *According to the inequality (26), the bias of the estimator $(f, \pi_t^{M\times N})$ is controlled by the number of particles per subset, N, and converges quadratically, while, for fixed N, the variance decays linearly with M. The MSE rate is $\mathcal{O}\left(\frac{1}{MN}\right)$ as long as $N \geq M$. Otherwise, the term $\frac{\hat{c}_t^2\|f\|_\infty^2}{N^2}$ becomes dominant and the resulting asymptotic error bound turns out higher.*

Remark 5 *While the convergence results presented here have been proved for the standard bootstrap filter, it is straightforward to extend them to other classes of particle filters for which Lemmas 1 and 2 hold. This includes most standard algorithms, including the auxiliary particle filter [27] for which numerical results are reported in Section 5.*

3.4 Comparison of parallelisation schemes via time–error indices

The advantage of parallel computation is the drastic reduction of the time needed to run the particle filter. Let the running time for a particle filter with K particles be of order $\mathcal{T}(K)$, where $\mathcal{T} : \mathbb{N} \to (0, \infty)$ is some strictly increasing function of K. The quantity $\mathcal{T}(K)$ includes the time needed to generate new particles, weight them and perform resampling. The latter step is the bottleneck for parallelisation, as it requires the interaction of all K particles. Also, a "straightforward" implementation of the resampling step leads to an execution time $\mathcal{T}(K) = K \log(K)$, although efficient algorithms exist that achieve to a linear time complexity, $\mathcal{T}(K) = K$. We can combine the MSE rate and the time complexity to propose a a time–error performance metric.

Definition 2 *We define the time–error index of a particle filtering algorithm with running time of order $\mathcal{T}$ and asymptotic MSE rate $\mathcal{R}$ as $\mathcal{C} \triangleq \mathcal{T} \times \mathcal{R}$.*

The smaller the index $\mathcal{C}$ for an algorithm, the more (asymptotically) efficient its implementation. For the standard (centralised) particle filter with K particles, the running time is of order $\mathcal{T}(K) = K$ and the MSE rate is of order $\mathcal{R}(K) = K^{-1}$, hence the time–error index becomes

$$\mathcal{C}_{spf}(K) = \mathcal{T}(K) \times \mathcal{R}(K) = 1.$$

For the computation of the ensemble approximation $\pi_t^{M \times N}$ we can run M independent particle filters in parallel, with $N = K/M$ particles each and no interaction among them. Hence, the execution time becomes of order $\mathcal{T}(M, N) = N$. Since the error rate for the ensemble approximation is of order $\mathcal{R}(M, N) = \left(\frac{1}{MN} + \frac{1}{N^2}\right)$, the time–error index of the ensemble approximation is

$$\mathcal{C}_{ipf}(M, N) = \mathcal{T}(M, N) \times \mathcal{R}(M, n) = \frac{1}{M} + \frac{1}{N}$$

and hence it vanishes with $M, N \to \infty$. In particular, since we have to choose $N \geq M$ to ensure a rate of order $\frac{1}{MN}$, then $\lim_{M \to \infty} \mathcal{C}_{ipf} = 0$. In any case, whenever $N > 1$ it is apparent that $\mathcal{C}_{ipf} < \mathcal{C}_{spf}$. Similar comparisons can be carried out for other parallel particle filtering schemes as long as it is possible to identify the effect of the overhead in the running time $\mathcal{T}(M, N)$.

3.5 Expectation of the approximate filter

Besides the computational view of Sections 3.3 and 3.4, Theorem 1 can also be exploited to assess the ensemble approximation $\pi_t^{M \times N}$ in terms of the total variation distance defined as follows.

Definition 3 *Let $\alpha, \beta \in \mathcal{P}(\mathcal{X})$ be two probability measures. The total variation distance (TVD) between α and β is usually defined as*

$$d_{TV}(\alpha, \beta) \triangleq \sup_{A \in \mathcal{B}(\mathcal{X})} |(I_A, \alpha) - (I_A, \beta)|,$$

where I_A is the indicator function

$$I_A(x) = \begin{cases} 1, & \textit{if } x \in A, \\ 0, & \textit{otherwise.} \end{cases}$$

If we regard the measures $\pi_t^{m,N}$ as i.i.d. realisations of the measure-valued r.v. π_t^N (which is the random outcome of a bootstrap filter with N particles), then the ensemble $\pi_t^{M \times N} = \frac{1}{M} \sum_{m=1}^{M} \pi_t^{m,N}$ can be interpreted as a sample mean approximation of the expectation of π_t^N. Indeed, if we introduce

$$\hat{\pi}_t^N \triangleq E[\pi_t^N] = E[\pi_t^{m,N}] \quad \text{for every } m,$$

then it is apparent that, for any $f \in B(\mathcal{X})$,

$$E\left[(f, \pi_t^N) - (f, \pi_t)\right] = (f, \hat{\pi}_t^N) - (f, \pi_t),$$

and $(f, \pi_t^{M \times N}) = \frac{1}{M} \sum_{m=1}^{M} (f, \pi_t^{m,N})$ is the sample-mean approximation of $(f, \hat{\pi}_t^N)$. Theorem 1 can be re-stated immediately as follows.

Corollary 3 *If $0 < (\mathbf{1}, \rho_t) < \infty$ for $t = 1, 2, ..., T$ and Assumptions 1 and 2 hold, then, for every $0 \leq t \leq T$, we obtain that*

$$d_{TV}\left(\hat{\pi}_t^N, \pi_t\right) \leq \frac{\hat{c}_t}{N},$$

where $\hat{c}_t < \infty$ is a constant independent of N.

Proof: From the definition of total variation distance,

$$\begin{aligned} d_{TV}\left(\hat{\pi}_t^N, \pi_t\right) &= \sup_{A \in \mathcal{B}(\mathcal{X})} \left|(I_A, \hat{\pi}_t^N) - (I_A, \pi_t)\right| \\ &\leq \sup_{f \in B(\mathcal{X}) : \|f\|_\infty \leq 1} \left|(f, \hat{\pi}_t^N) - (f, \pi_t)\right|, \end{aligned} \tag{29}$$

since $I_A \in B(\mathcal{X})$ and $\|I_A\|_\infty = 1$ for every Borel set A. However, $(f, \hat{\pi}_t^N) - (f, \pi_t) = E\left[(f, \pi_t^N) - (f, \pi_t)\right]$, hence a straightforward application of Theorem 1 completes the proof. □

4 Example: Stochastic Lorenz 63 model

4.1 The 3-dimensional Lorenz system

Let us illustrate the numerical performance of the proposed independent parallelisation scheme by means of some computer simulations. First, we consider the problem of tracking the state of a 3-dimensional Lorenz system [23] with additive dynamical noise and partial observations [5]. To be specific, consider a 3-dimensional stochastic process $\{X(s)\}_{s\in(0,\infty)}$ (s denotes continuous time) taking values on $\mathbb{R}^3$, whose dynamics is described by the system of stochastic differential equations

$$\begin{aligned} dX_1 &= -\mathsf{s}(X_1 - Y_1) + dW_1, \\ dX_2 &= \mathsf{r}X_1 - X_2 - X_1X_3 + dW_2, \\ dX_3 &= X_1X_2 - \mathsf{b}X_3 + dW_3, \end{aligned}$$

where $\{W_i(s)\}_{s\in(0,\infty)}$, $i = 1, 2, 3$, are independent 1-dimensional Wiener processes and

$$(\mathsf{s}, \mathsf{r}, \mathsf{b}) = \left(10, 28, \frac{8}{3}\right)$$

are static model parameters[2] (which yield chaotic dynamics). A discrete-time version of the latter system using Euler's method with integration step $T_d = 10^{-3}$ is straightforward to obtain and yields the model

