Compressed particle methods for expensive models with application in Astronomy and Remote Sensing

Luca Martino[⊤], Víctor Elvira[*], Javier López-Santiago[†], Gustau Camps-Valls[‡]

[⊤] Universidad Rey Juan Carlos (URJC), Madrid, Spain.

[*]University of Edinburgh, Edinburgh, EH9 3FD, U.K.

[†] Universidad Carlos III de Madrid (UC3M), Madrid, Spain.

[‡] Universitat de Valencia (UV), Valencia, Spain.

March 2, 2021

Abstract

In many inference problems, the evaluation of complex and costly models is often required. In this context, Bayesian methods have become very popular in several fields over the last years, in order to obtain parameter inversion, model selection or uncertainty quantification. Bayesian inference requires the approximation of complicated integrals involving (often costly) posterior distributions. Generally, this approximation is obtained by means of Monte Carlo (MC) methods. In order to reduce the computational cost of the corresponding technique, surrogate models (also called emulators) are often employed. Another alternative approach is the so-called Approximate Bayesian Computation (ABC) scheme. ABC does not require the evaluation of the costly model but the ability to simulate artificial data according to that model. Moreover, in ABC, the choice of a suitable distance between real and artificial data is also required. In this work, we introduce a novel approach where the expensive model is evaluated only in some well-chosen samples. The selection of these nodes is based on the so-called compressed Monte Carlo (CMC) scheme. We provide theoretical results supporting the novel algorithms and give empirical evidence of the performance of the proposed method in several numerical experiments. Two of them are real-world applications in astronomy and satellite remote sensing.

Keywords: *Numerical Inversion, Bayesian inference, Particle Filtering, Importance Sampling, Astronomy, Remote Sensing.*

1 Introduction

In many areas of science and engineering, systems are analyzed by studying physical models and running computer simulations, which serve as convenient approximations to reality. Depending on the body of literature, they are known as physics-based, processed-oriented, mechanistic models, or simply simulators [1, 2]. Simulators and their corresponding surrogate models are ubiquitous in physics, brain, Earth, climate, and social sciences [3–7], but also in industrial environments for developing new manufactured products and infrastructures, to quantify performance of engineering systems, to understand and assess supply chains, or in robotics and vehicle design [8–11]. Model simulations are needed to understand system behaviour, but also to perform interventional and counterfactual studies.

Since common forward models (simulators) are computationally costly, both running simulations or inverting them for parameter prediction becomes a big challenge. Machine learning models are widely used to learn both the forward and inverse functions, and nowadays they routinely replace complex models and sub-components to improve *scalability* and *mathematical tractability*. These models are commonly known as *emulators* and report excellent accuracy-speedup trade-offs compared to simulators, besides elegant ways to do uncertainty quantification, error propagation, and sensitivity analysis [12].

Bayesian methods are often applied for parameter inversion, model selection or uncertainty quantification [13, 14]. In their common implementation, these techniques require the evaluation of the possible complex and costly model. When the model is particularly expensive (or its pointwise evaluation is impossible), generally two approaches are employed. In the first one, the true model is replaced by a surrogate model (i.e., an emulator) that could be adaptively improved [15–18]. Then, Bayesian inference is carried out on the approximate and cheaper model. The second approach is the so-called *approximate Bayesian computation* (ABC) [19–21]. In the standard ABC scheme, model evaluation is substituted by evaluating a distance between the observed data and some artificial data generated according to the model. Therefore, ABC does not need to evaluate the model but to simulate artificial data from it. Different types of distances can be used. It is important to remark the choice of distance can be interpreted as a choice of an approximate observation model.

In this work, we consider an alternative approach. The core idea is to reduce the number of true model evaluations by a suitable selection of the inputs where we evaluate the model. In this way, we can obtain a great reduction in the required computational time, at the expense of a slight increase of the estimation error. The key point is a proper selection of inputs where to evaluate the costly model. We present the novel approach in the context of particle filtering where the variables of interest can also vary with time. The method is based on a technique called compressed Monte Carlo (CMC) scheme, which summarizes the information contained in N weighted Monte Carlo samples into $M < N$ weighted particles (called *summary particles*), based on a stratification approach [14, 22]. We aim at reducing the loss of information in terms of moment matching, in a similar fashion of the deterministic-based quadrature rules [23–28]. We provide different theoretical results supporting the novel approach.

We also give empirical evidence of the performance of the proposed method in four different numerical experiments, both over simulations and real challenging problems. In particular, we consider the problem of object detection (planets, satellites, etc.) in an N-body system observed from the Earth. The observation model is complex and costly, especially for some set of parameters (see Section 6). The second real model considers the inversion of a radiative transfer model (RTM) which encodes the energy transfer through the atmosphere. This model is used to understand and model vegetation, as well as to estimate the parameters that describe the status of the Earth from satellite observations by inversion.

The paper is structured as follows. Section 2 describes the problem statement and recalls some background material. In Section 3, we introduce the CMC approach. In Section 4, we provide some theoretical results. In Section 5, we introduce the novel compressed particle filtering algorithms. Numerical simulations are given in Section 6. Finally, some conclusions are provided in Section 7.

2 Problem statement

In many real-world applications, it is required to characterize the posterior probability density function (pdf) of a set of unknown parameters given the observed data. More specifically, denoting the vector of unknowns as $\mathbf{x} = [x_1, \ldots, x_{d_X}]^\top \in \mathcal{D} \subseteq \mathbb{R}^{d_X}$ and the observed data as $\mathbf{y} \in \mathbb{R}^{d_Y}$, the pdf is defined as

$$\bar{\pi}(\mathbf{x}|\mathbf{y}) = \frac{\ell(\mathbf{y}|\mathbf{x})g(\mathbf{x})}{Z(\mathbf{y})} \propto \ell(\mathbf{y}|\mathbf{x})g(\mathbf{x}) = \pi(\mathbf{x}|\mathbf{y}), \tag{1}$$

where $\bar{\pi}(\mathbf{x}|\mathbf{y})$ and $\pi(\mathbf{x}|\mathbf{y})$ denote the normalized and unnormalized posterior, respectively, $\ell(\mathbf{y}|\mathbf{x})$ is the likelihood function, $g(\mathbf{x})$ is the prior pdf, and $Z(\mathbf{y})$ is the normalization factor, which is usually called *marginal likelihood* or *Bayesian model evidence*. Hereinafter, we will remove the dependence on $\mathbf{y}$ to simplify the notation. A generic integral involving the density of the random variable $\mathbf{X} \sim \bar{\pi}(\mathbf{x}) = \frac{1}{Z}\pi(\mathbf{x})$ is given by

$$I(h) \triangleq \mathbb{E}_{\bar{\pi}}[h(\mathbf{X})] = \int_{\mathcal{D}} h(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x} \\ = \frac{1}{Z}\int_{\mathcal{D}} h(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}, \qquad (2)$$

where $h(\mathbf{x})$ is an integrable function of $\mathbf{x}$.[1] For the sake of simplicity, we assume that the functions $h(\mathbf{x})$ and $\bar{\pi}(\mathbf{x})$ are continuous in $\mathcal{D}$, and the integrand function, $h(\mathbf{x})\bar{\pi}(\mathbf{x})$, in Eq. (2) is integrable. In many practical scenarios, we cannot obtain an analytical solution for (2), and Monte Carlo methods are often applied. More generally, we are interested in obtaining a particle approximation $\widehat{\pi}^{(N)}(\mathbf{x})$ of the measure of $\bar{\pi}(\mathbf{x})$, formed by a cloud of weighted samples [13]. See next section, for further details.

2.1 Importance Sampling (IS) approximations

A well-known Monte Carlo approach is the importance sampling (IS) technique [14, 29]. Let us consider N samples $\{\mathbf{x}_n\}_{n=1}^{N}$ drawn from a proposal pdf, $q(\mathbf{x})$, such that $q(\mathbf{x}) > 0$ where $\bar{\pi}(\mathbf{x}) > 0$. We also assume that $q(\mathbf{x})$ has heavier tails than the target, $\bar{\pi}(\mathbf{x})$, since this assumption ensures that the resulting IS estimator has finite variance [13, 14]. We assign a weight to each sample and then we normalize them as follows,

$$w_n = \frac{\pi(\mathbf{x}_n)}{q(\mathbf{x}_n)}, \qquad \bar{w}_n = \frac{w_i}{\sum_{j=1}^{N} w_j}, \qquad (3)$$

[1] We assumed $h(\mathbf{x}) : \mathbb{R}^{d_X} \to \mathbb{R}$ and the integral $I(h) \in \mathbb{R}$ is a scalar value. However, more generally, we have $\mathbf{h}(\mathbf{x}) : \mathbb{R}^{d_X} \to \mathbb{R}^{\nu}$ and $\mathbf{I}(\mathbf{h}) \in \mathbb{R}^{\nu}$ where $\nu \geq 1$. For simplicity, we keep the simpler notation with $\nu = 1$.

with $n = 1, \dots, N$. Therefore, the moment of interest can be approximated as

$$\begin{aligned} \widehat{I}^{(N)}(h) &= \frac{1}{N\widehat{Z}} \sum_{n=1}^{N} w_n h(\mathbf{x}_n) && (4) \\ &= \sum_{n=1}^{N} \bar{w}_n h(\mathbf{x}_n), && (5) \end{aligned}$$

where $\widehat{Z} = \frac{1}{N} \sum_{n=1}^{N} w_n$ is a unbiased estimator of the marginal likelihood $Z = \int_{\mathcal{D}} \pi(\mathbf{x}) d\mathbf{x}$, which is a useful quantity for model selection and hypothesis testing [14]. The particle approximation of the measure of $\bar{\pi}$ is given by

$$\widehat{\pi}^{(N)}(\mathbf{x}) = \sum_{n=1}^{N} \bar{w}_n \delta(\mathbf{x} - \mathbf{x}_n), \qquad (6)$$

where $\delta(\mathbf{x})$ is the Dirac delta function.

2.2 Particle filtering for state-space models

In the sequential scenario, the inference problem often concerns a sequence of variables of interest $\mathbf{x}_{0:T} = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_T]$ (a.k.a., *trajectory*) given a sequence of related observations $\mathbf{y}_{1:T} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T]$, where T represents the last time step. The corresponding state-space model is completely defined by an initial density $p(\mathbf{x}_0)$, a transition density and the likelihood function), i.e.,

$$\begin{cases} \mathbf{x}_t \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}), \\ \mathbf{y}_t \sim p(\mathbf{y}_t|\mathbf{x}_t), \end{cases} \qquad t = 1, \dots, T. \qquad (7)$$

The complete posterior density is given by

$$\bar{\pi}(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}) \propto p(\mathbf{x}_0) \left[\prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}) \right] \left[\prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t) \right]. \qquad (8)$$

Efficient Monte Carlo techniques for approximating the posterior $\bar{\pi}(\mathbf{x}_{0:T}|\mathbf{y}_{1:T})$ are the so-called particle filtering algorithms. A particle filter (PF) combines the sequential importance sampling approach with resampling steps. A standard PF is detailed in Table 1. The resampling steps are performed when an effective sample size (ESS) approximation $\widehat{ESS}$ in smaller than ηN where $\eta \in [0,1]$ [30]. Examples of ESS are

$$\widehat{ESS}\big(\bar{w}_t^{(1:N)}\big) = \frac{1}{\sum_{n=1}^{N}(\bar{w}_t^{(n)})^2}, \tag{9}$$

$$\widehat{ESS}\big(\bar{w}_t^{(1:N)}\big) = \frac{1}{\max_n \bar{w}_t^{(n)}}. \tag{10}$$

Computational cost. Note that, at each iteration, we have N evaluations of the likelihood function (i.e., the observation model). Hence, after T iterations of the filter, we have NT model evaluations. Furthermore, the resampling steps (when performed) are done over N possible particles. The cost of the resampling grows with N. More precisely, the complexity of the resampling procedure is of $O(N)$ [31].

