

Unbiased and Consistent Nested Sampling via Sequential Monte Carlo

Robert Salomone*
Leah F. South †
Christopher C. Drovandi ‡
Dirk P. Kroese §

Abstract

We introduce a new class of sequential Monte Carlo methods called *Nested Sampling via Sequential Monte Carlo* (NS-SMC), which reframes the Nested Sampling method of Skilling (2006) in terms of sequential Monte Carlo techniques. This new framework allows one to obtain provably consistent estimates of marginal likelihood and posterior inferences when Markov chain Monte Carlo (MCMC) is used to produce new samples. An additional benefit is that marginal likelihood estimates are also unbiased. In contrast to NS, the analysis of NS-SMC does not require the (unrealistic) assumption that the simulated samples be independent. As the original NS algorithm is a special case of NS-SMC, this provides insights as to why NS seems to produce accurate estimates despite a typical violation of its assumptions. For applications of NS-SMC, we give advice on tuning MCMC kernels in an automated manner via a preliminary pilot run, and present a new method for appropriately choosing the number of MCMC repeats at each iteration. Finally, a numerical study is conducted where the performance of NS-SMC and temperature-annealed SMC is compared on several challenging and realistic problems. MATLAB code for our experiments is made available at <https://github.com/LeahPrice/SMC-NS>.

Keywords: Bayesian computation, marginal likelihood, posterior inference, Markov chain Monte Carlo, estimation of normalizing constants

*School of Mathematics and Physics, The University of Queensland, Australia
email: robert.salomone@uqconnect.edu.au

†Queensland University of Technology, Australia

‡Queensland University of Technology, Australia

§School of Mathematics and Physics, The University of Queensland, Australia

1 INTRODUCTION

A canonical problem in the computational sciences is the estimation of integrals of the form

$$\pi(\varphi) = \mathbb{E}_\pi[\varphi(\mathbf{X})] = \int_E \varphi(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}, \quad (1)$$

where π is a probability density on $E \subseteq \mathbb{R}^d$ and $\varphi : E \rightarrow \mathbb{R}$ is a π -integrable function. Note the “overloading” of notation for $\pi(\cdot)$, depending on whether the argument is a function φ or a vector \mathbf{x} . In Bayesian computation, which is the focus of this work, $\pi(\mathbf{x})$ is typically known only up to a normalizing constant, that is, $\pi(\mathbf{x}) = \gamma(\mathbf{x})/\mathcal{Z}$ for some known positive function γ which, in turn, decomposes into a product $\eta\mathcal{L}$, where η is another probability density function. In particular, in this setting, π is the *posterior* probability density, η is the *prior* probability density, \mathcal{L} the *likelihood* function, and $\mathbf{x} \in E$ represents a *parameter*. For clarity, the correspondence to the usual Bayesian notation (which is typically written in terms of parameter $\boldsymbol{\theta}$ and data \mathcal{D}) is as follows:

$$\underbrace{p(\boldsymbol{\theta} | \mathcal{D})}_{\pi(\mathbf{x})} \propto \underbrace{p(\boldsymbol{\theta})}_{\eta(\mathbf{x})} \underbrace{p(\mathcal{D} | \boldsymbol{\theta})}_{\mathcal{L}(\mathbf{x})}. \quad (2)$$

Even though π , η and \mathcal{L} are general functions, and do not have to be interpreted in terms of Bayesian computation, we henceforth refer to them as posterior, prior and likelihood function, respectively. Another quantity of interest is the normalizing constant

$$\mathcal{Z} = \int_E \eta(\mathbf{x})\mathcal{L}(\mathbf{x})d\mathbf{x}, \quad (3)$$

which, in the Bayesian context, is called the *marginal likelihood* (or model evidence) and is often used in model selection.

The most popular methodology for estimating (1) is to use *Markov Chain Monte Carlo* (MCMC). Here, an ergodic Markov chain with π as its invariant density is simulated, yielding samples approximately from π after a suitably long duration known as the burn-in period. The empirical distribution of these samples can then be used to estimate (1). For more details, see [Robert and Casella \(2004, Chapters 6–12\)](#).

Nested Sampling (NS) [Skilling \(2006\)](#) is a Monte Carlo/numerical quadrature method proposed initially for the estimation of marginal likelihoods, which also provides estimates of $\mathbb{E}_\pi[\varphi(\mathbf{X})]$ without requiring additional likelihood evaluations. The method is based on a sampling scheme that samples from progressively constrained (nested) versions of the prior. NS has achieved wide-spread acceptance as a tool for Bayesian computation in certain fields,

being particularly popular in astronomy (see for example [Vegetti and Koopmans \(2009\)](#) and [Veitch \(2015\)](#)) and more generally as a computational method in physics (examples here include [Baldock \(2017\)](#), [PÁartay et al \(2014\)](#), and [Murray et al \(2005\)](#)). However, NS has failed to achieve popularity more broadly in the statistical community, largely owing to a variety of theoretical problems, most notable of which is that the methodology assumes that one can obtain perfect and independent samples from constrained versions of the prior at each iteration, which is clearly unrealistic.