$$\begin{aligned} X_{1,n} &= X_{1,n-1} - T_d\mathsf{s}(X_{1,n-1} - X_{2,n-1}) \\ &\quad + \sqrt{T_d}U_{1,n}, \qquad (30) \\ X_{2,n} &= X_{2,n-1} + T_d(\mathsf{r}X_{1,n-1} - X_{2,n-1} - X_{1,n-1}X_{3,n-1}) \\ &\quad + \sqrt{T_d}U_{2,n}, \qquad (31) \\ X_{3,n} &= X_{3,n-1} + T_d(X_{1,n-1}X_{2,n-1} - \mathsf{b}X_{3,n-1}) \\ &\quad + \sqrt{T_d}U_{3,n}, \qquad (32) \end{aligned}$$

where $\{U_{i,n}\}_{n=0,1,\ldots}$, $i = 1, 2, 3$, are independent sequences of i.i.d. normal random variables with 0 mean and variance 1. System (30)-(32) is partially observed every 100 discrete-time steps. Specifically, we collect a sequence of scalar observations $\{Y_t\}_{t=1,2,\ldots}$, of the form

$$Y_t = X_{1,100t} + V_t, \qquad (33)$$

where $\{V_t\}_{t=1,2,\ldots}$ is a sequence of i.i.d. normal random variables with zero mean and variance $\sigma^2 = \frac{1}{2}$.

Let $X_n = (X_{1,n}, X_{2,n}, X_{3,n}) \in \mathbb{R}^3$ be the state vector at discrete time n. The dynamic model given by Eqs. (30)–(32) yields the family of kernels $\tau_{n,\theta}(dx|x_{n-1})$ and the observation model of Eq. (33) yields the likelihood function

$$g_{t,\theta}^{y_t}(x_{100t}) \propto \exp\left\{-\frac{1}{2\sigma^2}\left(y_t - x_{1,100t}\right)^2\right\},$$

[2] Note the difference in notation between the continuous time s and the parameter s.

both in a straightforward manner. The goal is to track the sequence of joint posterior probability measures π_t, $t = 1, 2, ...$, for $\{\hat{X}_t\}_{t=1,...}$, where $\hat{X}_t = X_{100t}$. Note that one can draw a sample $\hat{X}_t = \hat{x}_t$ conditional on $\hat{X}_{t-1} = \hat{x}_{t-1}$ by successively simulating

$$\tilde{x}_n \sim \tau_{n,\theta}(dx|\tilde{x}_{n-1}), \quad n = 100(t-1) + 1, ..., 100t,$$

where $\tilde{x}_{100(t-1)} = \hat{x}_{t-1}$ and $\hat{x}_t = \tilde{x}_{100t}$. The prior measure for the state variables is normal, namely

$$X_0 \sim \mathcal{N}(x_*, v_0^2 \mathcal{I}_3),$$

where $x_* = (-10.2410; -1.3984; -23.6752)$ is the mean[3] and $v_0^2 \mathcal{I}_3$ is the covariance matrix, with $v_0^2 = 10$ and $\mathcal{I}_3$ the 3-dimensional identity matrix.

4.2 Simulation setup

We aim at illustrating the gain in relative performance, taking into account both estimation errors and running time, that can be attained using ensembles of independent particle filters. With this purpose, we have applied

- the standard bootstrap filter (Algorithm 1), termed BF in the sequel,
- the double bootstrap method as described in [30], and
- the ensemble of independent bootstrap filters (BFs) that we have investigated in Section 3

to track the sequence of probability measures π_t generated by the 3-dimensional Lorenz model described in Section 4.1. We have generated a sequence of 200 synthetic observations, $\{y_t; t = 1, ..., 200\}$, spread over an interval of 20 continuous time units, corresponding to 2×10^4 discrete time steps in the Euler scheme (hence, one observation every 100 steps). The same observation sequence has been used for all the simulations.

The ensemble of independent particle filters consists of M filters with N particles each, abiding by the notation in Section 3, with resampling for every $t = 1, 2, ...$, i.e., every time an observation is collected and processed to obtain importance weights. Since the time scale of the Euler approximation of Eqs. (30)–(32) is $n = 100t$, a resampling step is taken every 100 steps of the underlying discrete-time system. Similarly, the double bootstrap algorithm consists of M subsets of particles (*particle islands*, in the terminology of [30]) with N particles per subset. Within each subset, the N particles are resampled for every $t = 1, 2, ...$, while the subsets are resampled for $t = 5k$, $k = 1, 2, ...$ The local resampling (of the particles in the same subset) can be carried out in parallel for the M islands. The standard bootstrap filter runs with K particles, where $K = MN$ (unless explicitly stated) for a fair comparison, and resampling for $t = 1, 2, ...$

We have coded the three algorithms in Matlab (version 7.11.0.584 [R2010b] with the parallel computing toolbox) and run the experiments using a pool of identical multi-processor machines, each one having 8 cores at 3.16 GHz and 32 GB of RAM memory. The standard (centralised) BF is run with $K = NM$ particles in a single core. For the ensemble of independent particle filters we allow the parallel

[3] Chosen from a typical trajectory of the *deterministic* Lorenz 63 model.

computing toolbox to allocate all available cores per server in order to run all BFs concurrently. For the double BF method, we allow to run the separate BFs in parallel up to the subset-level resampling steps.

To assess the approximation errors, we have computed empirical MSEs for the approximation of the posterior mean, $E[\hat{X}_t|Y_{1:t}] = (I, \pi_t)$, where $I(x) = x$ is the identity function, for the three algorithms at the last update step, $t = 200$. Note, however, that the integral (I, π_t) cannot be computed in closed form for this system. Therefore, we have used the "expensive" estimate

$$(I, \pi_t) \approx (I, \pi_t^J), \quad \text{with } J = 10^5 \text{ particles},$$

computed via the standard BF, as a proxy of the true value.

4.3 Numerical results

Figure 1 (left) displays the empirical MSE, averaged over 100 independent simulation runs, attained by the parallel schemes when the number of filters (respectively, particle islands for the double BF method) is fixed, $M = 20$, and the number of particles per filter (particle island) ranges from $N = 100$ to $N = 1000$. The outcome of the centralised BF with $K = MN$ particles, hence ranging from $K = 20 \times 100$ to $K = 20 \times 1000$, is also shown for comparison. We observe that proposed ensemble of independent BFs achieves a poor performance when the number of particles per filter, N, is relatively low ($N = 100$), while for moderate values ($N \geq 400$) it nearly matches the MSE of the centralised BF. The double BF method is more accurate than the independent ensemble for $N = 100, 200$, as it takes advantage of the interaction among the particle islands, but displays a slightly worse MSE than the centralised BF and the independent ensemble for $N \geq 400$.

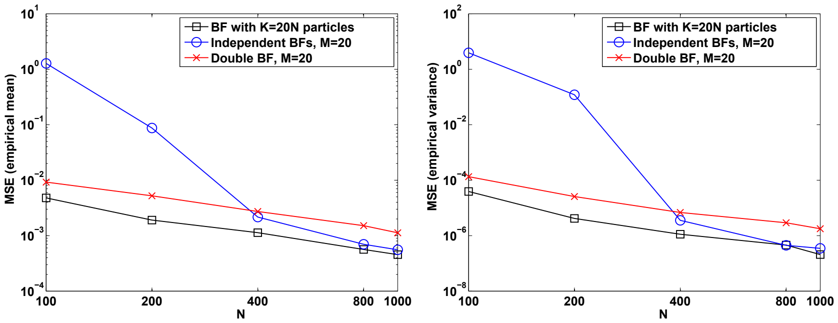

Fig. 1: Empirical mean (left) and variance (right) of the MSE for the centralised BF, the independent ensemble of BFs and the double BF method with $M = 20$ constant and $N = 100, 200, 400, 800, 1000$. All curves have been obtained from a set of 100 independent simulation trials.