3 Compressed Monte Carlo (CMC)

In this section, we describe a procedure for compressing the information contained in a set of N weighted samples $\{\mathbf{x}_n, \bar{w}_n\}_{n=1}^N$ obtained via importance sampling, with a smaller amount $M \leq N$ of weighted samples $\{\mathbf{s}_m, \widehat{a}_m\}_{m=1}^M$.[2] The CMC approach is based on the so-called stratified sampling [22, 34]. The idea is to divide the support domain $\mathcal{D}$ of the random variable $\mathbf{X}$ into M disjoint, mutually exclusive regions. More specifically, let us consider an integer $M \in \mathbb{N}^+$ with

[2] The case $M = N$ corresponds to the uncompressed scenario.

Table 1: A standard Particle Filter

```
**Initialization:** Choose $N$, $\eta \in [0,1]$, $\bar{\mathbf{x}}_0^{(i)}$, with $i = 1, \ldots, N$, and $\widehat{ESS}$ [30]. Set $w_0^{(n)} = 1$ for all $n$.

**For** $t = 1, \ldots, T$ :

    1. Draw $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t | \bar{\mathbf{x}}_{t-1}^{(i)})$, with $i = 1, \ldots, N$.
    2. Compute the $N$ weights
           $w_t^{(n)} = w_{t-1}^{(n)} p(\mathbf{y}_t | \mathbf{x}_t^{(i)}), \quad m = 1, \ldots, M.$   (11)
       and normalized them $\bar{w}_t^{(n)} = \frac{w_t^{(n)}}{\sum_{k=1}^N w_t^{(k)}}$.
    3. if $\widehat{ESS}\big(\bar{w}_t^{(1:N)}\big) \leq \eta N$:
        • Obtain $\{\bar{\mathbf{x}}_t^{(n)}\}_{n=1}^N$, by resampling $N$ times within $\{\mathbf{s}_m\}_{m=1}^M$ according to $\bar{w}_t^{(m)}$, with $m = 1, \ldots, M$.
        • Set $\widehat{Z}_t = \frac{1}{N} \sum_{n=1}^N w_t^{(n)}$ (see [32, 33]), and
              $w_t^{(1)} = \ldots = w_t^{(N)} = \widehat{Z}_t.$

**Outputs:** Return $\{\mathbf{x}_t^{(n)}, w_t^{(n)}\}_{n=1}^N$ for $t = 1, \ldots, T$.
```

$M < N$, and a partition $\mathcal{P} = \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_M\}$ of the state space with M disjoint subsets,

$$\begin{aligned} &\mathcal{X}_1 \cup \mathcal{X}_2 \cup \ldots \cup \mathcal{X}_M = \mathcal{D}, \\ &\mathcal{X}_i \cap \mathcal{X}_k = \emptyset, \qquad i \neq k, \quad \forall i, j \in \{1, \ldots, M\}. \end{aligned} \tag{12}$$

We assume that all $\mathcal{X}_m$ are convex sets. Now, let us consider N weighted samples $\{\mathbf{x}_n, \bar{w}_n\}_{n=1}^N$. Given the partition in Eq. (12), i.e., $\mathcal{X}_1 \cup \mathcal{X}_2 \cup \ldots \cup \mathcal{X}_M = \mathcal{D}$ formed by convex, disjoint sub-regions $\mathcal{X}_m$, we denote the subset of the set of indices $\{1, \ldots, N\}$,

$$\mathcal{J}_m = \{\text{all } i \in \{1, \ldots, N\} : \quad \mathbf{x}_i \in \mathcal{X}_m\},$$

which are associated to the samples in the m-th sub-region $\mathcal{X}_m$. The cardinality $|\mathcal{J}_m|$ denotes the number of samples in $\mathcal{X}_m$, and clearly we have $\sum_{m=1}^{M} |\mathcal{J}_m| = N$.

3.1 CMC approximation

We can summarize the information contained in the particle approximation $\widehat{\pi}^{(N)}(\mathbf{x})$ of Eq. (6), by constructing an empirical stratified approximation based on M weighted particles $\{\mathbf{s}_m, \widehat{a}_m\}_{m=1}^{M}$ (where the $\mathbf{s}_m$ are *summary particles*), i.e.,

$$\widetilde{\pi}^{(M)}(\mathbf{x}) = \sum_{m=1}^{M} \widehat{a}_m \delta(\mathbf{x} - \mathbf{s}_m), \quad \mathbf{s}_m \in \mathcal{X}_m, \tag{13}$$

where

$$\widehat{a}_m \approx \mathbb{P}(\mathbf{X} \in \mathcal{X}_m) = \int_{\mathcal{X}_m} \bar{\pi}(\mathbf{x}) d\mathbf{x} = \frac{1}{Z} \int_{\mathcal{X}_m} \pi(\mathbf{x}) d\mathbf{x}. \tag{14}$$

We here refer to $\widehat{a}_m$ as summary weights, and to $\mathbf{s}_m$ as summary particles.

3.2 Summary weights

The weights $\widehat{a}_m$ can be obtained using the IS approximation $\widehat{\pi}^{(N)}$ with N samples, i.e.,

$$\begin{aligned} \widehat{a}_m &= \int_{\mathcal{X}_m} \widehat{\pi}^{(N)}(\mathbf{x}) d\mathbf{x} = \sum_{n=1}^{N} \bar{w}_n \int_{\mathcal{X}_m} \delta(\mathbf{x} - \mathbf{x}_i) d\mathbf{x}, \\ &= \sum_{n \in \mathcal{J}_m} \bar{w}_n. \end{aligned} \tag{15}$$

By defining

$$\widehat{Z}_m = \frac{1}{N} \sum_{i \in \mathcal{J}_m} w_i, \qquad \widehat{Z} = \sum_{m=1}^{M} \widehat{Z}_m = \frac{1}{N} \sum_{n=1}^{N} w_n, \tag{16}$$

we can also obtain another expression $\widehat{a}_m$, i.e.,

$$\widehat{a}_m = \frac{\widehat{Z}_m}{\widehat{Z}} = \sum_{i \in \mathcal{J}_m} \frac{w_i}{\sum_{n=1}^{N} w_n} = \sum_{i \in \mathcal{J}_m} \bar{w}_i. \tag{17}$$

where $\widehat{Z}_m \approx \int_{\mathcal{X}_m} \pi(\mathbf{x}) d\mathbf{x}$ and $\widehat{Z} \approx \int_{\mathcal{X}} \pi(\mathbf{x}) d\mathbf{x}$. Note that $\sum_{m=1}^{M} \widehat{a}_m = 1$, i.e., they are normalized. Due to Eq. (17), the unnormalized CMC weights are defined as $a_m = \widehat{Z}_m \propto \widehat{a}_m$. They play the same role of the unnormalized weights in IS, indeed the arithmetic mean of a_m's is an estimator of the marginal likelihood.

3.3 Summary particles

We consider different strategies for the selection of the summary particles $\mathbf{s}_m$. The first one is a stochastic approach based on the stratified sampling: each summary particle $\mathbf{s}_m$ is resampled within the set of samples $\mathbf{x}_i \in \mathcal{X}_m$, i.e.,

$$\mathbf{s}_m \in \{\mathbf{x}_i, \text{ with } i \in \mathcal{J}_m\},$$

according to the normalized weights,

$$\bar{w}_{m,i} = \frac{w_i}{\sum_{k \in \mathcal{J}_m} w_k} = \frac{\bar{w}_i}{\sum_{k \in \mathcal{J}_m} \bar{w}_k} = \frac{\bar{w}_i}{\widehat{a}_m}, \quad i \in \mathcal{J}_m. \tag{18}$$

Namely, in that case,

$$\mathbf{s}_m \sim \widehat{\pi}_m(\mathbf{x}) = \sum_{i \in \mathcal{J}_m} \bar{w}_{m,i} \delta(\mathbf{x} - \mathbf{x}_i). \tag{19}$$

Deterministic choices are also possible, for instance setting

$$\mathbf{s}_m = \sum_{j \in \mathcal{J}_m} \bar{w}_{m,j} \mathbf{x}_j, \tag{20}$$

or, if we are interested on the approximation of a specific integral involving a function h, we can set

$$s_m = \sum_{j \in \mathcal{J}_m} \bar{w}_{m,j} h(\mathbf{x}_j). \tag{21}$$

These deterministic rules provide a good performance and enjoy interesting properties, as discussed in the next section. Table 2 summarizes the main notation of the work.

Table 2: Summary of the main notations.

IS		CMC	
w_n	$\bar{w}_n$	a_m	$\widehat{a}_m$
$\frac{\pi(\mathbf{x}_n)}{q(\mathbf{x}_n)}$	$\frac{w_n}{\sum\limits_{i=1}^{N} w_i}$	$\widehat{Z}_m = \frac{1}{N}\sum_{i\in\mathcal{J}_m} w_i$	$\sum\limits_{i\in\mathcal{J}_m} \bar{w}_i = \frac{\widehat{Z}_m}{\widehat{Z}}$
$n = 1, \ldots, N$		$m = 1, \ldots, M$	

Marginal likelihood estimator: $\widehat{Z} = \sum_{m=1}^{M} \widehat{Z}_m = \frac{1}{N}\sum_{n=1}^{N} w_n$.

Partial normalized weights: $\bar{w}_{m,i} = \frac{w_i}{\sum_{k\in\mathcal{J}_m} w_k} = \frac{\bar{w}_i}{\widehat{a}_m}, \quad i \in \mathcal{J}_m$.

CMC estimator: $\widetilde{I}^{(M)}(h) = \sum_{m=1}^{M} \widehat{a}_m h(\mathbf{s}_m)$.

Case of unweighted samples. Let us consider that we have N samples $\{\mathbf{x}_n\}_{n=1}^{N}$ generated by a direct sampling method [35], or an MCMC algorithm [14]. The CMC scheme works in the same manner by setting $\widehat{a}_m = \frac{|\mathcal{J}_m|}{N}$, that represents the ratio of samples within $\mathcal{X}_m$. Moreover, in this scenario, $\bar{w}_{m,i} = \frac{1}{|\mathcal{J}_m|}$ for all $i \in \mathcal{J}_m$.