On the other hand, *Sequential Monte Carlo* (SMC) is a general methodology that involves traversing a population of particles through a sequence of distributions, using a combination of mutation, correction, and selection steps. SMC has a rich theoretical basis, as it can be analyzed through interacting particle approximations to a flow of Feynman-Kac measures, see for example the technical monograph [Del Moral \(2013\)](#), or the tutorial [Del Moral and Doucet \(2014\)](#). The use of SMC methodology in a statistical setting began with the “Bootstrap Particle Filter” of [Gordon et al \(1993\)](#) for online inference in hidden Markov models, and has been the topic of much research in the statistical community (see for example, the survey [Doucet and Johansen \(2011\)](#)). However, SMC methods in general date much further back to the *multilevel splitting* method of [Kahn and Harris \(1951\)](#) for the estimation of rare-event probabilities. An overview of splitting techniques can be found in [Rubinstein and Kroese \(2017, Chapter 9\)](#), and such methods have continued to be active topic for research, see for example [Botev and Kroese \(2012\)](#), [C erou et al \(2012\)](#), and [C erou and Guyader \(2016\)](#).

The special case of SMC where all sampling distributions live on the same space E is discussed in [Del Moral et al \(2006\)](#). In this setting, one can sample from an arbitrary density π by introducing an artificial sequence of densities bridging from an easy to sample distribution (say η) to π . This approach is often referred to as SMC in the *static* setting. While static SMC samplers often make use of MCMC moves, they possess advantages over the pure MCMC approach in that they are naturally parallelizable, can cope with complicated posterior landscapes such as those containing multimodality, and have the added benefit of being able to produce consistent (and unbiased) estimates of the marginal likelihood as a byproduct.

The aim of this paper is to explore the connection between NS and SMC samplers, resolve some long-standing theoretical issues with NS by placing it in the SMC framework, and demonstrate not only how the resulting algorithm can be implemented effectively in practice, but also that it is able to obtain similar quality of results to existing SMC approaches under similar conditions on highly challenging examples.

To those ends, the contributions of this work are as follows:

1. We show that by implementing a special type of SMC sampler that takes two importance sampling paths at each iteration, one obtains an analogous SMC method to NS that resolves its main theoretical and practical issues. Most notably, the consistency of estimates of marginal likelihood and posterior inferences with our algorithm is easily established from the properties of SMC methods, and does not rely on obtaining perfect independent samples. Moreover, estimates of the marginal likelihood are unbiased.
2. We introduce an improved version of NS, of which the original NS method can be interpreted as a “rough” version. This gives insights as to why NS seems to work in practice when samples are dependent, despite the original formulation of the method requiring independent samples.
3. We provide recommendations on how to ensure robust performance of SMC samplers in practice, including how to tune MCMC kernels and determine an appropriate amount of MCMC repeats.
4. Using these techniques, we present the first extensive comparison between the popular temperature–annealed SMC approach and our NS–SMC approach, for both the purpose of marginal likelihood estimation and posterior sampling on difficult realistic statistical problems.
5. Having demonstrated that the ideas behind NS find their true home within SMC methodology, we conclude by discussing the variety of theoretical and methodological avenues of possible future research.

2 NESTED SAMPLING

Nested Sampling (NS) (Skilling, 2006) is based on the identity

$$\mathcal{Z} = \int_E \eta(\mathbf{x}) \mathcal{L}(\mathbf{x}) \, d\mathbf{x} = \mathbb{E}_\eta[\mathcal{L}(\mathbf{X})] = \int_0^\infty \mathbb{P}(\mathcal{L}(\mathbf{X}) > l) \, dl, \quad (4)$$

where \mathcal{L} is a function mapping from some space E to \mathbb{R}_+ , and $\mathbf{X} \sim \eta$. Note that $\mathbb{P}(\mathcal{L}(\mathbf{X}) > l)$ is simply the tail cdf (survival function) of the random variable $\mathcal{L}(\mathbf{X})$. We denote this survival function by $\bar{F}_{\mathcal{L}(\mathbf{X})}$. A simple inversion argument yields

$$\int_0^\infty \bar{F}_{\mathcal{L}(\mathbf{X})}(l) \, dl = \int_0^1 \bar{F}_{\mathcal{L}(\mathbf{X})}^{-1}(p) \, dp, \quad (5)$$

where $\bar{F}_{\mathcal{L}(\mathbf{X})}^{-1}(p)$ is the $(1-p)$ -quantile function of the likelihood under η . This simple one-dimensional representation suggests that if one had access to the function $\bar{F}_{\mathcal{L}(\mathbf{X})}^{-1}$, the integral could then be approximated by numerical methods. For example, for a discrete set of values, $0 < p_T < \dots < p_1 < p_0 = 1$, one could compute the Riemann sum