The empirical variance of the MSE for the same set of 100 simulation trials is displayed in Figure 1 (right). The results show, again, that the double BF algorithm makes an efficient use of the island resampling step when N is low, so that the M islands remain balanced and the overall filter works properly, but falls short of the independent ensemble scheme for larger values of N.

Finally, we look into the relationship between the MSE and the running time for the three algorithms. With the number of filters (correspondingly, particle islands) $M = 20$ fixed, we have run 100 independent simulation trials for each value $N = 100, 200, 400, 800$ and 1000, and computed the empirical MSE and the average running time for the two parallel schemes and each combination of M and N. Correspondingly, we have also run the centralised BF with $K = MN$ particles, hence for $K = 2 \times 10^3, 4 \times 10^3, 8 \times 10^3, 16 \times 10^3$ and 20×10^3.

Figure 2 (left) displays the resulting empirical MSE versus the running time for the three methods. If we qualify an algorithm as more efficient than another one when it is capable of attaining a lower MSE in the same amount of time, then this set of simulations shows that the independent ensemble scheme is more efficient than both the centralised BF and the double BF method. Indeed, a close look at Figure 2 (left) reveals that the ensemble of $M = 20$ independent BFs with $N = 1000$ particles per filter achieves an empirical MSE of $\approx 6 \times 10^{-4}$ with a running time of ≈ 2.9 seconds, while the centralised BF attains the same performance with $K = 20 \times 800$ particles and a running time of ≈ 27.2 seconds (as shown by the dashed horizontal line in the plot). The double BF method falls short of this MSE value even with $M = 20$ and $N = 1000$. Figure 2 (right) shows the empirical variance of the MSE, versus the running time, for the same set of computer simulations[4].

5 Example: A stochastic dynamical network

5.1 State space model

As a second example we have studied a dynamical complex model consisting of a network of *modified* stochastic FitzHugh-Nagumo (FH-N) nodes. This model is designed to mimic some of the dynamical patterns that can be observed in the electrical activity of the human heart during atrial fibrillation episodes (see [20] for a survey of the topic). Let us consider a 2-dimensional rectangular grid consisting of $J \times J$ nodes, where each node is a stochastic dynamical subsystem that can be described by the classical FitzHugh-Nagumo equations [20] plus

- a stochastic (additive noise) term,
- a random stimulus, and
- a coupling term that determines the interaction with the neighbour nodes.

[4] These results show that a straightforward implementation of the double BF in Matlab, to be run on a multicore server, is not particularly efficient, because of the overhead due to the subset-level resampling. We do not imply that other implementations of this algorithm (e.g., using GPUs, FPGAs or simply a different programming language) should be equally inefficient.

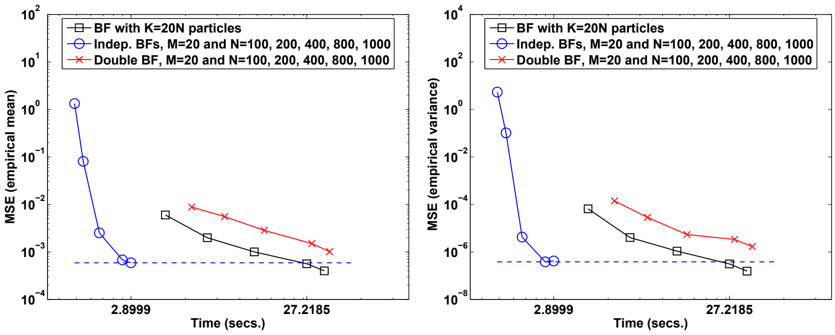

Fig. 2: Empirical mean (left) and variance (right) of the MSE versus the running time for the centralised BF, the independent ensemble of BFs and the double BF method. The two parallel schemes are run with $M = 20$ constant and $N = 100, 200, 400, 600, 800$ and 1000. The centralised BF is run with $K = 20N$ particles, where N takes values in the same way as for the parallel algorithms. The dashed horizontal lines indicate where the mean (left) and variance (right) of the MSE match for the independent ensemble and the centralised BF. The running times for the two algorithms at that MSE level are shown as labels on the horizontal axis.

To be specific, the continuous-time dynamics of the node in the (i, j) position of the grid, with $1 \le i \le J$ and $1 \le j \le J$, follows the stochastic differential equation

$$\begin{aligned} dU_{i,j} = & \left(p_3(U_{i,j}) - V_{i,j} + \frac{1}{D} \sum_{(l,r) \in \mathcal{N}_{i,j}} U_{l,r} \right) dt \\ & + \left(m_{i,j} F + \Psi_{i,j} \right) dt + \tilde{\sigma} dB_{i,j}, \end{aligned} \tag{34}$$

where:

- the continuous-time stochastic process $\{U_{i,j}(s)\}_{s \ge 0}$ usually represents an action potential (or any other voltage-related signal) in biological models [20];
- $p_3(u) = \sum_{r=0}^{3} \alpha_r u^r$ is a polynomial of order 3 with known fixed coefficients α_r, $r = 0, 1, 2, 3$;
- $\{V_{i,j}(s)\}_{s \ge 0}$ is the so-called recovery process, that evolves according to the differential equation

$$dV_{i,j} = \beta_0 U_{i,j} + \beta_1 V_{i,j} + \beta_2 ds \tag{35}$$

  with known parameters β_r, $r = 0, 1, 2$;
- the set $\mathcal{N}_{i,j} \subset \{1, ..., J\} \times \{1, ..., J\}$ contains the neighbours, in the grid, of the (i, j)-th node;
- the coupling coefficient $D > 0$ is known and fixed;
- $F(s) : (0, \infty) \to \mathbb{R}^+$ is a known, typically periodic, forcing signal;
- $m_{i,j} \in \{0, 1\}$ is a (known and fixed) binary indicator that determines which nodes are excited by the forcing signal $F(s)$;

- $\Psi_{i,j}(s)$ is a random stimulus that can be applied to nodes which are returning to a state of quiescence[5], and
- $\{B_{i,j}(s)\}_{s\geq 0}$ are standard and independent Wiener processes and the scale parameter $\tilde{\sigma}$ is assumed known.

For the simulations in this section we have obtained simple discrete-time versions of (34) and (35) using Euler's method with an integration time step T_d, which yields

$$\begin{aligned}
U_{i,j,t} &= U_{i,j,t-1} + T_d\left(p_3(U_{i,j,t-1}) - V_{i,j,t-1}\right) \\
&\quad + T_d\left(\frac{1}{D}\sum_{(l,r)\in\mathcal{N}_{i,j}} U_{l,r,t-1} + m_{i,j}F_t + \Psi_{i,j,t}\right) \\
&\quad + \tilde{\sigma}\sqrt{T_d}B_{i,j,t}, \qquad (36) \\
V_{i,j,t} &= V_{i,j,t-1} + T_d\left(\beta_0 U_{i,j,t-1} + \beta_1 V_{i,j,t-1} + \beta_2\right), \qquad (37)
\end{aligned}$$

where $t = 1, 2, ...$ denotes discrete time, $\{U_{i,j,t}\}_{t=0,1,...}$ is the signal sequence at the (i,j)-th node, $\{V_{i,j,t}\}_{t=0,1,...}$ is the recovery sequence at the same node, $\{\Psi_{i,j,t}\}_{t=1,2,...}$ is a sequence of random stimuli and $\{B_{i,j,t} : 1 \leq i \leq J, 1 \leq j \leq J, t \geq 0\}$ is a set of i.i.d. Gaussian random variables with zero mean and unit variance.