Examples of partition rules. Given the N samples $\mathbf{x}_n = [x_{n,1}, \ldots, x_{n,d_X}]^\top \in \mathcal{D} \subseteq \mathbb{R}^{d_X}$, with $n = 1, \ldots, N$. Then, we list three practical choices from the simplest to the more sophisticated strategy:

P1 Random grid, where each component of the elements of the grid is contained within the intervals $\min\limits_{n\in\{1,\ldots,N\}} x_{n,i}$ and $\max\limits_{n\in\{1,\ldots,N\}} x_{n,i}$, for each $i = 1, \ldots, d_X$.

P2 Uniform deterministic grid, where each component of the elements of the grid is contained within the intervals $\min_{n\in\{1,\dots,N\}} x_{n,i}$ and $\max_{n\in\{1,\dots,N\}} x_{n,i}$, for each $i = 1,\dots,d_X$.

P3 Voronoi partition obtained by a clustering algorithm with M clusters (e.g., the well-known k-means algorithm).

The procedures above are just possible examples. Note that using a particular partitioning procedure, we can obtain different performance of the resulting algorithms. However, in all the proposed schemes, the theoretical and practical benefits can be observed even applying the simplest rule P1, as we show in the next section and in the numerical experiments (Sect. 6). Finally, note that even the simple procedures P1 and P2 take into account the sample information for building the partition.

4 Properties of CMC

In this section, we discuss some theoretical properties of the CMC schemes. The corresponding proofs are given below or in the related appendix.

Definition. A partition procedure is called *proper* if, when $M = N$, then $|\mathcal{J}_m| = 1$ (note that $m = n$ in this case). Namely, in the limit case of $M = N$, we consider all the MC samples as summary samples, $\mathbf{s}_i = \mathbf{x}_i$ for $i = 1,\dots,N$.

Theorem 1. *Let us consider a fixed set of weighted samples $\mathcal{S} = \{(\mathbf{x}_n, w_n)\}_{n=1}^N$ and a given partition $\mathcal{P}$ (obtained with a proper procedure). Considering the stochastic selection of $\mathbf{s}_m \sim \widehat{\pi}_m(\mathbf{x})$ in Eq. (19), the CMC estimator*

$$\widetilde{I}^{(M)}(h) = \sum_{m=1}^{M} \widehat{a}_m h(\mathbf{s}_m), \qquad \mathbf{s}_m \sim \widehat{\pi}_m(\mathbf{x}), \tag{22}$$

is an unbiased estimator of $\widehat{I}^{(N)}(h)$ *in Eq.* (5)*, i.e.,*

$$\mathbb{E}[\widetilde{I}^{(M)}(h)|\mathcal{S}] = \widehat{I}^{(N)}(h) = \sum_{n=1}^{N} \bar{w}_n h(\mathbf{x}_n). \tag{23}$$

Furthermore, if the partition rule is proper for $M = N$*, the CMC estimator coincides with exactly* $\widehat{I}^{(N)}(h)$*.*

Proof: See Appendix A for the proof. □

Theorem 2. *Let us consider a proper partition procedure. As* $M \rightarrow N$ *and* $N \rightarrow \infty$*, the consistency of CMC estimator is ensured.*

Proof: The IS estimator is consistent as $N \rightarrow \infty$ [14]. If the partition procedure if proper, for $M = N$ the CMC estimator coincides with the standard IS estimator, i.e., $\widetilde{I}^{(M)}(h) = \widehat{I}^{(N)}(h)$ (recall $M = N$). Hence, the corresponding CMC estimator is also consistent. □

Proposition 1. *The estimator of the marginal likelihood* $\widehat{Z} = \frac{1}{N}\sum_{n=1}^{N} w_n$ *is reconstructed with no loss by the CMC estimator* $\widetilde{I}^{(M)} = \frac{1}{M}\sum_{m=1}^{M} a_m$*, i.e.,* $\widetilde{I}^{(M)} = \widehat{Z}$*.*

Proof: Since $a_m = \widehat{Z}_m$ (see Table 2), we have

$$\widetilde{I}^{(M)} = \frac{1}{M}\sum_{m=1}^{M} a_m = \frac{1}{M}\sum_{m=1}^{M} \widehat{Z}_m = \widehat{Z},$$

as shown in Eq. (16). □

If we are interested only in one specific integral $I(h) = \int_{\mathcal{D}} h(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x}$, it is convenient to apply

CMC with the following deterministic choice of the summary particles

$$s_m = \sum_{j \in \mathcal{J}_m} \bar{w}_{m,j} h(\mathbf{x}_j), \tag{24}$$

as highlighted by the theorem below.

Theorem 3. *If s_m is chosen as in Eq. (24), for $m = 1, \ldots, M$,[3] and the linear mapping $f(x) = x$, we have $\widehat{I}^{(N)}(h) = \widetilde{I}^{(M)}(f)$, i.e., we have a perfect reconstruction of the IS estimator.*

Proof: See Appendix B for the proof. □

5 Compressed Particle Filtering

In this section, we show how CMC can be employed for a performance improvement or a decrease of the computational cost of benchmark particle filtering (PF) algorithms. Let us recall the state-space model

$$\begin{cases} \mathbf{x}_t | \mathbf{x}_{t-1} \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}), \\ \mathbf{y}_t | \mathbf{x}_t \sim p(\mathbf{y}_t | \mathbf{x}_t), \end{cases} \qquad t = 1, \ldots, T, \tag{25}$$

described by the propagation kernel, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, and the likelihood function $p(\mathbf{y}_t|\mathbf{x}_t)$. Below, we provide two novel PFs based on CMC. In the first one, called compressed bootstrap particle filter (CBPF) and given in Table 3, based on the so-called bootstrap particle filter, where the resampling is applied at each iteration. In the second one, described in Table 4, where the resampling is applied at each iteration when $\widehat{ESS} \leq \eta N$. We describe the benefits of both compressed particle filter (CPF) techniques.

Benefit 1. In both proposed methods, the compression is applied before the evaluation of the likelihood function $p(\mathbf{y}_t|\mathbf{x}_t)$. The reduction in computational cost is twofold (as shown also in the

[3] Note that in this case $s_m \in \mathbb{R}$ is a scalar value since, for simplicity, we have assumed $h(\mathbf{x}) : \mathbb{R}^{d_X} \to \mathbb{R}$, instead of the more general assumption $\mathbf{h}(\mathbf{x}) : \mathbb{R}^{d_X} \to \mathbb{R}^s$ with $s \geq 1$. All the considerations are also valid for $s \geq 1$.

next point). First of all, both algorithms require the evaluation of the likelihood function only $M < N$ times, at the summary particles $\mathbf{s}_m$. This is particularly convenient if the evaluation of the likelihood is costly due to the number of data, or to a complex measurement model.

Benefit 2. The resampling step is performed over M weighted samples instead of N. This advantage can be found also in other filters proposed in the literature [36,37], which present lower complexity than standard particle filters (decreasing the cost of the resampling steps). Recall that the computational complexity of the resampling procedure is of $O(N)$ in a standard PF, whereas in CPF is $O(M)$ with $M \leq N$ [31].

Benefit 3. Additionally, the application of CMC also helps to prevent the sample impoverishment caused by the resampling operation as also shown in a similar approach [38]. This is due to the fact that the summary particles contain also spatial information regarding the uncompressed particles $\{\mathbf{x}_t^{(n)}\}_{n=1}^N$ [38]. Therefore, the resampling in CPFs takes into account both, the normalized weights and spatial information (not only the weights, as in resampling steps in standard particle filters without applying CMC). The results in Section 6.1 confirm that the application of CMC ensures a better approximation of the empirical measure defined by N weighted samples.

In summary, CPFs are clearly cheaper and faster than the corresponding classical particle filters. Note that the CMC weights $\widehat{a}_m$ are included in particle weights in Eq. (27). The weighted summary particles $\{\mathbf{s}_m, \widehat{a}_m\}_{m=1}^M$ play a similar role than the sigma points in the unscented Kalman filter (UKF) [24,26].

Remark. If the computational time is mainly specified by the likelihood evaluation, the CPFs provide better performance than the corresponding standard particle filtering schemes, for a fixed time budget.

Limitations and further considerations. Compared with a standard filter with N particles, the CPFs provide some performance loss in terms of estimation error, since the compressed filters use less likelihood evaluation ($M < N$). However, for a fixed budget of evaluations of the likelihood function, the compressed filters provide the best results, as depicted in Figure 4, which shows the benefits of the proposed compression procedure. The use of the compressed filters is recommended only when the cost of the evaluation of likelihood function is significantly higher than the operations required in the compression (e.g., inequalities checking and sums). Therefore, the use of the CPF is required only when the evaluation of the model is costly. It is also remarkable that the CPFs share some features and also present a robust behavior, similarly to the so-called *Gaussian particle filters* [36, 37] and the *approximate-grid particle filters* [39, 40]. Another limitation of the proposed scheme is that the CPFs are less suitable for parallelization than standard schemes. Finally, the design of refined adaptive partitions and adaptation of the compression to more sophisticated filters, as the auxiliary particle filters [41], deserve and require additional future works.

Table 3: The Compressed Bootstrap Particle Filter (CBPF)

```
Initialization: Choose $N$, $M < N$, and $\bar{\mathbf{x}}_0^{(i)}$, with $i = 1, \dots, N$.

For $t = 1, \dots, T$ :
    1. Draw $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t | \bar{\mathbf{x}}_{t-1}^{(i)})$, with $i = 1, \dots, N$.
    2. Apply a CMC scheme to $\{\mathbf{x}_t^{(n)}, \frac{1}{N}\}_{n=1}^N$ obtaining $\{\mathbf{s}_m, \widehat{a}_m\}_{m=1}^M$.
    3. Compute the $M$ weights
           $w_t^{(m)} = \widehat{a}_m p(\mathbf{y}_t | \mathbf{s}_m), \quad m = 1, \dots, M.$    (26)
       and normalized them $\bar{w}_t^{(m)} = \frac{w_t^{(m)}}{\sum_{k=1}^M w_t^{(k)}}$.
    4. Obtain $\{\bar{\mathbf{x}}_t^{(n)}\}_{n=1}^N$, by resampling $N$ times within $\{\mathbf{s}_m\}_{m=1}^M$ according to $\bar{w}_t^{(m)}$, with
       $m = 1, \dots, M$.
```

Table 4: Generic CPF

Initialization: Choose M and N such that N is a multiple of M, i.e.,

$$K = \frac{N}{M} \in \mathbb{N}^{+}.$$

Moreover, choose $\eta \in [0,1]$, $\bar{\mathbf{x}}_0^{(i)}$, with $i = 1, \ldots, N$, and an effective sample size approximation $\widehat{ESS}$ [30]. Set $\rho_0^{(i)} = \frac{1}{N}$ for all $i = 1, \ldots, N$.