$$\sum_{t=1}^T (p_t - p_{t-1}) \bar{F}_{\mathcal{L}(\mathbf{X})}^{-1}(p_t), \quad (6)$$

as a (deterministic) approximation of \mathcal{Z} . Unfortunately, the quantile function of interest is often intractable. NS provides an approximate way of performing quadrature such as (6). The core insight underlying NS is as follows. For N independent samples $\mathbf{X}_1, \dots, \mathbf{X}_N$ from a density of the form

$$\eta(\mathbf{x}; l) := \frac{\eta(\mathbf{x}) \mathbb{I}\{\mathcal{L}(\mathbf{x}) > l\}}{\mathbb{P}_\eta(\mathcal{L}(\mathbf{X}) > l)}, \quad \mathbf{x} \in E, \quad l \in \mathbb{R}_+, \quad (7)$$

we have that

$$\frac{\bar{F}_{\mathcal{L}(\mathbf{X})}(\min_k \mathcal{L}(\mathbf{X}^k))}{\bar{F}_{\mathcal{L}(\mathbf{X})}(l)} \sim \text{Beta}(N, 1). \quad (8)$$

Put simply, consider that one has N independent samples distributed according to the prior subject to a given likelihood constraint, and then introduces a new constraint determined by choosing the minimum likelihood value of the samples. This will define a region that has less (unconstrained) prior probability by a factor that has a $\text{Beta}(N, 1)$ distribution. As samples from this new distribution will be compressed into a smaller region of the original prior, (8) is often referred to as a *compression factor*.

With this in mind, Skilling (2006) proposes the NS procedure that proceeds as follows. Initially, a population of N independent samples (henceforth called particles) are drawn from η . Then, for each iteration $t = 1, \dots, T$, the particle with the smallest value of \mathcal{L} is identified. This “worst performing” particle at iteration t is denoted by $\check{\mathbf{X}}_t$ and its likelihood value by L_t . Finally, this particle is moved to a new position that is determined by drawing a sample according to $\eta(\cdot; L_t)$. By construction, this procedure results in a population of samples from η that is constrained to lie above higher values of \mathcal{L} at each iteration.

After T iterations, we then have $\{L_t\}_{t=1}^T$. Each L_t corresponds to an unknown p_t satisfying $L_t = \bar{F}_{\mathcal{L}(\mathbf{X})}^{-1}(p_t)$. Skilling proposes to (deterministically) approximate the p_t values by assuming that at each iteration the compression factor (8) is equal to its *geometric mean*, i.e., $\exp(\mathbb{E} \log(C)) = \exp(-1/N)$. Thus, we have the approximation $p_t = \exp(-t/N)$. This is the most popular implementation, and thus will be the version we consider for the remainder of this paper; however, it is worth noting that there exists another variant which randomly assigns $p_{t+1} = p_t B_t$ at each iteration, where $B_t \sim \text{Beta}(N, 1)$. With the pairs

$(L_t, p_t)_{t=1}^T$ in hand, the numerical integration is then of the form

$$\widehat{\mathcal{Z}} = \sum_{t=1}^T \underbrace{(p_t - p_{t-1})L_t}_{\widehat{\mathcal{Z}}_t}. \quad (9)$$

In practice, the number of iterations T is not set in advance, but rather the iterative sampling procedure is repeated until some termination criterion is satisfied. The standard approach is to continue until $p_t \cdot \max_{1 \leq j \leq N} \mathcal{L}(\mathbf{X}^j) < \epsilon \sum_{j=1}^t \widehat{\mathcal{Z}}_j$, where ϵ is some small value, say 10^{-8} . This choice attempts to ensure that the remaining integral is sufficiently small so that error arising from omission of the final $[0, p_T]$ in the quadrature is negligible.

In addition to estimates of the model evidence \mathcal{Z} , estimates of posterior expectations $\mathbb{E}_\pi[\varphi(\mathbf{X})]$, as in (1), can be obtained by assigning to each $\check{\mathbf{X}}_t$ the weight $w_t = \widehat{\mathcal{Z}}_t$, and using

$$\sum_{t=1}^T \varphi(\check{\mathbf{X}}_t) w_t / \sum_{t=1}^T w_t, \quad (10)$$

as an estimator. A formal justification for this is given in [Chopin and Robert \(2010, Section 2.2\)](#), though in essence it is based on the fact that the numerator and denominator of (10) are (NS) estimators of their corresponding terms in the identity

$$\mathbb{E}_\pi[\varphi(\mathbf{X})] = \int_E \eta(\mathbf{x}) \mathcal{L}(\mathbf{x}) \varphi(\mathbf{x}) d\mathbf{x} / \int_E \eta(\mathbf{x}) \mathcal{L}(\mathbf{x}) d\mathbf{x}. \quad (11)$$

Pseudocode for NS is provided in [Algorithm 1](#).