The interest in this model is to study dynamical patterns in the network described by Eqs. (36)-(37), which are caused by the sequence of random stimuli $\mathbf{\Psi}_t = \{\Psi_{i,j,t} : 1 \leq i \leq j, 1 \leq j \leq J\}_{t\geq 0}$. It is known that when a sufficiently strong stimulus (in the form of a positive shift of the voltage signal $U_{i,j,t}$) is applied to the network in a region that had been recently excited by the periodic forcing, the propagation of this stimulus can lead to a re-entrant wavefront that becomes stable and can prevail over the excitations caused by the forcing signal $F(s)$ (see, e.g., [21] for models related to cardiac tissue).

We consider the following model for the sequence $\mathbf{\Psi}_t$. For each $t \geq 1$ and a pair of given thresholds $\mathsf{u}_- < \mathsf{u}^+$, we identify the stimulation region

$$\mathfrak{S}_t := \Big\{(i,j) \in \{1, ..., J\}^2 : \mathsf{u}_- < U_{r,k,t-1} < \mathsf{u}^+ \text{ for some } (r,k) \in \{(i,j)\} \cup \mathcal{N}_{i,j}\Big\}, \qquad (38)$$

i.e., $\mathfrak{S}_t$ consists of the nodes (i,j) such that the state of the node at time $t-1$ lies between the thresholds, $U_{i,j,t-1} \in (\mathsf{u}_-, \mathsf{u}^+)$, *or* the state of some neighbour at time $t-1$, $(r,k) \in \mathcal{N}_{i,j}$, lies between the thresholds, $U_{r,k,t-1} \in (\mathsf{u}_-, \mathsf{u}^+)$. At each time step, a new stimulus is applied to a group of neighbour nodes in the activation region $\mathfrak{S}_t$ with (a small) probability $\epsilon \geq 0$. To be specific, let

$$\mathsf{B}_t^\epsilon \sim \text{Bernoulli}(\epsilon), \quad t = 1, 2, ..., \qquad (39)$$

be a sequence of i.i.d. Bernoulli random variables with parameter $0 \leq \epsilon < 1$ and, for each $(i,j) \in \{1, ..., J\}^2$, let

$$(i_t^*, j_t^*) \sim \mathsf{Uniform}(\mathfrak{S}_t) \qquad (40)$$

[5] The specific random procedure producing this stimulus is described in Section 5.2. The goal is to allow the model to generate re-entries of excitation waves which, in turn, lead to stable and self-sustained dynamical patterns of different classes.

be a single pair of indices drawn from the uniform distribution with support on the stimulus region $\mathfrak{S}_t$. Then, we can generate a random indicator $Q^{\epsilon}_{i,j,t}$ of the form

$$Q^{\epsilon}_{i,j,t} = \begin{cases} 0, \text{ if } (i,j) \notin \{(i^*_t, j^*_t)\} \cup \mathcal{N}_{i^*_t, j^*_t} \\ \mathsf{B}^{\epsilon}_t \text{ if } (i,j) \in \{(i^*_t, j^*_t)\} \cup \mathcal{N}_{i^*_t, j^*_t} \end{cases} \tag{41}$$

which selects a set of neighbouring nodes where a new stimulus is to be applied at time t. Then, the sequence of stimuli in the (i,j) node can be written as

$$\Psi_{i,j,t} = \tilde{F} \max \left\{ 1, \sum_{l=0}^{\ell_0 - 1} Q^{\epsilon}_{i,j,t-l} \right\}, \tag{42}$$

where $\tilde{F}$ is the amplitude of each single stimulus, which is sustained during ℓ_0 consecutive discrete-time steps.

Let us denote

$$X_{i,j,t} = (U_{i,j,t}, V_{i,j,t}, Q_{i,j,t-\ell_0+1:t}) \in \mathbb{R}^{2+\ell_0} \quad \text{and}$$
$$X_t = (X_{1,1,t}, \dots, X_{1,J,t}, \dots, X_{J,J,t}) \in \mathbb{R}^{(2+\ell_0)J^2}.$$

The $(2+\ell_0)J^2$-dimensional sequence $\{X_t\}_{t \geq 0}$ is a Markov process in discrete time. We do not attempt to write down the associated transition kernel $\tau_t(dx|x_{t-1})$ explicitly, yet it is straightforward to generate a sample X_t conditional on $X_{t-1} = x_{t-1}$ using Eqs. (38)–(42), (36) and (37), in this particular order, over the set of indices $\{(i,j) : 1 \leq i \leq J, 1 \leq j \leq J\}$.

To complete the state-space model description, we assume the ability to observe the signal (voltage) variables, $U_{i,j,t}$, in a subset of the nodes of the grid. To be specific, at time t we collect the measurements

$$Y_{i,j,t} = U_{i,j,t} + \bar{\sigma} \bar{B}_{i,j,t}, \quad (i,j) \in \mathcal{S}_y, \tag{43}$$

where $\mathcal{S}_y \subset \{1, ..., J\} \times \{1, ..., J\}$ is the set of observed nodes, $\{\bar{B}_{i,j,t} : (i,j) \in \mathcal{S}_y\}$ is a set of i.i.d. standard Gaussian random variables (centred and with unit variance) and $\bar{\sigma}^2$ is a known scale parameter. The full observation at time t is denoted

$$Y_t = \{Y_{i,j,t} : (i,j) \in \mathcal{S}_y\} \in \mathbb{R}^{|\mathcal{S}_y|}.$$

The likelihood function is Gaussian, namely

$$g^{y_t}_t(x_t) \propto \exp \left\{ -\frac{1}{2\bar{\sigma}^2} \sum_{(i,j) \in \mathcal{S}_y} (y_{i,j,t} - u_{i,j,t})^2 \right\}.$$

Equations (38)–(42), (36), (37) and (43) describe a Markov state–space model in discrete time, with conditionally independent observations. We aim at tracking the sequence of probability measures

$$\pi_t(A) = \mathbb{P}\{X_t \in A | Y_{1:t} = y_{1:t}\}, \quad t = 1, 2, ..., T,$$

where $A \in \mathcal{B}(\mathcal{X})$ and $y_{1:T}$ is a given sequence of observations.

5.2 Simulation setup

We have run simulations for a network of $J^2 = 32^2 = 1,024$ modified stochastic FH-N nodes, interconnected in a regular square grid. Therefore, for an "inner" node $(i, j) \in \{1, ..., J\}^2$ with $1 < i, j < J$ the set of neighbours is

$$\mathcal{N}_{i,j} = \left\{(r, l) \in \{1, ..., J\}^2 : r = i \pm 1, l = j \pm 1\right\},$$

whereas for the nodes in the "corners" of the grid the sets of neighbours are

$$\begin{aligned}
\mathcal{N}_{1,1} &= \{(1, 2), (2, 1)\}, \\
\mathcal{N}_{1,J} &= \{(1, J-1), (2, J)\}, \\
\mathcal{N}_{J,1} &= \{(J, 2), (J-1, 1)\}, \\
\mathcal{N}_{J,J} &= \{(J-1, J), (J, J-1)\},
\end{aligned}$$

and for the nodes on the "sides" of the grid

$$\begin{aligned}
\mathcal{N}_{1,j} &= \{(1, j \pm 1), (2, j)\}, \\
\mathcal{N}_{j,1} &= \{(j \pm 1, 1), (j, 2)\}, \\
\mathcal{N}_{j,J} &= \{(j \pm 1, J), (j, J-1)\}, \\
\mathcal{N}_{J,j} &= \{(J, j \pm 1), (J-1, j)\},
\end{aligned}$$

where $1 < j < J$ in all cases. The time discretisation period is $T_d = 5 \times 10^{-3}$ continuous-time units and the coupling constant, that sets the "strength" of the links between neighbours, is $\frac{1}{D} = 4.5 \times 10^{-3}$.