For $t = 1, \ldots, T$:

1. Draw $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t | \bar{\mathbf{x}}_{t-1}^{(i)})$, with $i = 1, \ldots, N$.
2. Apply a CMC scheme to $\{\mathbf{x}_t^{(n)}, \rho_{t-1}^{(n)}\}_{n=1}^{N}$ obtaining $\{\mathbf{s}_m, \widehat{a}_m\}_{m=1}^{M}$.
3. Compute the M weights
$$w_t^{(m)} = \widehat{a}_m p(\mathbf{y}_t | \mathbf{s}_m), \quad m = 1, \ldots, M. \tag{27}$$
and normalized them $\bar{w}_t^{(m)} = \frac{w_t^{(m)}}{\sum_{k=1}^{M} w_t^{(k)}}$.
4. if $\widehat{ESS}\big(\bar{w}_t^{(1:M)}\big) \leq \eta M$:
    - Obtain $\{\bar{\mathbf{x}}_t^{(n)}\}_{n=1}^{N}$, by resampling N times within $\{\mathbf{s}_m\}_{m=1}^{M}$ according to $\bar{w}_t^{(m)}$, with $m = 1, \ldots, M$.
    - Set
    $$\rho_t^{(1)} = \ldots.. = \rho_t^{(N)} = \sum_{m=1}^{M} w_t^{(m)}.$$
    For further details see [32, 33].
5. Otherwise, if $\widehat{ESS}\big(\bar{w}_t^{(1:M)}\big) > \eta M$, set
$$\bar{\mathbf{x}}_t^{(n)} = \mathbf{s}_m, \quad \rho_t^{(n)} = w_t^{(m)}, \quad \text{with} \quad m = \left\lceil \frac{n}{K} \right\rceil,$$
with $n = 1, \ldots, N$.

Regularized Resampling. In the resampling steps of CPFs, we have

$$\bar{\mathbf{x}}_t^{(n)} \sim \sum_{m=1}^{M} \bar{w}_t^{(m)} \delta(\mathbf{x} - \mathbf{s}_m). \tag{28}$$

The use of CMC provides clear advantages as discussed above in Benefit 3. Additionally, in both CPF schemes, we can also employed a regularized resampling in order to reduce also the loss of diversity in the cloud of particles. We can replace the delta functions in Eq. (28) with other kernel functions. For instance, we can consider Gaussian kernels $K(\mathbf{x}|\mathbf{s}_m, \mathbf{\Sigma}_m)$, of mean $\mathbf{s}_m$ and with a $d_X \times d_X$ covariance matrix $\mathbf{\Sigma}_m$ the $d_X \times d_X$ obtained by an empirical estimation considering the samples in $\mathcal{X}_m$, i.e.,

$$\mathbf{\Sigma}_m = \sum_{j \in \mathcal{J}_m} \bar{w}_{m,j} (\mathbf{x}_j - \mathbf{s}_m)(\mathbf{x}_j - \mathbf{s}_m)^\top + \epsilon \mathbf{I}, \tag{29}$$

where $\mathbf{s}_m$ is defined in Eq. (20) and $\epsilon > 0$. Hence, in this case, we have

$$\bar{\mathbf{x}}_t^{(n)} \sim \sum_{m=1}^{M} \bar{w}_t^{(m)} K(\mathbf{x}|\mathbf{s}_m, \mathbf{\Sigma}_m), \tag{30}$$

where $K(\cdot)$ represents a kernel function with location parameter $\mathbf{s}_m$ and covariance matrix $\mathbf{\Sigma}_m$. A similar regularized resampling is implicitly used in [36, 37].

Adapting M. Let us consider the CBPF algorithm in Table 3. We can adapt the number of summary particles M used at each iteration. The underlying idea is that if $\widehat{ESS}$ is small, we need a less number M of summary particles to summarize the information in $\{\mathbf{x}_n, \bar{w}_n\}_{n=1}^N$. Otherwise, if $\widehat{ESS}$ is high, we could need more summary particles for encoding all the statistical information contained in $\{\mathbf{x}_n, \bar{w}_n\}_{n=1}^N$. Then we can set

$$M_t = \max \left[\gamma \left\lfloor \widehat{ESS} \right\rfloor, M_{\mathtt{min}} \right] \quad t = 1, \ldots, T, \tag{31}$$

with $\gamma > 0$ and $M_{\mathtt{min}} \geq 1$.

6 Numerical experiments

In the section, we test the proposed method in five different numerical experiments, comparing its performance with benchmark methods. The first three numerical examples consider artificial models and simple distributions, showing the advantages of the proposed compression scheme even in these scenarios. The fourth numerical experiment considers the problem of object detection (planet, satellite, etc.) in an N-body system observed from the Earth. The observation model is complex and costly, especially for some set of parameters. In the last experiment, we consider the inversion of a radiative transfer model (RTM) called PROSAIL, which models the energy transfer through the atmosphere. This model is used to model and understand vegetation status from satellite observations.

6.1 CMC versus standard resampling

For simplicity, Let us consider $x \in \mathbb{R}^+$. Moreover, we consider two possible target densities: the first one is a Gamma pdf

$$\bar{\pi}(x) \propto x^{\alpha-1} \exp\left(-\frac{x}{\kappa}\right), \tag{32}$$

with $\alpha = 4$ and $\kappa = 0.5$, and the second one is a mixture of two Gaussians, with $x \in \mathbb{R}$,

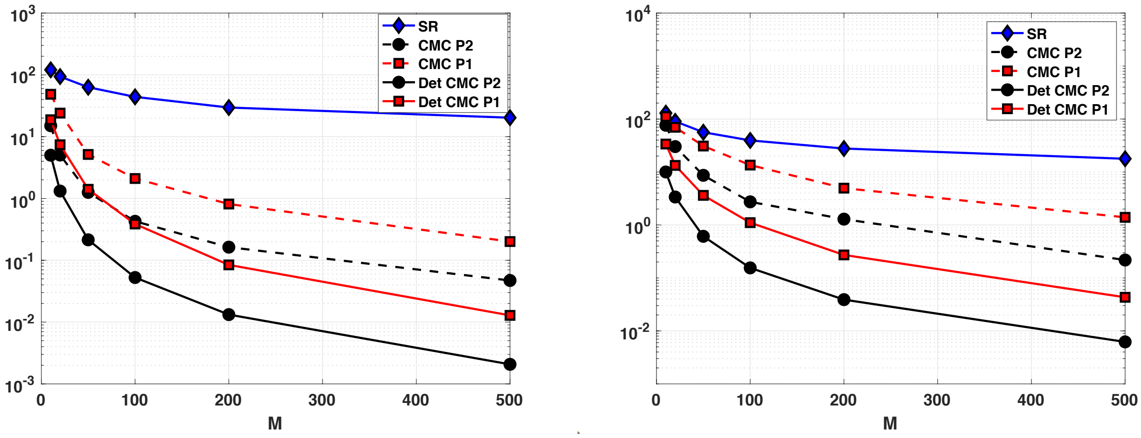
$$\bar{\pi}(x) = \frac{1}{2}\mathcal{N}(x|-2,1) + \frac{1}{2}\mathcal{N}(x|4,0.25). \tag{33}$$

Experiment: At each run, we generate $N = 10^5$ Monte Carlo samples $\{x_n, \frac{1}{N}\}_{n=1}^N$ from the target pdfs. We compare the standard resampling (SR) strategy with different CMC schemes. Namely, with SR, we resample uniformly M times within $\{x_n\}_{n=1}^N$ obtaining $\{s_m, \frac{1}{M}\}_{m=1}^M$ and, with the CMC schemes, we obtain $\{s_m, \widehat{a}_m\}_{m=1}^M$. Then, at each run, we compute the Root Mean Square Error (RMSE) for estimating the first 5 moments of the corresponding target pdf (using the M summary particles). Regarding the CMC schemes, we consider two kind of partition procedures:

random (P1) and uniform (P2) described in Section 3.1. Furthermore we compare the stochastic and the deterministic choices of the summary particles s_m described in Section 3.3. For the deterministic CMC we refer to the use of Eq. (20) for s_m. We repeat the experiment 10^3 independent runs and average the results.

Figure 1 depicts the averaged RMSE as function of the number M of summary particles. Figure 1-**(a)** refers to the Gamma target pdf, whereas Figure 1-**(b)** corresponds to the Gaussian mixture pdf. The results of the SR method are displayed with triangles. The stochastic CMC schemes are shown with dashed lines, whereas the deterministic CMC schemes with solid lines.

Discussion: In all cases, CMC outperforms SR and the deterministic CMC schemes provide the better results. Clearly, the partition P2 (circles) outperforms P1 (squares). Note that P1 represents the simplest and perhaps the worst possible construction of the partition. However, it is important to remark that the CMC schemes, even with P1, outperform the SR method. In this experiment, the differences in computational time are negligible, and the CMC schemes provide always the best performance.



(a) Gamma target pdf

(b) Mixture target pdf

Figure 1: RMSE as function of M. The results obtained by SR is depicted with a solid line and rhombuses. The results of CMC with a random partition (P1) and with a grid partition (P2) are shown by squares and circles, respectively. The results obtained with the deterministic choice of $\mathbf{s}_m$ in Eq. (20) are shown with solid lines (squares and circles), whereas the results random choice of $\mathbf{s}_m$ are provided with dashed lines (squares and circles).

6.2 Second Experiment

This section is devoted to analyze the performance of the compressed bootstrap particle filter (CBPF) described in Table 3. Let us consider the state-space model

$$\begin{cases} x_t = |x_{t-1}| + v_t, \\ y_t = \log(x_t^2) + u_t, \end{cases} \qquad t = 1, \ldots, T, \tag{34}$$

where $v_t \sim \mathcal{N}(0,1)$ and $u_t \sim \mathcal{N}(0,1)$. The goal is to track x_t for $T = 100$ steps, with a particle filtering algorithm considering $N \in \{100, 1000\}$ particles. We compare the bootstrap particle filter (BPF) [42] with its compressed version (i.e., CBPF) in terms of the Root Mean Square Error (RMSE) in estimation of $x_{1:T}$. We apply CBPF with different values of M (clearly, with $M \leq N$). We consider the deterministic CMC scheme with a uniform construction P2 of the partition.

Figure 2 shows the RMSE (averaged over 5000 independent runs) as function of the compression rate $\frac{M}{N}$. The solid lines represent the RMSE obtained by the BPF. The dashed line with squares corresponds to the CBPF (using the deterministic compression) with $N = 100$, whereas the dashed line with circles corresponds to the CBPF with $N = 1000$. Note that CBPF virtually obtains the same performance of the BPF with approximately 85% less evaluations of the likelihood function. Clearly, in this toy example, the likelihood evaluation is not expensive, and the differences in computational time are negligible. This is not the case in the two real-world experiments, in Sections 6.4 and 6.5, where the gain in computational time is relevant. We also recall that the N resampling steps are performing over M particles instead of N. Furthermore, fixing the compression rate $\frac{M}{N}$, It is interesting to note that the performance of CBPF improves when N grows.