Algorithm 1: Nested Sampling

input : population size N

$t \leftarrow 0$

for $k = 1$ **to** N **do** draw $\mathbf{X}^k \sim \eta$

while (not terminate) **do**

$t \leftarrow t + 1$

$m \leftarrow \operatorname{argmin}_{1 \leq k \leq N} \mathcal{L}(\mathbf{X}^k)$ // identify worst-performing sample

$L_t \leftarrow \mathcal{L}(\mathbf{X}^m)$

$w_t \leftarrow (\exp(-(t-1)/N) - \exp(-t/N))L_t$

$\check{\mathbf{X}}_t \leftarrow \mathbf{X}^m$ // save sample for inference

$\mathbf{X}^m \leftarrow$ a sample from $\eta(\cdot; L_t)$ // move worst-performing particle

end

$T \leftarrow t$

return estimator of the evidence $\widehat{\mathcal{Z}} = \sum_{t=1}^T w_t$ and weighted samples $\{\check{\mathbf{X}}_t, w_t\}_{t=1}^T$.

While the estimator (10) bears some resemblance to importance sampling (which is intro-

duced in Section 3) in its use of a ratio estimator and weighted samples, it is not precisely the same.

2.1 Why isn't nested sampling more popular with statisticians?

There are several potential issues with NS that we speculate are the reasons why it has not achieved mainstream adoption in the statistics community. We outline what we believe to be the main four objections to NS below (in decreasing order of severity), as well as a discussion on relevant works that have attempted to address them.

1. **Assumption of Independent Samples.** The property (8) requires at each iteration independent samples with the correct distribution. This is a strong condition, as generating samples from constrained densities of the form (7) is in general difficult. The sampling method originally proposed is to move the worst performing particle at each iteration to the position of one of the other particles, and then run an MCMC algorithm for sufficiently many iterations to create an (approximately) independent sample. This procedure itself does not ensure the assumption of independence is satisfied, as it only produces independent samples asymptotically in the number of MCMC iterations. Moreover, for problems with likelihoods that have multiple well-separated modes, the constrained density will have increasingly isolated islands of support as the algorithm progresses, making it difficult for most samplers to cross between modes in any reasonable amount of time. Thus, even *approximate* independence may be difficult to achieve (and verify) in practice.

Indeed, now over a decade after the introduction of the NS method, establishing consistency when MCMC transitions are used for sampling with NS remains a challenging open problem. [Chopin and Robert \(2010\)](#) remark that “a reason why such a theoretical result seems difficult to establish is that each iteration involves both a different MCMC kernel and a different invariant distribution”. In order to overcome the need for MCMC sampling, they propose a variant of NS for which the sampling can be performed exactly, and that demonstrate it can perform well in low dimensional problems for which π is approximately Gaussian.

In a separate attempt to overcome dependency between samples, there is a class of approximate sampling methods called *region sampling* that attempts to generate independent samples by reparameterizing the problem so the constrained sampling problem is one of sampling uniformly within constrained regions of a unit hypercube. The most popular of these methods is MultiNest ([Feroz and Hobson \(2008\)](#)), which uses the population of particles to construct a region that is a union of hyperellipsoids,

sampling from this region, and accepting samples which satisfy the constraint. There is no way however to ensure the proposal region is a superset of the *actual* region. [Buchner \(2016\)](#) proposes a method that is more robust (but still not immune) to this problem; however, the results show it can be an order of magnitude more inefficient and is more susceptible to the curse of dimensionality.