The dynamics of the FH-N system is highly dependent on the choice of the polynomial $p_3(u)$ in Eq. (36), which for this set of simulations is selected as

$$p_3(u) = u\left(u + \sqrt{\frac{18}{5}}\right)\left(u - \sqrt{\frac{18}{5}}\right),$$

and the forcing signal $F(s)$, which hereafter consists of a periodic sequence of pulses of the form

$$F(s) = \sum_{k=0}^{\infty} \sqcap(s - kS_\sqcap),$$

where $\sqcap(s)$ is the square waveform

$$\sqcap(s) = \begin{cases} \tilde{F} \text{ if } 0 \leq s \leq S_\sqcap \\ 0 \text{ otherwise} \end{cases},$$

the period of $F(s)$ is $S_\sqcap = 20$ time units and the amplitude of the pulses is $\tilde{F} = 200$. The discrete-time forcing signal is $F_t = F(s = tT_d)$.

To construct the stimulus region $\mathfrak{S}_t$ given by Eq. (38) we use the thresholds

$$\mathsf{u}_- = -1.8 \quad \text{and} \quad \mathsf{u}^+ = -1.6,$$

which correspond to the back tail of a wave propagating over the network. The parameter of the Bernoulli distribution in (39) is $\epsilon = 10^{-3}$, the amplitude of the stimulus in (42) is $\tilde{F} = 200$ and it is sustained for $\ell_0 = 25$ discrete time steps.

The variance of the noise term in Eq. (36) is $\tilde{\sigma}^2 = \frac{1}{2}$ and the specification of the dynamics is complete with the parameters $(\beta_0, \beta_1, \beta_2) = (2.1, -0.6, 0.6)$ in Eq. (37).

The observations are collected at a grid of 5×5 equally-spaced zones, each zone consisting of four nodes forming a 2×2 square, as shown in Figure 3. Therefore, we collect observations from $5 \times 5 \times 2 \times 2 = 100$ nodes out of $J^2 = 1,024$ in the network. For each observed node, say in the position (r, l), we obtain the measurement specified by Eq. (43), where the noise variance is $\bar{\sigma}^2 = \frac{1}{2}$.

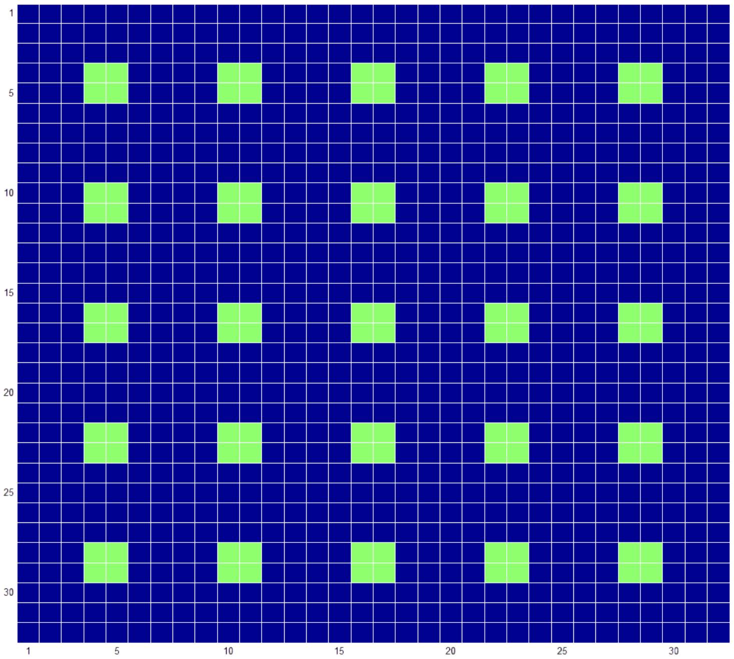

Fig. 3: Location of the observations within the 32×32 grid of stochastic FH-N nodes with random stimuli. The blue squares represent the nodes in the network, each one following the dynamics specified by Eqs. (38)–(42), (36) and (37). Observed nodes are displayed in green. Each observation zone includes four nodes, forming a 2×2 square. The set $\mathcal{S}_y$ in (43) contains exactly the green-coloured nodes in this figure.

5.3 Numerical results

We have run computer simulations for this model using Matlab version R2012b (32 bits), with the parallel computing toolbox enabled, on an a multicore Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz server. All the results reported here are based on a set of 20 independent forward simulations of the stochastic FH-N network model described in Sections 5.1 and 5.2. The state trajectories for these simulations have been recorded and synthetic observations have been generated from them according to (43).

Figure 4 (top) displays the actual variables $U_{i,j,t}$ for $i, j = 1, ..., 32$ (i.e., the complete network) at different time steps for one of the simulations. This is a typical realisation of the process U_t, with a wave propagating over the network and occasional stimuli appearing behind the wave. The random stimulus in frame 9 is strong enough to start a wavefront that propagates over the network and moves the system away from the periodic behaviour that would be induced by the forcing signal F_t alone.

In order to track U_t (or, indeed, the whole state X_t) we have applied an ensemble of independent auxiliary particle filters [27], which are labeled AIPF in Figure 4 (bottom). In particular, we have run $M = 10$ independent filters with $N = 5,000$ particles each and the figure shows how the algorithm successfully detects the first stimulus, follows the resulting wavefront and, later, detects the various random stimuli (in frames 7, 8, 9 and 13) and also tracks the wave started by the stimulus in frame 9.

Figure 5 (left) shows the empirical MSE per node of the state estimates generated by different ensembles of auxiliary particle filters, namely $M = 10$ independent filters with the number of particles per filter varying from $N = 100$ to $N = 5,000$. The same MSE for the centralised auxiliary particle filters with $K = MN$ particles (i.e., with $K = 1,000, \ldots, 50,000$) is also plotted for reference. The same as for the Lorenz 63 example, we observe that the relative performance of the ensemble of independent filters is poor for smaller values of N, yet for $N \geq 2,000$ it nearly matches the MSE of the centralised particle filter. The results have been averaged over the 20 independent data sets described at the beginning of this Section. The empirical variance of the MSE per node for the same set of simulations is depicted in Figure 5 (right).

Finally, Figure 6 (left) displays the empirical MSE (per node) versus the running time (per discrete time step) for an ensemble of $M = 10$ independent auxiliary particle filters and for the centralised auxiliary particle filter with $K = MN$ particles. For the ensemble of independent filters, each point in the curve corresponds to a different value of N (particles per filter), and for the centralised filter, each point corresponds to a different value of $K = 10N$, the total number of particles. In particular, we have obtained results for $N = 100, 500, 1000, 2000$ and 5000. A close look at the figure shows that the ensemble of $M = 10$ independent filters with $N = 5,000$ particles each achieves an empirical MSE of ≈ 0.1813 with a running time of ≈ 12.56 seconds, while the centralised particle filter attains the same MSE with $K = 10 \times 2,000$ particles and a running time of ≈ 28.67 seconds (no improvement is observed for $K = 10 \times 5,000$ particles, while the running time scales up to ≈ 73 seconds). Therefore, the ensemble of independent filters turns out more efficient than the centralised algorithm for this example as well.