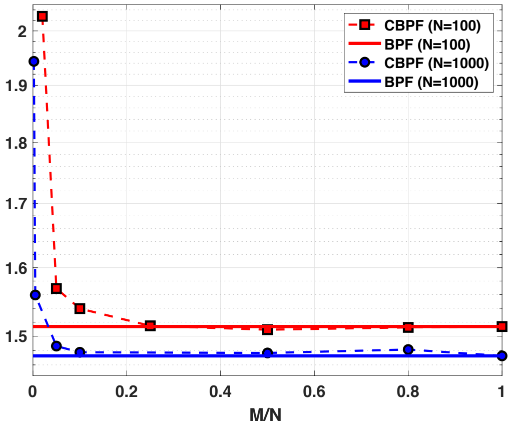


Figure 2: RMSE (log-domain in the y-axis) as function of the ratio $\frac{M}{N}$. The dashed line with squares corresponds to the CBPF with $N = 100$, whereas with circles corresponds to the CBPF with $N = 1000$. The solid lines corresponds to the bootstrap particle filter (BPF) with $N = 100, 1000$. CBPF virtually obtains the same performance of the bootstrap particle filter with approximately 85% less evaluations of the likelihood function.

6.3 Third Experiment

We now repeat the previous experiment considering another state-space model. More specifically, we consider the benchmark growth model, as in [39], using also the the same parameters as in [39], i.e.,

$$\begin{cases} x_t = f_t(x_{t-1}) + v_t, \\ y_t = \frac{1}{20} x_t^2 + u_t, \end{cases} \qquad t = 1, \dots, T, \tag{35}$$

where

$$f_t(x_{t-1}) = \frac{1}{2} x_{t-1}^2 + \frac{25 x_{t-1}}{1 + x_{t-1}^2} + \cos(1.2t),$$

and $v_t \sim \mathcal{N}(0, 10)$ and $u_t \sim \mathcal{N}(0, 1)$. The goal is to estimate the temporal trajectory of the state x_t for $T = 100$ steps, with a particle filtering algorithm considering $N \in \{100, 1000\}$ particles. Again, we compare the bootstrap particle filter (BPF) [42], with its compressed version, CBPF, with different values of $M \leq N$. in terms of the Root Mean Square Error (RMSE) in the estimation of the trajectory $x_{1:T}$. We consider the deterministic CMC scheme with a uniform construction P2 of the partition.

Figure 3 shows the RMSE (averaged over 10^3 independent runs) as function of the ratio $\frac{M}{N}$. The solid lines provide the RMSE obtained by the standard BPF. The dashed line with squares corresponds to the CBPF with $N = 100$, whereas the dashed line with circles corresponds to the CBPF with $N = 1000$. Note that CBPF, with $N = 100$, obtains virtually the same performance of the BPF with 70% less likelihood evaluations. With $N = 1000$, CBPF obtains the same performance of the BPF with 98% less likelihood evaluations. As in Section 6.2, in this toy example, the likelihood evaluation is not expensive, and the differences in computational time are negligible. Fixing the compression rate $\frac{M}{N}$, we can also observe that the performance of CBPF improves when N grows.

In Figure 4, we compare the standard BPF and CBPF but, in this case, considering the same number of likelihood evaluations. Therefore, both the standard BPF and CBPF waste M evaluations of the likelihood function per iteration. We can observe that CBPF provides always the smallest RMSE and the difference in RMSE increases when the compression is bigger, i.e., in left side of the figure. Thus, fixing the likelihood evaluation budget, the proposed compression scheme is an efficient procedure for managing this budget. The results in both Figures 3-4 show the benefits of the proposed approach.

6.4 Inference in Kepler's models

In recent years, the problem of revealing objects orbiting other stars has acquired large attention. Different techniques have been proposed to discover exo-objects but, nowadays, the radial velocity technique is still the most used [43–46]. The problem consists in fitting a model (the so-called radial velocity curve) to data acquired at different moments spanning during long time periods (up to years). The model is highly non-linear and it is costly in terms of computation time (specially, for certain sets of parameters). Obtaining a value to compare to a single observation involves numerically integrating a differential equation in time or an iterative procedure for solving to a non-linear equation. Typically, the iteration is performed until a threshold is reached or

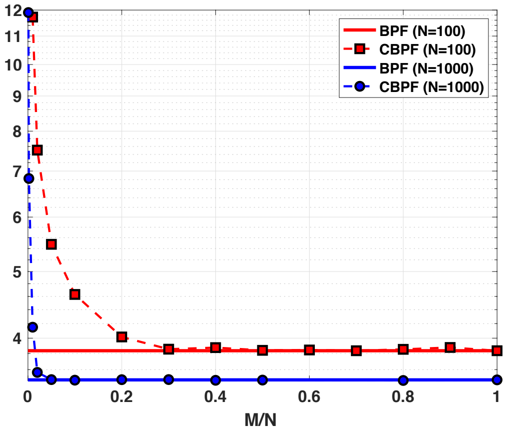


Figure 3: RMSE (log-domain in the y-axis) as function of the ratio $\frac{M}{N}$. The dashed line with (squares and circles) corresponds to the CBPF with $N \in \{100, 1000\}$. The solid lines correspond to the BPF with $N \in \{100, 1000\}$. With $N = 1000$, CBPF obtains the same performance of the BPF, with 98% less evaluations of the likelihood function (i.e., only $M = 20$ evalutions).

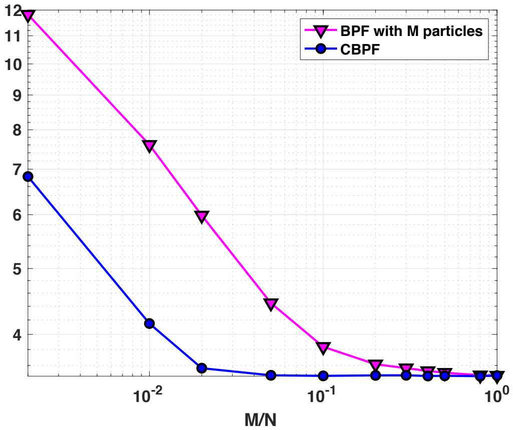


Figure 4: RMSE (log-log-domain) as function of the ratio $\frac{M}{N}$. The circles correspond to the CBPF with different values of M from 2 to 1000, and $N = 1000$. The triangles correspond to the standard BPF with M particles. Both filters, CBPF and BPF, have the same number of likelihood evaluations. CBPF always provides the smallest RMSE and the difference in RMSE increases when the compression is larger (left side of the figure).

Table 5: Description of parameters in Eq. (36).

Parameter	Description	Units
For each planet		
K_i	amplitude of the curve	$\mathrm{m\,s^{-1}}$
$u_{i,t}$	true anomaly	rad
$\omega_{i,t}$	longitude of periastron	rad
e_i	orbit's eccentricity	...
P_i	orbital period	s
τ_i	time of periastron passage	s
Below: not depending on the number of objects/satellite		
V_0	mean radial velocity	$\mathrm{m\,s^{-1}}$

10^6 iterations are performed. The problem of radial velocity curve fitting is applied in several related applications. It is similar to the problem of determining the orbits of spectroscopic binary stars [47, 48] or the stars surrounding the galactic center [49]. In the following, we describe an orbital model, which is equivalent for any N-body system observed from Earth, i.e. exoplanetary systems, binary stellar system, double pulsars, etc.

6.4.1 Likelihood and transition functions

When analysing radial velocity data of an exoplanetary system, it is commonly accepted that the *wobbling* of the star around the centre of mass is caused by the sum of the gravitational force of each planet independently and that they do not interact with each other. Each planet follows a Keplerian orbit and the radial velocity of the host star is given by

$$y_{r,t} = V_0 + \sum_{i=1}^{S} K_i \left[\cos\left(u_{i,t} + \omega_{i,t}\right) + e_i \cos\left(\omega_{i,t}\right)\right] + \xi_t, \tag{36}$$

with $t = 1, \ldots, T$ and $r = 1, \ldots, R$. The number of objects in the system is S, that is consider known in this experiment (for the sake of simplicity). Both $y_{r,t}$, $u_{i,t}$ depend on time t, and then ξ_t is a Gaussian noise perturbation with variance σ_e^2. For the sake of simplicity, we consider this value known, $\sigma_e^2 = 1$. The meaning of each parameter in Eq. (36) is given in Table 5. The likelihood function is defined by (36) and some indicator variables described below. The angle $u_{i,t}$ is the true anomaly of the planet i and it can be determined from

$$\frac{du_{i,t}}{dt} = \frac{2\pi}{P_i} \frac{\left(1 + e_i \cos u_{i,t}\right)^2}{\left(1 - e_i\right)^{\frac{3}{2}}} \tag{37}$$

As mentioned above, this equation has analytical solution. As a result, the true anomaly u_t can be determined from the mean anomaly M. However, the analytical solution contains a non linear term that needs to be determined by iterating. First, we define the mean anomaly $M_{i,t}$ as

$$M_{i,t} = \frac{2\pi}{P_i} \left(t - \tau_i\right), \tag{38}$$

where τ_i is the time of periastron passage of the planet i and P_i is the period of its orbit (see Table 5). Then, through the Kepler's equation,

$$M_{i,t} = E_{i,t} - e_i \sin E_{i,t}, \tag{39}$$

where $E_{i,t}$ is the eccentric anomaly. Equation (39) has no analytic solution and it must be solved by an iterative procedure. A Newton-Raphson method is typically used to find the roots of this equation [**50**]**.** For certain sets of parameters this iterative procedure, can be particularly slow. We also have

$$\tan \frac{u_{i,t}}{2} = \sqrt{\frac{1 + e_i}{1 - e_i}} \tan \frac{E_{i,t}}{2}, \tag{40}$$

The variables $\omega_{i,t}$'s, for $i = 1, \ldots, N$, can vary with time. In particular, if the central body is much heavier than the other objects orbiting it or if the objects are very close, the so-called orbital precession is observed (e.g. [51]). The state variable $\mathbf{x}_t$ is the vector

$$\mathbf{x}_t = [V_0, K_1, \omega_{1,t}, e_1, P_1, \tau_1, \ldots, K_S, \omega_{S,t}, e_S, P_S, \tau_S], \tag{41}$$