2. **Effect of Quadrature on Posterior Inferences.** As shown in (10), the ratio of two NS estimators (from a single run) can be used for posterior inferences. However, the precise effect of the use of quadrature in both estimators on estimates of $\pi(\varphi)$ is not well understood. The algorithm replaces the integral of \mathcal{L} over a (random) shell $\{\mathbf{x} \in E : L_t < \mathcal{L}(\mathbf{x}) < L_{t+1}\}$ with a single value, and assigns a volume to that shell according to a geometric expectation. To our knowledge, the only work toward better understanding this unique form of error is that of [Higson et al \(2018\)](#), which quantifies it through bootstrapping techniques.
3. **Parallelization.** While NS can be parallelized across runs, NS does not allow one to make use of parallel computing architectures *within* runs without modifying the algorithm. The most natural way to parallelize NS, first proposed by [Burkoff et al \(2012\)](#) is as follows. If we generalize (8) to consider the K -th order statistic instead of simply the minimum ($K = 1$), then $1 - C$ has a $\text{Beta}(K, N + 1 - K)$ distribution, with expectation $K/(N + 1)$. Thus, at each iteration, we can instead remove the K points with the lowest likelihood, set $p_t = (1 - K/(N + 1))^t$, and parallelize the sampling across K threads. The approach will not only increase the bias of the algorithm by introducing additional quadrature error, but will also compound the problem mentioned in the previous issue (as a single value will now be used to represent the mean of a larger shell).
4. **Truncation Error.** Finally, of lesser concern, yet still worth noting is that NS commits an $\mathcal{O}(\exp(-T/N))$ truncation error ([Evans, 2007](#)) in the evidence estimate as a result of not performing quadrature on the entire $[0, 1]$ interval. A heuristic originally proposed by Skilling, which we call the *filling in* procedure is to simply add $\frac{1}{N} \sum_{k=1}^N \mathcal{L}(\mathbf{X}^k)$ after termination to the final evidence estimate. However, this is somewhat out of place with the rest of the quadrature. Using point process theory and techniques from the literature on unbiased estimation, [Walter \(2017\)](#) proposes an unbiased version of NS. However, this unbiasedness relies on the assumption of independent sampling, and comes with a cost of additional variance.

As mentioned earlier, all of these potential issues stem from the use of quadrature in NS. Indeed, the combined Monte Carlo/quadrature approach of NS seems to give somewhat of an overall awkwardness to the method. In the next section, we introduce SMC methodology,

which we will soon discover allows us to retain the essence of NS, but allay the objections just discussed.

3 SEQUENTIAL MONTE CARLO

We begin with an introduction to importance sampling, which is the fundamental idea behind SMC. Recall that, in our setting, $\pi(\mathbf{x}) \propto \gamma(\mathbf{x})$, where γ is a known function. For any probability density ν such that $\nu(\mathbf{x}) = 0 \Rightarrow \pi(\mathbf{x}) = 0$, it holds that

$$\begin{aligned} \pi(\varphi) &= \mathbb{E}_\pi[\varphi(\mathbf{X})] = \int_E \varphi(\mathbf{x})w(\mathbf{x})\nu(\mathbf{x})d\mathbf{x} \Big/ \int_E w(\mathbf{x})\nu(\mathbf{x})d\mathbf{x} \\ &= \mathbb{E}_\nu[\varphi(\mathbf{X})w(\mathbf{X})]/\mathbb{E}_\nu[w(\mathbf{X})], \end{aligned} \tag{12}$$

where $w(\mathbf{x}) = \gamma(\mathbf{x})/\nu(\mathbf{x})$ is called the *weight* function.

This suggests that one can draw $\mathbf{X}^1, \dots, \mathbf{X}^N \sim \nu$ and estimate (12) via the ratio

$$\sum_{k=1}^N \varphi(\mathbf{X}^k)w(\mathbf{X}^k) \Big/ \sum_{k=1}^N w(\mathbf{X}^k) = \sum_{k=1}^N \varphi(\mathbf{X}^k) \underbrace{\left(w(\mathbf{X}^k) \Big/ \sum_{k=1}^N w(\mathbf{X}^k) \right)}_{W^k},$$

where we call the $\{W^k\}_{k=1}^N$ the *normalized weights*.

A common measure of the quality of using ν with regard to approximating $\pi(\varphi)$ is the *effective sample size* (ESS),

$$\text{ESS} := \mathbb{E}_\nu [w(\mathbf{X})]^2 / \mathbb{E}_\nu [w(\mathbf{X})^2].$$

In practice, this can be estimated via

$$\widehat{\text{ESS}} = \left(\sum_{k=1}^N w(\mathbf{X}^k) \right)^2 \Big/ \sum_{k=1}^N w(\mathbf{X}^k)^2 = \left(\sum_{k=1}^N (W^k)^2 \right)^{-1}, \tag{13}$$

see [Liu \(2001, Chapter 2.5\)](#) for a full discussion. Unfortunately, in difficult high-dimensional settings, it is often hard to make a choice of importance sampling density to ensure that the ESS will be high (equivalently, that the variance of the normalized weights will be low).

SMC samplers ([Del Moral et al, 2006](#)) extend the idea of importance sampling to a general method for sampling from a sequence of probability densities $\{\pi_t\}_{t=1}^T$ defined on a common space E , as well as estimating their associated normalizing constants $\{\mathcal{Z}_t\}_{t=1}^T$ in a sequential

manner. This is accomplished by obtaining at each time step $t = 1, \dots, T$ a collection of random samples (called *particles*) with associated (normalized) weights $\{\mathbf{X}_t^k, W_t^k\}_{t=1}^T$, for $k = 1, \dots, N$, such that the weighted empirical measures of the cloud of particles,

$$\pi_t^N(d\mathbf{x}) = \sum_{k=1}^N W_t^k \delta_{\mathbf{X}_t^k}(d\mathbf{x}), \quad t = 1, \dots, T, \quad (14)$$

converge to their corresponding *target* measures $\pi_t(d\mathbf{x})$ as $N \rightarrow \infty$.