6 Discussion

We have addressed the problem of parallelising the standard particle filtering algorithm[6] by splitting the total number of particles K into M subsets, running one

[6] While we have restricted the research to the standard (bootstrap) filter for simplicity of presentation, the analysis, and the whole argument about parallelisation by means of

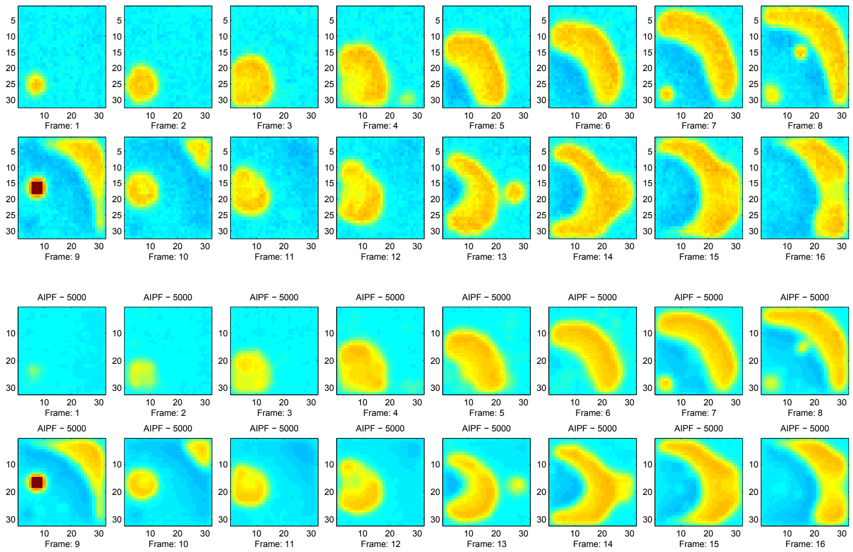

Fig. 4: **Top:** Example of a wave propagating over the 32x32 grid, where hotter colours describe higher action potential $U_{i,j,t}$ values while colder ones represent nodes at their rest state. A stimulus is applied in the first frame and then propagates through the network creating a wave. Random stimuli behind this wave appear in frames 7, 8, 9 and 13. The stimulus that appears in frame 9 is strong enough to initiate another wavefront that propagates from that location, hence disorganising the periodic behaviour induced by the forcing signal F_t. **Bottom:** Posterior-mean estimation of the process U_t using an ensemble of $M = 10$ independent auxiliary particle filters (labeled AIPF) with $N = 5,000$ particles each. The filter detects the initial stimulus and tracks the resulting wave. It also "catches" the random stimuli in frames 7, 8, 9 and 13, and tracks the wavefront initiated by the stimulus in frame 9.

independent particle filter per subset, and then building an average filter measure using the ensemble of i.i.d. random approximations produced by the filters. This approach avoids all dependences among the filters, an idea which goes against recent approaches to the problem [3,31,30,26,17] and, to some extent, against the intuition that a certain interaction is needed to make the M filters, with N particle each, work together with the same performance as a centralised filter with $K = MN$ particles.

The rationale to advocate the averaging of independent particle filters instead of the (algorithmically more sophisticated) schemes based on controlled interactions, such as [3,31,30], relies on both theoretical and practical arguments. Theoretically, the simple analysis in this paper shows that the parallelisation based on independent filters has the same asymptotic performance as a centralised

independent filters, extends in an almost straightforward manner to more sophisticated algorithms that may use tailored importance functions and/or auxiliary variables for the generation of particles.

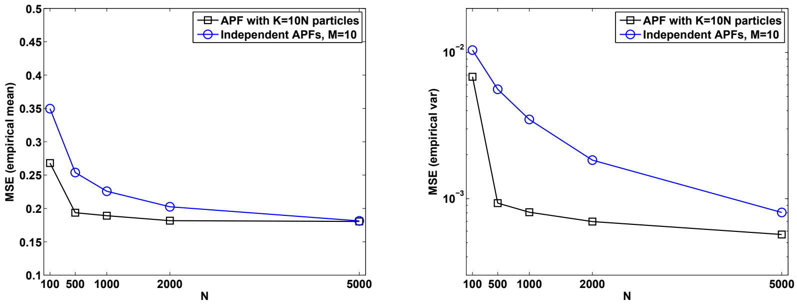

Fig. 5: **Left:** Empirical MSE (per node) versus the number of particles per independent filter, N, averaged over 20 independent simulation runs. The results correspond to an ensemble of $M = 10$ independent auxiliary particle filters. **Right:** Empirical variance of the MSE (per node) versus the number of particles per independent filter, N. The performance of the centralised auxiliary particle filter with $K = MN$ particles is also plotted, for reference, in the two plots.

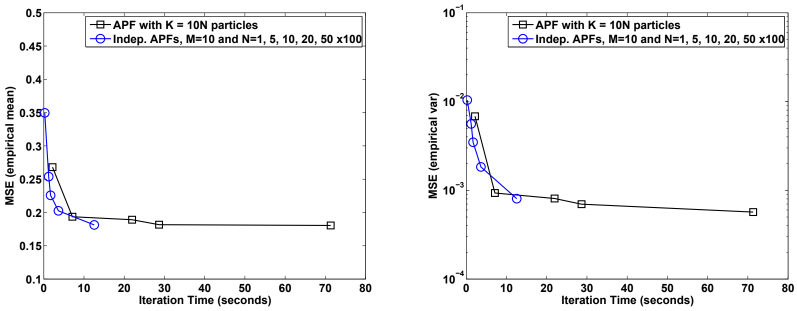

Fig. 6: **Left:** Empirical MSE (per node) versus running time (per discrete time step), averaged over 20 independent simulation runs. Each point in the curves corresponds to a different value of N, namely $N = 100, 500, 1000, 2000$ and 5000. The centralised auxiliary particle filter is run with $K = 10 \times N$ particles, for the same values of N. **Right:** Empirical variance of the MSE (per node) versus the running time (per discrete time step) for the same set of simulations.

particle filter. We have obtained this result by looking at the mean square approximation error for integrals of bounded functions w.r.t. the filter measure, decomposed into variance and bias terms. The bounds we have obtained depend explicitly on M (the number of filters) and N (the number of particles per filter), and show that there is no asymptotic performance loss for schemes with $N \geq M$. This result is actually aligned with recent contributions in the field of machine learning regarding the statistical properties of averages of independent estimators (including classifiers and regressors) [29]. We have also utilized the asymptotic convergence rates to propose a time–error index that enables a quantitative comparison of centralised and parallel particle filtering scheme in terms of asymptotic accuracy and running time. These analytical results can be extended to account for stronger forms of convergence (under additional assumptions on the model, see [17]) and adapted to continuous-time state-space systems (see [16] for the analysis of the bias and the MSE in this context).

From a practical perspective, we have shown that the averaging of independent filters should be preferred when N, the number of particles per independent filter, is sufficiently large. Indeed, our computer simulations suggest that if we seek a scheme with a large number of parallel filters (M) and a relatively small number of particles per filter (N) then parallelisation schemes that exploit a certain degree of interaction between filters should be preferred –at the expense of a computational overhead to implement such interaction. On the other hand, if N is large enough to make the parallel filters work (even roughly), then our simulations show that interaction is not needed anymore and independence can be fully exploited both in terms of accuracy and running time.

The interest in designing particle filtering schemes that can have fast implementations using massively parallel hardware has followed the surge of several problems in science (geophysics, biochemistry or systems medicine) and engineering (sensor networks, multi target tracking) where the fundamental task is the tracking of a complex, high-dimensional dynamical system. In this paper we have investigated an example where the system of interest is a network of 1,024 stochastic FitzHugh-Nagumo nodes with random stimuli, interconnected through a 2-dimensional regular grid. This kind of complex stochastic network is a good representative of models commonly used for excitable media in ecology or medicine [20] and illustrates the kind of models for which parallel particle filters are needed.