For a single object (e.g., a planet or a natural satellite), the dimension of $\mathbf{x}_t$ is $d_X = 5+1 = 6$, with two objects the dimension of $\mathbf{x}_t$ is is $d_X = 11$ etc. Generally, we have $d_X = 1+5S$. We also include in the likelihood function the $V_0 \in [-20, 20]$, $K_i \in [0, 50]$, $e_i \in [0, 1]$, $P_i \in [0, 365]$, $\omega_{i,t} \in [0, 2\pi]$, $\tau_i \in [0, P_i]$ by means of indicator variables (i.e., the likelihood is zero outside these intervals), for all $i = 1, \ldots, S$. This means that the likelihood function is zero when the particles fall out of these intervals. Note that the interval of τ_i is conditioned to the value P_i. This parameter is the time of periastron passage, i.e. the time passed since the object passed the closest point in its orbit. It has the same units of P_i and can take values from 0 to P_i. All the Eqs. (36)–(40), jointly with the previous parameter constrains, induce a likelihood function

$$\begin{aligned} p(\mathbf{y}_{1:T}|\mathbf{x}_{1:T}) &= \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t), \\ &= \prod_{t=1}^{T}\prod_{r=1}^{R} p(y_{r,t}|\mathbf{x}_t). \end{aligned}$$

where $\mathbf{y}_t = [y_{1,t}, \ldots, y_{R,t}]^\top$. Note that all the variables $\omega_{i,t}$ vary with the time, whereas the remaining components of $\mathbf{x}_t$ are static parameters. Sophisticated particle approaches could be used, for instance, combining MCMC and particle filtering schemes for addressing the inference of both dynamic and static parameters [32, 52, 53]. The compressed particle idea can easily adapted to this scenario, or within more complicated particle algorithms. For the sake of simplicity, we leave it for future works. Here, we use the simpler approach where we consider an artificial time-evolution of the parameters [54], i.e., $\mathbf{x}_t = [V_{0,t}, K_{1,t}, \omega_{1,t}, e_{1,t}, P_{1,t}, \tau_{1,t}, \ldots, K_{S,t}, \omega_{S,t}, e_{S,t}, P_{S,t}, \tau_{S,t}]$, in order to have a prior transition equation for the entire state $\mathbf{x}_t$. We consider

$$\omega_{i,t} = \omega_{i,t-1} + v_t, \qquad t = 1, \ldots, T, \tag{42}$$

where v_t is a Gaussian noise perturbation with zero mean and variance $\sigma_V^2 = 0.5$. For the rest of parameters, we consider

$$\begin{cases} V_{0,t} = V_{0,t-1} + \xi_{0,t}, \\ K_{i,t} = K_{i,t-1} + \xi_{i,t}^{(1)}, \\ e_{i,t} = e_{i,t-1} + \xi_{i,t}^{(2)}, \\ P_{i,t} = P_{i,t-1} + \xi_{i,t}^{(3)}, \\ \tau_{i,t} = \tau_{i,t-1} + \xi_{i,t}^{(4)}, \end{cases} \tag{43}$$

for $t = 1, \dots, T$ and $\xi_{0,t}$, $\xi_{i,t}^{(j)}$, $i = 1, \dots, S$, are Gaussian noises with zero mean and variance $\sigma_\xi^2 = 0.1$. Thus, we have a Gaussian transition probability $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. The initial probability $p(\mathbf{x}_0) = \prod_{i=1}^{d_X} p(x_{i,0})$ is a product of marginal prior pdf, where $p(x_{i,0})$ is a uniform pdf for the parameter with constrains (see above) and, for the rest of parameters, $p(x_{i,0})$ is a Gaussian with zero mean and variance equal to 10. The complete posterior is

$$p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}) = \frac{1}{p(\mathbf{y}_{1:T})} p(\mathbf{y}_{1:T}|\mathbf{x}_{1:T}) \left[\prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1})\right] p(\mathbf{x}_0).$$

where $Z = p(\mathbf{y}_{1:T})$ is the complete marginal likelihood, that is also unknown. Note that to compute $Z = p(\mathbf{y}_{1:T})$ we have to integrate out all the sequence of parameters $\mathbf{x}_{0:T}$ (trajectory), .i.e.,

$$p(\mathbf{y}_{1:T}) = \int_{\mathcal{X}^{T+1}} p(\mathbf{y}_{1:T}|\mathbf{x}_{1:T}) \left[\prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1})\right] p(\mathbf{x}_0) d\mathbf{x}_{0:T}.$$

We will approximate this integral via particle filtering (i.e., sequential importance sampling with resampling steps). See [32, 33] for further details regarding the sequential estimation of Z.

6.4.2 Inferring the number of objects orbiting the central mass

Given a set of data $\{y_{r,t}\}_{r=1}^{T}$ for all time instants $t = 1, \ldots, T$ generated according to the model (see the initial parameter values below), our goal is to infer the number of objects. For this purpose, we have to approximate the model evidence $Z = p(\mathbf{y}_{1:T})$ via standard PF and the generic CPF in Table 4. For a fair comparison, we consider the same ESS approximation, $\widehat{ESS} = \frac{1}{\sum_{n=1}^{N}(\bar{w}_t^n)^2}$ and $\eta = 0.5$. In all experiments, we set $R = 5$ and $T = 50$ and average the results over 500 independent runs. We consider three different experiments: **(E1)** $S = 0$, i.e., no object, **(E2)** $S = 1$ (one object) and **(E2)** the case of two objects $S = 2$. We set $V_0 = 2$, in all cases. For the first object in **E1** and **E2**, we set $K_{1,0} = 25$, $\omega_{1,0} = 0.61$, $e_{1,0} = 0.1$, $P_{1,0} = 15$, $\tau_{1,0} = 3$. For **E2**, we also consider a second object with $K_{2,0} = 5$, $\omega_{2,0} = 0.17$, $e_{2,0} = 0.3$, $P_{2,0} = 115$, $\tau_{2,0} = 25$ (in that case $S = 2$). Note that the SNR associate to the second object is low (so that the detection of this planet is not straightforward). The rest of trajectories are generated according to the transition model (and the corresponding measurements $y_{r,t}$ according to the observation model). We consider $N = 10^5$ total number of particles and just $M = 100$ summary particles for CPF in Table 4 ($\frac{M}{N} = 10^{-3}$).

6.4.3 Results

At each run and for each experiment **E1-E2-E3**, we run the particle filters considering different state dimensions and likelihood functions (according to Eq. (36)) computing $\widehat{Z}^{(1)}$ (corresponding to "no planet"), $\widehat{Z}^{(2)}$ (corresponding to "one planet") and $\widehat{Z}^{(3)}$ (corresponding to "two objects"). Then we obtain

$$j^* = \arg \max_{j^* \in \{1,2,3\}} \widehat{Z}^{(j)}.$$

If $j^* = 0$, we decide that there is no planet/satellite. If $j^* = 1$, we decide that there is one object and if $j^* = 2$ we assume that there are 2 objects. The results are given in Tables 6–8. We compute the rate of each decision over the 500 independent runs and for each scenario **E1-E2-E3**.

Moreover, we provide the "ranking" of each decision, namely, how many times the specific decision has been the first choice, the second choice or the third choice. For instance, let us consider the decision "zero object" in Table 6: the ranking in this case is $100-0-0$ which means that 100% of cases the choice "zero object" have been the first one (i.e., with greater Bayesian evidence). In the same table, the ranking of the decision "one object" is $0-100-0$, i.e., this choice has been always the second possibility (with the second greater Bayesian evidence). We can observe that, in **E1**-**E2**, we have no loss with CPF in term of detection, since we obtain the same results of the standard PF. However, CPF requires less computational time, saving almost the 70% of the required time with the standard PF. In **E3**, CPF decides more times (67%) that there is only one object, which is an error since we have two objects in this scenario. However, also the standard PF decides 63% of times "one object". In both cases, we always decide that there is at least one object (the choice "zero object" has been never selected). Therefore, CPF provides very similar performance than a standard PF with much less computational cost.

Table 6: Experiment 1 **E1** (no planet): percentage of the decisions (over 500 runs) and the normalized computational time spent by each method.

Method		Zero	One	Two	Time
PF	decision	100%	0%	0%	1
	ranking	100-0-0	0-100-0	0-0-100	
CPF	decision	100%	0%	0%	0.32
	ranking	100-0-0	0-100-0	0-0-100	

Table 7: Experiment 2 **E2** (one planet): percentage of the decisions (over 500 runs) and the normalized computational time spent by each method.

Method		Zero	One	Two	Time
PF	decision	0%	88%	12%	1
	ranking	0-0-100	88-12-0	12-88-0	
CPF	decision	0%	88%	12%	0.32
	ranking	0-0-100	88-12-0	12-88-0	

Table 8: Experiment 3 **E3** (two objects): percentage of the decisions (over 500 runs) and the normalized computational time spent by each method.

Method		Zero	One	Two	Time
PF	decision	0%	64%	36%	1
	ranking	0-0-100	64-36-0	36-64-0	
CPF	decision	0%	67%	33%	0.32
	ranking	0-0-100	67-33-0	33-67-0	

6.5 PROSAIL inversion with time-varying physical parameters

Earth observation from satellite sensors offers the possibility to monitor our planet with unprecedented accuracy. Radiative transfer models (RTMs) are forward models that encode the energy transfer through the atmosphere, and are used to model and understand the Earth system. These models also allow us to estimate the parameters that describe the status of the Earth from satellite observations by inverse modeling. However, performing inference over such simulators is generally an ill-posed problem because of the difficulty to invert the system and to compute the marginal likelihood. Generally, RTMs are non-differentiable and computationally very costly models, which adds on a high level of difficulty in inference.

Here we will test our method for inverting a commonly used radiative transfer model for vegetation monitoring. The so-called PROSAIL RTM is the most widely used model over the last two decades in remote sensing studies [55]. It simulates reflectance as a function of:

1) A set of leaf optical properties, given by the mesophyll structural parameter (MSP), leaf chlorophyll (Chl), dry matter also referred as “leaf mass per unit area” (Cm), water (Cw), carotenoid (Car) and brown pigment (Cbr) contents.

2) A set of canopy level characteristics, determined by leaf area index (LAI), the average leaf angle inclination (ALA) and the hot-spot parameter (Hotspot). System geometry is described by the solar zenith angle (θ_s), view zenith angle θ_v), and the relative azimuth angle between both angles ($\Delta\Theta$).

In our experiments, we consider the inference of 7 of these variables, that we also assuming varying in time. The rest of parameters are keep fixed to the default values in the PROSAIL code (http://teledetection.ipgp.jussieu.fr/prosail/), so that for simplicity they are assumed known. At time instant t, our state is

$$\mathbf{x}_t = [x_{1,t}, x_{2,t}, x_{3,t}, x_{4,t}, x_{5,t}, x_{6,t}, x_{7,t}]^\top$$

where $x_{1,t} = \mathrm{Chl}(t)$, $x_{2,t} = \mathrm{Car}(t)$, $x_{3,t} = \mathrm{Cbr}(t)$, $x_{4,t} = \mathrm{Cw}(t)$, $x_{5,t} = \mathrm{Cm}(t)$, $x_{6,t} = \mathrm{MSP}(t)$, $x_{7,t} = \mathrm{LAI}(t)$. The likelihood function at time t is

$$p(\mathbf{y}_t|\mathbf{x}_t) = \exp\left(-\frac{1}{2\sigma^2}||\mathbf{y}_t - \mathbf{f}(\mathbf{x}_t)||^2\right)\mathbb{I}_{\mathcal{R}}(\mathbf{x}_t), \tag{44}$$

where $\mathbf{f}(\mathbf{x}_t) : \mathbb{R}^7 \rightarrow \mathbb{R}^{2100}$ represents the PROSAIL model, and $\mathbb{I}_{\mathcal{R}}(\mathbf{x}_t)$ is an indication function which is 1 if $\mathbf{x}_t \in \mathcal{R}$ otherwise is 0, if $\mathbf{x}_t \notin \mathcal{R}$. The region $\mathcal{R}$ is defined as $\mathcal{R} = \prod_{i=1}^{7} \mathcal{I}_i$ with $\mathcal{I}_1 = [0, 100]$ (μg/cm^2), $\mathcal{I}_2 = [0, 25]$ (μg/cm^2), $\mathcal{I}_3 = [0, 1]$, $\mathcal{I}_4 = [0, 0.05]$ (cm), $\mathcal{I}_5 = [0, 0.02]$ (g/cm^2), $\mathcal{I}_6 = [1, 3]$, $\mathcal{I}_7 = [0, 1]$. The function $\mathbf{f}(\mathbf{x}_t) = \mathrm{PROSAIL}(\mathbf{x}_t)$ is the high-nonlinear model represented by the code given at http://teledetection.ipgp.jussieu.fr/prosail/. The vector $\mathbf{y}_t \in \mathbb{R}^{2100}$ contains the measurements obtained by the satellite. The transition model is