SMC samplers have three main elements:

1. **Mutation.** For each iteration $t > 1$, the population of particles are moved from $\{\mathbf{X}_{t-1}^k\}_{k=1}^N$ to $\{\mathbf{X}_t^k\}_{k=1}^N$ according to a (forward in time) Markov kernel K_t , for which we denote the associated density $K_t(\mathbf{x}' | \mathbf{x})$. This implicitly defines a new importance sampling density at each iteration via the recursive formula

$$\nu_t(\mathbf{x}') = \int_E \nu_{t-1}(\mathbf{x}) K_t(\mathbf{x}' | \mathbf{x}) d\mathbf{x}. \quad (15)$$

2. **Correction.** The weights of the particles are updated via an *incremental importance weight* function \tilde{w}_t , to ensure the particle system is correctly reweighted with respect to the next target density. This update involves multiplying the current weight of each particle by a corresponding incremental weight.
3. **Selection.** The particles are resampled according to their weights, which are then reset to $1/N$. A variety of resampling schemes can be used (see for example [Doucet and Johansen \(2011, Section 3.4\)](#)). However, the simplest is multinomial resampling. Here, the resampled population contains C_k copies of \mathbf{X}_t^k for each $k = 1, \dots, N$, where $(C_1, \dots, C_N) \sim \text{Multinomial}(N, (W_t^k)_{k=1}^N)$.

[Del Moral et al \(2006\)](#) show that one can use an arbitrary mutation kernel at each stage, without being required to compute the corresponding importance sampling density ν_t at each iteration. This is achieved by introducing an artificial backward (in time) kernel, which transforms the problem into one in the well-understood setting of filtering (for a comprehensive survey, see [Doucet and Johansen \(2011\)](#)). Here, sample paths of the particles' positions take values on the product space E^T , with the artificial joint distribution admitting each $\pi_t \propto \gamma_t$ (where γ_t is the unnormalized density) as a marginal. SMC samplers can be formulated in many different ways. For our purpose, we require SMC samplers for which K_t for $t > 1$ is a π_t -invariant MCMC kernel (or several iterations thereof). This approach is most straightforward and related directly to NS. For this case, a suitable choice of the *incremental weight* function at time t (i.e., one that will ensure the convergence (14))

is

$$\tilde{w}_t(\mathbf{x}_{t-1}) = \gamma_t(\mathbf{x}_{t-1})/\gamma_{t-1}(\mathbf{x}_{t-1}). \quad (16)$$

In this setting, the implicit backward kernel will be a good approximation to the *optimal* backward kernel, provided that π_t and π_{t-1} are sufficiently close.

SMC samplers give an approximation of $\pi_t(\varphi)$ at each iteration via

$$\pi_t^N(\varphi) := \sum_{k=1}^N W_t^k \varphi(\mathbf{X}_t^k). \quad (17)$$

Further to this, at each iteration SMC samplers give (somewhat remarkably) unbiased estimates of the ratios of normalizing constants

$$\widehat{\mathcal{Z}_t/\mathcal{Z}_{t-1}} = \underbrace{\sum_{k=1}^N W_{t-1}^k \tilde{w}_t(\mathbf{X}_{t-1}^k)}_{\pi_t^N(\tilde{w}_t)}, \text{ and } \widehat{\mathcal{Z}_t/\mathcal{Z}_1} = \prod_{k=2}^t \widehat{\mathcal{Z}_k/\mathcal{Z}_{k-1}}.$$

It follows readily that one can also obtain unbiased estimates of \mathcal{Z}_t if \mathcal{Z}_1 is known, by including the \mathcal{Z}_1 term when γ_1 appears in the incremental weights.

Remark 1 (*Adaptivity*) Introducing any sort of adaptivity into the SMC algorithm, for example resampling only if some criteria is met, choosing the next distribution online, or setting the parameters of K_t according to the particle population, will not necessarily preserve the unbiasedness or convergence properties of the SMC estimators. The analysis of adaptive SMC methods is technically involved. However, there are consistency results for certain adaptive schemes, see for example [Del Moral et al \(2012b\)](#), [Beskos et al \(2016\)](#), and [C erou and Guyader \(2016\)](#). Of course, one can always first run the algorithm adaptively, save the values of any adaptively chosen parameters, and then rerun the algorithm a second time and with these fixed.