Acknowledgements The work of J. M. and G. R. was partially supported by *Ministerio de Economía y Competitividad* of Spain (project TEC2012-38883-C02-01 COMPREHENSION) and the Office of Naval Research Global (award no. N62909- 15-1-2011. D. C. and J. M. would also like to acknowledge the support of the Isaac Newton Institute through the program "Monte Carlo Inference for High-Dimensional Statistical Models". The authors would like to thank Dr. Katrin Achutegui for her valuable assistance in obtaining the numerical results displayed in Section 4.

A Proof of Lemma 2

We proceed by induction in the time index t. For $t = 0$, $\rho_0 = \tau_0 = \pi_0$ and, since $x_0^{(i)}$, $i = 1, ..., N$, are drawn from π_0, the equality $E[(f, \rho_0^N)] = (f, \rho_0)$ is straightforward.

Let us assume that

$$E\left[(f, \rho_{t-1}^N)\right] = (f, \rho_{t-1}) \tag{44}$$

for some $t > 0$ and any $f \in B(\mathcal{X})$. If we use $\bar{\mathcal{F}}_t$ to denote the σ-algebra generated by the set of random variables $\{x_{0:t-1}^{(i)}, \bar{x}_{1:t}^{(i)} : 1 \leq i \leq N\}$ then we readily find that

$$E\left[(f, \rho_t^N)|\bar{\mathcal{F}}_t\right] = E\left[G_t^N(f, \pi_t^N)|\bar{\mathcal{F}}_t\right] = G_t^N(f, \bar{\pi}_t^N), \tag{45}$$

since G_t^N is measurable w.r.t. $\bar{\mathcal{F}}_t$ and $E[(f, \pi_t^N)|\bar{\mathcal{F}}_t] = (f, \bar{\pi}_t^N)$. Moreover, if we recall that

$$(f, \bar{\pi}_t^N) = \sum_{i=1}^N w_t^{(i)} f(\bar{x}_t^{(i)}) = \sum_{i=1}^N \frac{g_t^{y_t}(\bar{x}_t^{(i)}) f(\bar{x}_t^{(i)})}{\sum_{j=1}^N g_t^{y_t}(\bar{x}_t^{(j)})} = \frac{(f g_t^{y_t}, \xi_t^N)}{(g_t^{y_t}, \xi_t^N)}$$

then it is apparent from the definition of G_t^N in (11) that

$$G_t^N(f, \bar{\pi}_t^N) = G_{t-1}^N(f g_t^{y_t}, \xi_t^N). \tag{46}$$

Taking together (45) and (46) we have

$$E\left[(f,\rho_t^N)|\bar{\mathcal{F}}_t\right] = G_{t-1}^N(fg_t^{y_t},\xi_t^N). \tag{47}$$

Let $\mathcal{F}_{t-1}$ be the σ-algebra generated by the set of variables $\{x_{0:t-1}^{(i)}, \bar{x}_{0:t-1}^{(i)} : 1 \leq i \leq N\}$. Since $\mathcal{F}_{t-1} \subseteq \bar{\mathcal{F}}_t$, Eq. (47) yields

$$\begin{aligned} E\left[(f,\rho_t^N)|\mathcal{F}_{t-1}\right] &= E\left[G_{t-1}^N(fg_t^{y_t},\xi_t^N)|\mathcal{F}_{t-1}\right] \\ &= G_{t-1}^N E\left[(fg_t^{y_t},\xi_t^N)|\mathcal{F}_{t-1}\right], \end{aligned} \tag{48}$$

since G_{t-1}^N is measurable w.r.t. $\mathcal{F}_{t-1}$. Moreover, for any $h \in B(\mathcal{X})$, it is straightforward to show that

$$E[(h,\xi_t^N)|\mathcal{F}_{t-1}] = (h,\tau_t\pi_{t-1}^N) = \left((h,\tau_t),\pi_{t-1}^N\right),$$

hence, as $fg_t^{y_t} \in B(\mathcal{X})$, we readily obtain

$$E\left[(fg_t^{y_t},\xi_t^N)|\mathcal{F}_{t-1}\right] = \left((fg_t^{y_t},\tau_t),\pi_{t-1}^N\right). \tag{49}$$

Substituting (49) into (48) we arrive at

$$\begin{aligned} E\left[(f,\rho_t^N)|\mathcal{F}_{t-1}\right] &= G_{t-1}^N\left((fg_t^{y_t},\tau_t),\pi_{t-1}^N\right) \\ &= \left((fg_t^{y_t},\tau_t),\rho_{t-1}^N\right), \end{aligned} \tag{50}$$

where (50) follows from the definition of the estimate of ρ_{t-1}, namely $\rho_{t-1}^N = G_{t-1}^N\pi_{t-1}^N$. If we take unconditional expectations on both sides of Eq. (50), we obtain

$$\begin{aligned} E\left[(f,\rho_t^N)\right] &= E\left[\left((fg_t^{y_t},\tau_t),\rho_{t-1}^N\right)\right] \\ &= \left((fg_t^{y_t},\tau_t),\rho_{t-1}\right) &(51) \\ &= (f,g_t^{y_t}\cdot\tau_t\rho_{t-1}) &(52) \\ &= (f,\rho_t), &(53) \end{aligned}$$

where equality (51) follows from the induction hypothesis (44), (52) is obtained by simply re-ordering (51) and Eq. (53) follows from the recursive definition of ρ_t in (6).

B Proof of Lemma 3

For $t = 0$, $\rho_0^N = \pi_0^N$, hence the result follows from Lemma 1. At any time $t > 0$, since $\rho_t^N = G_t^N\pi_t^N$, we readily have

$$\begin{aligned} E\left[\left|(f,\rho_t^N)-(f,\rho_t)\right|^p\right] &= E\left[\left|\frac{1}{N}\sum_{i=1}^N G_t^N f(x_t^{(i)}) - (f,\rho_t)\right|^p\right] \\ &= E\left[\left|\frac{1}{N}\sum_{i=1}^N Z_t^{(i)}\right|^p\right], \end{aligned} \tag{54}$$

where $Z_t^{(i)} = G_t^N f(x_t^{(i)}) - (f,\rho_t)$, $i = 1,...,N$. It is apparent that the random variables $Z_t^{(i)}$, $i = 1,...,N$, are conditionally independent given the σ-algebra $\bar{\mathcal{F}}_t$ generated by the set $\{x_{0:t-1}^{(j)}, \bar{x}_{0:t}^{(j)} : 1 \leq j \leq N\}$. It can also be proved that every $Z_t^{(i)}$ is centred and bounded, as explicitly shown in the sequel.