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{\Lambda}), \tag{45}$$

where $\mathbf{\Lambda}$ is a diagonal 7×7 matrix with $\mathrm{diag}[\mathbf{\Lambda}] = [1, 0.4, 10^{-2}, 10^{-3}, 10^{-3}, 0.4, 0.4]^\top$. We recall that $t = 1, \ldots, T$. We generate synthetic data $\{\mathbf{x}_{1:T}, \mathbf{y}_{1:T}\}$ (setting $T = 20$ and $\sigma^2 = 1$) according to the model starting with $\mathbf{x}_0 = [40, 8, 0.2, 0.01, 0.009, 2.5, 0.5]^\top$. We compare a standard PF with $N = 10^4$ particles with a CPF with $M \in \{1000, 2000, 5000\}$ (and $N = 10^4$). We also consider a standard PF with $N = M$ in order to show the benefits of the compression in CPF. For all the filters we employ $\widehat{ESS} = \frac{1}{\sum_{n=1}^{N}(\bar{w}_t^n)^2}$ with $\eta = 0.5$. Figure 5 depicts the data $\mathbf{y}_t$ at $t = 15$ (solid line) and the model values corresponding to 50 particles $\mathbf{f}^{(i)} = \mathbf{f}(\mathbf{x}_t^{(i)})$ (as an example). We compute the Root Mean Square Error (RMSE) in the estimation of the trajectory of parameters $\mathbf{x}_{1:T}$. The RMSE is obtained by averaging the square errors over all the component of the state, and over each time. We have averaged the results over 10^3 independent runs. The results are given in Table 9. We can observe that CPF provides very similar results than the standard PF with $N = 10^4$ with much less computational cost. For instance, CPF with $M = 10^3$ saves more than 95% of the computational time with an increase of the MSE of only 5%. The comparison between CPF and the standard PFs with $N = M \in \{1000, 2000, 5000\}$ shows the benefit of the compression in CPF.

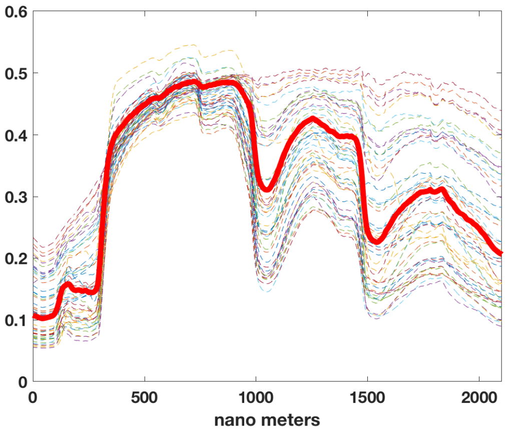


Figure 5: The data $\mathbf{y}_t$ (solid line) and the model values corresponding to 50 particles, $\mathbf{f}^{(i)} = \mathbf{f}(\mathbf{x}_t^{(i)})$ (dashed lines), at $t = 15$.

Table 9: Results of the PROSAIL inversion.

Method	RMSE	Norm. TIME
Standard PF - $N = 10^4$	3.99	1
CPF - $M = 10^3$	4.08	0.0112
CPF - $M = 2 \cdot 10^3$	4.04	0.1938
CPF - $M = 5 \cdot 10^3$	3.99	0.4215
Standard PF - $N = 10^3$	4.30	0.0984
Standard PF - $N = 2 \cdot 10^3$	4.19	0.1856
Standard PF - $N = 5 \cdot 10^3$	4.10	0.4181

7 Conclusions

We have introduced a novel approach for performing sequential Bayesian inference in the context of complex and costly models. In the proposed scheme, the expensive model is evaluated only in some well-chosen samples. The selection of these nodes is based on the so-called compressed Monte Carlo (CMC) scheme. The application of CMC within particle filtering schemes and the corresponding benefits are described and discussed. The provided theoretical and numerical results, which include applications in astronomy and remote sensing, showed the advantages of the proposed method.

Acknowledgements

LM acknowledges support by the Agencia Estatal de Investigación AEI (project PID2019-105032GB-I00). JL-S acknowledges support by the Office of Naval Research (N00014-19-1-2226), Spanish Ministry of Science, Innovation and Universities (RTI2018-099655-B-I00) and Regional Ministry of Education and Research for the Community of Madrid (Y2018/TCS-4705).

References

[1] T. Santner, B. Williams, and W. Notz, *The design and analysis of computer experiments.* Springer Verlag, 2003.

[2] B. Wescott, *Every Computer Performance Book*, 1st ed. USA: CreateSpace Independent Publishing Platform, 2013.

[3] M. Raissi, P. Perdikaris, and G. Karniadakis, "Machine learning of linear differential equations using Gaussian processes," *Journal of Computational Physics*, vol. 348, pp. 683–693, Aug 2017.

[4] A. Sandberg, "Feasibility of whole brain emulation," in *Philosophy and theory of artificial intelligence.* Springer, 2013, pp. 251–264.

[5] J. Verrelst, N. Sabater, J. Rivera, J. Muñoz Marí, J. Vicent, G. Camps-Valls, and J. Moreno, "Emulation of leaf, canopy and atmosphere radiative transfer models for fast global sensitivity analysis," *Remote Sensing*, vol. 8(8), 673, no. 8, 2016.

[6] A. J. Majda and B. Gershgorin, “Quantifying uncertainty in climate change science through empirical information theory,” *Proceedings of the National Academy of Sciences*, vol. 107, no. 34, pp. 14 958–14 963, 2010. [Online]. Available: https://www.pnas.org/content/107/34/14958

[7] P. C. Young and M. Ratto, “Statistical emulation of large linear dynamic models,” *Technometrics*, vol. 53, no. 1, pp. 29–43, 2011.

[8] S. Conti and A. OHagan, “Bayesian emulation of complex multi-output and dynamic computer models,” *Journal of statistical planning and inference*, vol. 140, no. 3, pp. 640–651, 2010.

[9] S. Koziel, S. Ogurtsov, and L. Leifsson, “Physics-based surrogates for low-cost modeling of microwave structures,” *Procedia Computer Science*, vol. 18, pp. 869–878, 2013.

[10] S. Koziel and L. Leifsson, “Multi-level cfd-based airfoil shape optimization with automated low-fidelity model selection,” *Procedia Computer Science*, vol. 18, pp. 889–898, 2013.

[11] K. Cutajar, M. Pullin, A. Damianou, N. Lawrence, and J. González, “Deep Gaussian processes for multi-fidelity modeling,” *arXiv preprint arXiv:1903.07320*, 2019.

[12] A. O’Hagan, “Bayesian analysis of computer code outputs: A tutorial,” *Reliability Engineering and System Safety*, vol. 91, no. 10-11, pp. 1290–1300, 2006.

[13] J. S. Liu, *Monte Carlo Strategies in Scientific Computing.* Springer, 2004.

[14] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods.* Springer, 2004.

[15] D. Busby, “Hierarchical adaptive experimental design for Gaussian process emulators,” *Reliability Engineering and System Safety*, vol. 94, pp. 1183–1193, 2009.

[16] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, and K. Crombecq, “A surrogate modeling and adaptive sampling toolbox for computer based design,” *J. Mach. Learn. Res.*, vol. 11, pp. 2051–2055, 2010.

[17] Z. Wang, S. Yan, and C. Zhang, “Active learning with adaptive regularization,” *Pattern Recognition*, vol. 44, pp. 2375–2383, 2011.

[18] D. Svendsen, L. Martino, and G.Camps-Valls, “Active emulation of computer codes with Gaussian processes - application to remote sensing,” *Pattern Recognition*, vol. 100, p. 107103, 2020.

[19] M. A. Beaumont, “Approximate Bayesian computation in evolution and ecology,” *Annual Review of Ecology, Evolution, and Systematics*, vol. 41, no. 1, pp. 379–406, 2010.

[20] U. Simola, J. Cisewski-Kehe, M. U. Gutmann, and J. Corander, “Adaptive Bayesian computation tolerance selection,” *arXiv:1907.01505*, pp. 1–29, 2019.

[21] B. M. Turner and T. V. Zandt, “A tutorial on approximate Bayesian computation,” *Journal of Mathematical Psychology*, vol. 56, no. 2, pp. 69 – 85, 2012.

[22] A. Owen, *Monte Carlo theory, methods and examples.* http://statweb.stanford.edu/∼owen/mc/, 2013.

[23] I. Arasaratnam and S. Haykin, “Cubature Kalman filters,” *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[24] S. J. Julier and J. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, vol. 92, no. 2, pp. 401–422, March 2004.

[25] F. Huszár and D. Duvenaud, “Optimally-weighted herding is Bayesian quadrature,” *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence (UAI-12)*, pp. 377–386, 2012.

[26] S. Särkkä, *Bayesian Filtering and Smoothing.* Cambridge University Press, 2013.

[27] L. Martino and V. Elvira, “Compressed Monte Carlo for distributed Bayesian inference,” *viXra:1811.0505*, pp. 1–14, 2018.

[28] V. Elvira, L. Martino, and P. Closas, “Importance Gaussian quadrature,” *arXiv:2001.03090*, pp. 1–13, 2020.

[29] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric, “Adaptive importance sampling: The past, the present, and the future,” *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 60–79, 2017.

[30] L. Martino, V. Elvira, and M. F. Louzada, “Effective Sample Size for importance sampling based on the discrepancy measures,” *Signal Processing*, vol. 131, pp. 386–401, 2017.

[31] M. Bolić, P. M. Djurić, and S. Hong, “Resampling algorithms for particle filters: A computational complexity perspective,” *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 15, pp. 2267–2277, November 2004.

[32] L. Martino, V. Elvira, and G. Camps-Valls, “Group Importance Sampling for Particle Filtering and MCMC,” *Digital Signal Processing*, vol. 82, pp. 133–151, 2018.

[33] L. Martino, V. Elvira, and F. Louzada, “Weighting a resampled particle in Sequential Monte Carlo,” *IEEE Statistical Signal Processing Workshop, (SSP)*, vol. 122, pp. 1–5, 2016.

[34] P. LEcuyer, “Efficiency improvement and variance reduction,” *In Proceedings of the 1994 Winter Simulation Conference*, pp. 122–132, 1994.

[35] L. Martino, D. Luengo, and J. Miguez, *Independent Random Sapling methods.* Springer, 2018.

[36] J. Kotecha and P. M. Djurić, “Gaussian particle filtering,” *IEEE Transactions Signal Processing*, vol. 51, no. 10, pp. 2592–2601, October 2003.