SMC Samplers for Static Models

[Del Moral et al \(2006\)](#) provide a strategy for using an SMC sampler to sample from a fixed density π by defining a sequence of densities $\pi_1, \pi_2, \dots, \pi_T$ that transition from something that is easy to sample from (for example, the prior density) to π . This can be accomplished in a number of ways. We outline the two most common in the SMC literature. One approach ([Chopin \(2002\)](#)) is to define the sequence of target distributions such that at each stage the effect of the likelihood is gradually introduced by considering more data than the

last. The second method, first explored by Neal (2001) is called *temperature annealing*. Here, we have the sequence of densities

$$\pi_t(\mathbf{x}) \propto \nu(\mathbf{x})^{1-l_t} \pi(\mathbf{x})^{l_t}, \quad t = 1, \dots, T. \quad (18)$$

parametrized by some *temperature schedule*

$$l_1 = 0 < l_1 < \dots < l_{T-1} < l_T = 1,$$

where ν is some initial importance sampling density. In the Bayesian setting, a natural choice is the gradual change from prior to the posterior:

$$\pi_t(\mathbf{x}) \propto \eta(\mathbf{x}) \mathcal{L}(\mathbf{x})^{l_t}, \quad t = 1, \dots, T.$$

In practice, it is often difficult to make a good choice for the temperature schedule. This can be achieved (approximately) by choosing the next temperature $l_{t+1} \in (l_t, 1]$ adaptively online according to the criterion of effective sample size (ESS), as proposed by Jasra et al (2011). This ensures successive distributions are sufficiently close. For some $\alpha \in (0, 1)$, one can approximately maintain an ESS of αN between successive distributions by choosing the next temperature online so that a given ESS is maintained. In other words, for a collection of particles, we choose L_{t+1} (and thus the next density) so that the ESS for the current importance sampling step is equal to some desired amount. Formally stated, for $\tilde{w}_{t+1}^k(l) := W_t^k \exp(- (l - L_t) \log \mathcal{L}(\mathbf{X}_t^k))$, we solve

$$L_{t+1} = \inf_{l: L_t < l \leq 1} \left\{ \left(\sum_{k=1}^N \tilde{w}_{t+1}^k(l) \right)^2 / \sum_{k=1}^N \tilde{w}_{t+1}^k(l)^2 = \alpha N \right\}, \quad (19)$$

via the bisection method, for example. Pseudocode for adaptive temperature-annealed SMC (TA-SMC) is given in Algorithm 2.

4 NESTED SAMPLING VIA SEQUENTIAL MONTE CARLO

The similarity between SMC and NS at this point is evident. Both methods draw from some initial distribution (in our case, the prior distribution), and involve traversing a population of particles through a sequence of distributions, which is of an adaptive nature in NS, but may be either adaptive or fixed in SMC. From the outset, it would *seem* that nested sampling is some sort of SMC algorithm, yet it is distinct in its use of a quadrature rule.

Algorithm 2: Adaptive Temperature–Annealed SMC

input : population size N

$t \leftarrow 1, L_1 \leftarrow 0, \mathcal{Z} \leftarrow 1$

for $k = 1$ **to** N **do** draw $\mathbf{X}_1^k \sim \eta$ and set $W_1^k = 1/N$

while $\gamma_t \neq 1$ **do**

$t \leftarrow t + 1$

$L_t \leftarrow$ solution to (19) obtained via bisection

for $k = 1$ **to** N **do** $w_t^k \leftarrow W_{t-1}^k \mathcal{L}(\mathbf{X})^{L_t - L_{t-1}}$

$\widehat{\mathcal{Z}} \leftarrow \widehat{\mathcal{Z}} \left(\sum_{k=1}^N w_t^k \right)$

$\{\widetilde{\mathbf{X}}_{t-1}^k\}_{k=1}^N \leftarrow$ resample $\{\mathbf{X}_{t-1}^k\}_{k=1}^N$ according to $\{w_t^k\}_{k=1}^N$

for $k = 1$ **to** N **do** $W_t^k \leftarrow 1/N$

$\{\mathbf{X}_t^k\}_{k=1}^N \leftarrow$ move $\left(\{\widetilde{\mathbf{X}}_{t-1}^k\}_{k=1}^N, K_t\right)$

end

return samples $\{\mathbf{X}_t^k\}_{k=1}^N$ and estimator of the marginal likelihood, $\widehat{\mathcal{Z}}$.

Further, it has an interesting point of difference in that NS does not transition from the prior to the posterior.

It turns out, somewhat suprisingly, that this difference is largely a matter of interpretation. Nested Sampling *is* a special type of adaptive SMC algorithm, where weights are assigned in a suboptimal way. In order to demonstrate this in a straightforward manner, we proceed as follows. We first present a general class of SMC methods called Nested Sampling via Sequential Monte Carlo (NS-SMC) methods. Then, we will proceed to show the correspondence with the original NS method by introducing an adaptive version of NS–SMC, and finally modifying this adaptive version further so that it more closely resembles (and is equivalent as $N \rightarrow \infty$) to NS.