To see that $Z_t^{(i)}$ has zero mean, let us note first that

$$E\left[G_t^N f(x_t^{(i)})|\bar{\mathcal{F}}_t\right] = G_t^N(f,\bar{\pi}_t^N),$$

since G_t^N is measurable w.r.t. $\bar{\mathcal{F}}_t$. Moreover, by the same argument as in the proof of Lemma 2, one can show that $G_t^N(f, \bar{\pi}_t^N) = G_{t-1}^N(fg_t^{y_t}, \xi_t^N)$ and, therefore,

$$\begin{aligned} E\left[G_t^N f(x_t^{(i)})|\mathcal{F}_{t-1}\right] &= E\left[G_{t-1}^N(fg_t^{y_t}, \xi_t^N)|\mathcal{F}_{t-1}\right] \\ &= G_{t-1}^N\left((fg_t^{y_t}, \tau_t), \pi_{t-1}^N\right), \end{aligned} \tag{55}$$

where we have used the fact that, for any $h \in B(\mathcal{X})$, $E[(h, \xi_t^N)|\mathcal{F}_{t-1}] = ((h, \tau_t), \pi_{t-1}^N)$. However, since $\rho_{t-1}^N = G_{t-1}^N \pi_{t-1}^N$, Eq. (55) amounts to

$$E\left[G_t^N f(x_t^{(i)})|\mathcal{F}_{t-1}\right] = \left((fg_t^{y_t}, \tau_t), \rho_{t-1}^N\right)$$

and taking (unconditional) expectations on both sides of the equation above yields

$$\begin{aligned} E\left[G_t^N f(x_t^{(i)})\right] &= E\left[\left((fg_t^{y_t}, \tau_t), \rho_{t-1}^N\right)\right] \\ &= \left((fg_t^{y_t}, \tau_t), \rho_{t-1}\right) && (56) \\ &= (f, \rho_t), && (57) \end{aligned}$$

where (56) follows from Lemma 2 (i.e., ρ_{t-1}^N is unbiased) and (57) is a straightforward consequence of the definition of ρ_t in (6). Equation (57) states that $E[Z_t^{(i)}] = E[G_t^N f(x_t^{(i)}) - (f, \rho_t)] = 0$.

To see that (every) $Z_t^{(i)}$ is bounded, note that, for any finite t,

$$G_t^N \le \prod_{k=1}^t \|g_k^{y_k}\|_\infty < \infty, \tag{58}$$

whereas

$$\begin{aligned} (f, \rho_t) &= ((fg_t^{y_t}, \tau_t), \rho_{t-1}) \\ &= ((((fg_t^{y_t}, \tau_t)g_{t-1}^{y_{t-1}}, \tau_{t-1})g_{t-2}^{y_{t-2}}, ..., \tau_1), \pi_0) \\ &\le \|f\|_\infty \prod_{k=1}^t \|g_k^{y_k}\|_\infty < \infty. \end{aligned} \tag{59}$$

Taking (58) and (59) together we arrive at

$$|Z_t^{(i)}| \le 2\|f\|_\infty \prod_{k=1}^t \|g_k^{y_k}\|_\infty \tag{60}$$

which is finite for any finite t (indeed, for every $t \le T$).

Since the variables $Z_t^{(i)}$, $i = 1, ..., N$, in (54) are bounded, with zero mean and conditionally independent given $\bar{\mathcal{F}}_t$, it is not difficult to show (see, e.g., [7, Lemma A.1]) that

$$E\left[\left|(f, \rho_t^N) - (f, \rho_t)\right|^p\right] \le \frac{2^p \check{c}_t^p \|f\|_\infty^p \prod_{k=1}^t \|g_k^{y_k}\|_\infty^p}{N^{\frac{p}{2}}}, \tag{61}$$

where the constant $\check{c}_t$ is finite and independent of N. From (61) we easily obtain the inequality (15) in the statement of Lemma 3, with $\tilde{c}_t = 2\check{c}_t\|f\|_\infty \prod_{k=1}^t \|g_k^{y_k}\|_\infty < \infty$ for any $t \le T < \infty$.

References

1. C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society B*, 72:269–342, 2010.
2. A. Bain and D. Crisan. *Fundamentals of Stochastic Filtering.* Springer, 2008.

3. M. Bolić, P. M. Djurić, and S. Hong. Resampling algorithms and architectures for distributed particle filters. *IEEE Transactions Signal Processing*, 53(7):2442–2450, July 2005.
4. O. Cappé, S. J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.
5. A. J. Chorin and P. Krause. Dimensional reduction for a Bayesian filter. *PNAS*, 101(42):15013–15017, October 2004.
6. D. Crisan. Particle filters - a theoretical perspective. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, chapter 2, pages 17–42. Springer, 2001.
7. D. Crisan and J. Miguez. Particle-kernel estimation of the filter density in state-space models. *Bernoulli*, (in press) (arXiv:1111.5866v5 [stat.CO]), 2013.
8. P. Del Moral. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer, 2004.
9. P. Del Moral and A. Guionnet. On the stability of interacting processes with applications to filtering and genetic algorithms. *Annales de l'Institut Henri Poincaré (B) Probability and Statistics*, 37(2):155–194, 2001.
10. R. Douc, O. Cappé, and E. Moulines. Comparison of resampling schemes for particle filtering. In *Proceedings of the 4^{th} International Symposium on Image and Signal Processing and Analysis*, pages 64–69, September 2005.
11. A. Doucet, N. de Freitas, and N. Gordon. An introduction to sequential Monte Carlo methods. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, chapter 1, pages 4–14. Springer, 2001.
12. A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, New York (USA), 2001.
13. A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo Sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.
14. A. Gelencsér-Horváth, G. Tornai, A. Horváth, and G. Cserey. Fast, parallel implementation of particle filtering on the gpu architecture. *EURASIP Journal on Advances in Signal Processing*, 2013(1):1–16, 2013.
15. N. Gordon, D. Salmond, and A. F. M. Smith. Novel approach to nonlinear and non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, 1993.
16. W. Han. *On the Numerical Solution of the Filtering Problem*. Ph.D. Thesis. Department of Mathematics, Imperial College London, 2013.
17. K. Heine and N. Whiteley. Fluctuations, stability and instability of a distributed particle filter with local exchange. *arXiv*, 1505.02390v1 [stat.ME], 2015.
18. G. Hendeby, R. Karlsson, and F. Gustafsson. Particle filtering: the need for speed. *EURASIP Journal on Advances in Signal processing*, 2010:22, 2010.
19. O. Hlinka, O. Sluciak, F. Hlawatsch, P. Djuric, and M. Rupp. Likelihood consensus and its application to distributed particle filtering. *IEEE Transactions on Signal Processing*, 60(8):4334–4349, 2012.
20. J. P. Keener and J. Sneyd. *Mathematical Physiology: I: Cellular Physiology*, volume 1. Springer Verlag, 2008.
21. J. P. Keener and J. Sneyd. *Mathematical Physiology: II: Systems Physiology*, volume 1. Springer Verlag, 2008.
22. H. R. Künsch. Particle filters. *Bernoulli*, 19(4):1391–1403, 2013.
23. E. N. Lorenz. Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20(2):130–141, 1963.
24. J. Míguez. Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization. *Digital Signal Processing*, 17:787–807, 2007.
25. J. Míguez, D. Crisan, and P. M. Djurić. On the convergence of two sequential Monte Carlo methods for maximum a posteriori sequence estimation and stochastic global optimization. *Statistics and Computing*, 23(1):91–107, 2013.
26. J. Míguez and M. A. Vázquez. A proof of uniform convergence over time for a distributed particle filter. *arXiv*, 1504.01079v1 [stat.CO], 2015.
27. M. K. Pitt and N. Shephard. Auxiliary variable based particle filters. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, chapter 13, pages 273–293. Springer, 2001.
28. B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, Boston, 2004.

29. J. Rosenblatt and B. Nadler. On the optimality of averaging in distributed statistical learning. *arxiv:1407.2724 [stat.ML] [stat.ML]*, 2014.
30. C. Vergé, C. Dubarry, P. Del Moral, and E. Moulines. On parallel implementation of sequential monte carlo methods: the island particle model. *Statistics and Computing*, pages 1–18, 2013.
31. N. Whiteley, A. Lee, and K. Heine. On the role of interaction in sequential monte carlo algorithms. *arXiv:1309.2918 [stat.CO]*, 2013.