[37] ——, “Gaussian sum particle filtering,” *IEEE Transactions Signal Processing*, vol. 51, no. 10, pp. 2602–2612, October 2003.

[38] T. Li, T. P. Sattar, and S. Sun, “Deterministic resampling: Unbiased sampling to avoid sample impoverishment in particle filters,” *Signal Processing*, vol. 92, no. 7, pp. 1637–1645, 2012.

[39] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[40] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter.* Boston: Artech House, 2004.

[41] M. K. Pitt and N. Shephard, “Auxiliary variable based particle filters,” in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. Springer, 2001, ch. 13, pp. 273–293.

[42] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice.* New York: Springer, 2001.

[43] P. C. Gregory, “Bayesian re-analysis of the Gliese 581 exoplanet system,” *Monthly Notices of the Royal Astronomical Society*, vol. 415, no. 3, pp. 2523–2545, Aug. 2011.

[44] S. C. C. Barros, D. J. A. Brown, G. Hébrard, Y. Gómez Maqueo Chew, D. R. Anderson, P. Boumis, L. Delrez, K. L. Hay, K. W. F. Lam, J. Llama, M. Lendl, J. McCormac, B. Skiff, B. Smalley, O. Turner, M. Vanhuysse, D. J. Armstrong, I. Boisse, F. Bouchy, A. Collier Cameron, F. Faedi, M. Gillon, C. Hellier, E. Jehin, A. Liakos, J. Meaburn, H. P. Osborn, F. Pepe, I. Plauchu-Frayn, D. Pollacco, D. Queloz, J. Rey, J. Spake, D. Ségransan, A. H. M. Triaud, S. Udry, S. R. Walker, C. A. Watson, R. G. West, and P. J. Wheatley, “WASP-113b and WASP-114b, two inflated hot Jupiters with contrasting densities,” *Astronomy and Aastrophysics*, vol. 593, p. A113, 2016.

[45] L. Affer, M. Damasso, G. Micela, E. Poretti, G. Scand ariato, J. Maldonado, A. F. Lanza, E. Covino, A. Garrido Rubio, J. I. González Hernández, R. Gratton, G. Leto, A. Maggio, M. Perger, A. Sozzetti, A. Suárez Mascareño, A. S. Bonomo, F. Borsa, R. Claudi, R. Cosentino, S. Desidera, P. Giacobbe, E. Molinari, M. Pedani, M. Pinamonti, R. Rebolo, I. Ribas, and B. Toledo-Padrón, “HADES RV program with HARPS-N at the TNG. IX. A super-Earth around the M dwarf Gl 686,” *arXiv:1901.05338*, vol. 622, p. A193, Feb. 2019.

[46] T. Trifonov, S. Stock, T. Henning, S. Reffert, M. Kürster, M. H. Lee, B. Bitsch, R. P. Butler, and S. S. Vogt, "Two Jovian Planets around the Giant Star HD 202696: A Growing Population of Packed Massive Planetary Pairs around Massive Stars?" *The Astronomical Journal*, vol. 157, no. 3, p. 93, Mar. 2019.

[47] K. G. Strassmeier, D. S. Hall, F. C. Fekel, and M. Scheck, "A catalog of chromospherically active binary stars (second edition)," *Astronomy and Astrophysics Supplement Series*, vol. 100, pp. 173–225, 1993.

[48] M. C. Gálvez, D. Montes, M. J. Fernández-Figueroa, and J. López-Santiago, "Chromospheric Activity and Orbital Solution of Six New Late-type Spectroscopic Binary Systems," *Astrophysics and Space Science*, vol. 304, no. 1-4, pp. 59–61, Aug. 2006.

[49] S. Gillessen, P. M. Plewa, F. Eisenhauer, R. Sari, I. Waisberg, M. Habibi, O. Pfuhl, E. George, J. Dexter, S. von Fellenberg, T. Ott, and R. Genzel, "An Update on Monitoring Stellar Orbits in the Galactic Center," *The Astrophysical Journal*, vol. 837, no. 1, p. 30, 2017.

[50] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C++ : the art of scientific computing*. Springer, 2002.

[51] L.-J. Li and S.-B. Qian, "Observations and Orbital Analysis of the High-Amplitude Delta Scuti Star SZLyncis: The Unusual Orbital Precession," *Publications of the Astronomical Society of Japan*, vol. 65, p. 116, Dec. 2013.

[52] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods," *J. R. Statist. Soc. B*, vol. 72, no. 3, pp. 269–342, 2010.

[53] L. Martino, "A review of multiple try MCMC algorithms for signal processing," *Digital Signal Processing*, vol. 75, pp. 134 – 152, 2018.

[54] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. Springer, 2001, ch. 10, pp. 197–223.

[55] S. Jacquemoud, W. Verhoef, F. Baret, C. Bacour, P. Zarco-Tejada, G. Asner, C. François, and S. Ustin, "PROSPECT+ SAIL models: A review of use for vegetation characterization," *Remote sensing of environment*, vol. 113, pp. S56–S66, 2009.

A Proof of Theorem 1

First of all, we need to obtain some additional relationships. Let us define the partial estimators as

$$\widehat{I}_m(h) = \sum_{i\in\mathcal{J}_m} \bar{w}_{m,i} h(\mathbf{x}_i), \tag{46}$$

an estimator of the integral

$$\int_{\mathcal{X}_m} h(\mathbf{x})\bar{\pi}(\mathbf{x})d\mathbf{x} = \int_{\mathcal{X}} h(\mathbf{x})\bar{\pi}_m(\mathbf{x})d\mathbf{x}, \tag{47}$$

where we have set $\bar{\pi}_m(\mathbf{x}) = \bar{\pi}(\mathbf{x})\mathbb{I}(\mathcal{X}_m)$. with these definitions, note a that

$$\begin{aligned}
\widehat{I}^{(N)}(h) = \sum_{i=1}^{N} \bar{w}_i h(\mathbf{x}_i) &= \sum_{m=1}^{M}\sum_{i\in\mathcal{J}_m} \bar{w}_i h(\mathbf{x}_i), \\
&= \sum_{m=1}^{M} \widehat{a}_m \sum_{i\in\mathcal{J}_m} \bar{w}_{m,i} h(\mathbf{x}_i) \\
&= \sum_{m=1}^{M} \widehat{a}_m \widehat{I}_m(h),
\end{aligned} \tag{48}$$

where we have used $\bar{w}_{m,i} = \frac{\bar{w}_i}{\widehat{a}_m}$ as shown in Eq. (18). Namely, the estimator $\widehat{I}^{(N)}(h)$ of $I(h)$ can be expressed as a convex combination of the M partial estimators. A similar expression is valid for the particle approximations, i.e.,

$$\widehat{\pi}^{(N)}(\mathbf{x}) = \sum_{m=1}^{M} \widehat{a}_m \widehat{\pi}_m(\mathbf{x}), \quad \text{where} \tag{49}$$

$$\widehat{\pi}_m(\mathbf{x}) = \sum_{i\in\mathcal{J}_m} \bar{w}_{m,i} \delta(\mathbf{x}-\mathbf{x}_i). \tag{50}$$

Proof. Assume that $\mathcal{S} = \{\mathbf{x}_n, \bar{w}_n\}_{n=1}^N$ (hence also N) and the partition $\mathcal{P}$ are given and fixed (hence M as well). Then, the summary weights $\widehat{a}_m$ are also fixed. The unique stochastic part in $\widetilde{I}^{(M)}(h)$ is the selection of $\mathbf{s}_m$'s. Let us consider the case when $\mathbf{s}_m$ is resampled randomly in each partition, according to the weights $\bar{w}_{m,j}$ in Eq. (18), i.e.,

$$\mathbf{s}_m \sim \widehat{\pi}_m(\mathbf{x}).$$

Given the set of weighted samples $\mathcal{S} = \{\mathbf{x}_n, \bar{w}_n\}_{n=1}^N$, note that

$$\mathrm{E}[h(\mathbf{s}_m)|\mathcal{S}] = \sum_{j\in\mathcal{J}_m} \bar{w}_{m,j} h(\mathbf{x}_j) = \widehat{I}_m(h). \tag{51}$$

Given Eq. (48), we can also write

$$\widehat{I}^{(N)}(h) = \sum_{m=1}^{M} \widehat{a}_m \widehat{I}_m(h) = \sum_{m=1}^{M} \widehat{a}_m \mathrm{E}[h(\mathbf{s}_m)|\mathcal{S}]. \tag{52}$$

Note also that

$$\widetilde{I}^{(M)}(h) = \sum_{m=1}^{M} \widehat{a}_m h(\mathbf{s}_m),$$

Taking the expectation of both sides

$$\begin{aligned}
\mathrm{E}[\widetilde{I}^{(M)}(h)|\mathcal{S}] &= \mathrm{E}\left[\sum_{m=1}^{M} \widehat{a}_m h(\mathbf{s}_m)\right] \\
&= \sum_{m=1}^{M} \widehat{a}_m \mathrm{E}[h(\mathbf{s}_m)|\mathcal{S}], \\
&= \widehat{I}^{(N)}(h),
\end{aligned}$$

where we have used Eq. (52).

B Proof of Theorem 3

Theorem 3 states that, with the choice $s_m = \sum_{j\in\mathcal{J}_m} \bar{w}_{m,j} h(\mathbf{x}_j)$ in (21), we have $\widetilde{I}^{(M)}(f) \equiv \widehat{I}^{(N)}(h)$, for a specific function $h(\mathbf{x})$ and $f(\mathbf{x}) = \mathbf{x}$. Indeed, we have

$$\begin{aligned}
\widetilde{I}^{(M)}(f) &= \sum_{m=1}^{M} \widehat{a}_m s_m \\
&= \sum_{m=1}^{M} \widehat{a}_m \left[\sum_{j\in\mathcal{J}_m} \bar{w}_{m,j} h(\mathbf{x}_j)\right],
\end{aligned}$$

where we have used $s_m = \sum_{j \in \mathcal{J}_m} \bar{w}_{m,j} h(\mathbf{x}_j)$. Replacing $\bar{w}_{m,j}$ with the expression $\bar{w}_{m,j} = \bar{w}_j / \widehat{a}_m$ in Eq. (18), we obtain a further simplification,

$$\begin{aligned} \widetilde{I}^{(M)}(f) &= \sum_{m=1}^{M} \widehat{a}_m \left[\sum_{j \in \mathcal{J}_m} \frac{\bar{w}_j}{\widehat{a}_m} h(\mathbf{x}_j) \right], \\ &= \sum_{m=1}^{M} \sum_{j \in \mathcal{J}_m} \bar{w}_j h(\mathbf{x}_j) \\ &= \sum_{j=1}^{N} \bar{w}_j h(\mathbf{x}_j) = \widehat{I}^{(N)}(h), \end{aligned} \tag{53}$$

where we have also used the fact that, if we consider all the M possible sums within $\mathcal{J}_m$, then we are considering all the possible N samples and weights, i.e., $\sum_{m=1}^{M} \sum_{j \in \mathcal{J}_m} \bar{w}_j h(\mathbf{x}_j) = \sum_{j=1}^{N} \bar{w}_j h(\mathbf{x}_j)$.