We begin by considering a set *threshold schedule*,

$$l_1 = -\infty < l_2 < \dots < l_T < l_{T+1} = \infty, \quad (20)$$

which in turn parametrizes a sequence of nested sets

$$E_1 = E \supset E_2 \supset \dots \supset E_{T-1} \supset E_T,$$

via

$$E_t := \{\mathbf{x} \in E : \mathcal{L}(\mathbf{x}) \geq l_t\}, \quad t = 1, \dots, T.$$

Next, define the sequence of constrained densities:

$$\eta_t(\mathbf{x}) = \frac{\eta(\mathbf{x})\mathbb{I}\{\mathbf{x} \in E_t\}}{\underbrace{\mathbb{P}_\eta(\mathbf{X} \in E_t)}_{\mathcal{P}_t}}, \quad t = 1, \dots, T. \quad (21)$$

We now consider directly shells of \mathcal{L} , via the sets,

$$\check{E}_t = \{\mathbf{x} \in E : l_t < \mathcal{L}(\mathbf{x}) \leq l_{t+1}\}, \quad t = 1, \dots, T.$$

Observe that sets $(\check{E}_t)_{t=1}^T$ form a partition of E , $\check{E}_t \subset E_t$ for $t = 1, \dots, T-1$, and that because $l_{T+1} = \infty$, we have that $\check{E}_T = E_T$. Next, we define a second set of densities, corresponding to constrained versions of π to these shells,

$$\pi_t(\mathbf{x}) = \frac{\gamma(\mathbf{x})\mathbb{I}\{\mathbf{x} \in \check{E}_t\}}{\underbrace{\int_E \gamma(\mathbf{x})\mathbb{I}\{\mathbf{x} \in \check{E}_t\}d\mathbf{x}}_{\mathcal{Z}_t}}, \quad t = 1, \dots, T.$$

With the above in mind, we define a class of SMC Samplers called NS-SMC samplers, that have the following two properties:

1. Given samples targeting η_{t-1} , the importance sampling branches into two paths. One path targets the next constrained prior η_t , while the second targets (and terminates at) the constrained posterior π_{t-1} . This branching of importance sampling paths occurs for all but η_T , which proceeds only to π_T . This is illustrated in Figure 1.

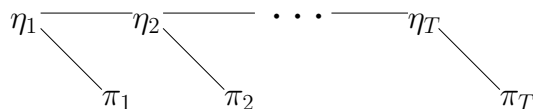


Figure 1: Importance sampling scheme for NS-SMC.

The importance sampling procedure just described results in T (dependent) SMC samplers which output estimators of \mathcal{Z}_t , as well as samples that can be used to estimate $\pi_t(\varphi)$ for each $t = 1, \dots, T$.

2. The resulting estimators for both \mathcal{Z}_t and $\pi_t(\varphi)$ for $t = 1, \dots, T$ are used together to

estimate (1) via use of the identity

$$\pi(\varphi) = \sum_{t=1}^T \mathbb{P}_\pi(\mathbf{X} \in \check{E}_t) \mathbb{E}_{\pi_t}[\varphi(\mathbf{X})] = \sum_{t=1}^T \frac{\mathcal{Z}_t}{\mathcal{Z}} \pi_t(\varphi). \quad (22)$$

For simplicity (and similarity to the original NS method), we consider the case where each η_t is used *directly* as an importance sampling density for π_t without any further resampling or moving. In such a case, we need only consider an SMC sampler that sequentially targets η_1, \dots, η_T , because all terms in (22) can be rewritten in terms of expectations with respect to those densities. Thus, NS–SMC can be viewed as an extension to the rare–event SMC (multilevel splitting) method of C erou et al (2012), which uses density sequences of the form (21) in order to estimate the probability (normalizing constant) \mathcal{P}_T .

For ease of presentation, below we use shorthand notation analogously to (17). For example, instead of $\sum_{k=1}^N W_t^k \mathbb{I}\{\mathbf{X}_t^k \in E_t\} \mathcal{L}(\mathbf{X}_t^k) \varphi(\mathbf{X}_t^k)$, we write $\eta_t^N(\mathbb{I}_{E_t} \mathcal{L} \varphi)$.

Noting that $\pi_t/\eta_t = \mathcal{L} \mathbb{I}_{\check{E}_t}$, we have

$$\pi(\varphi) = \sum_{t=1}^T \frac{\mathcal{Z}_t}{\mathcal{Z}} \pi_t(\varphi) = \sum_{t=1}^T \frac{\mathcal{Z}_t}{\mathcal{Z}} \frac{\eta_t(\mathcal{L} \mathbb{I}_{\check{E}_t} \varphi)}{\eta_t(\mathcal{L} \mathbb{I}_{E_t})}, \quad (23)$$

which is estimated via

$$\pi^N(\varphi) = \sum_{t=1}^T \frac{\widehat{\mathcal{Z}}_t}{\sum_{s=1}^T \widehat{\mathcal{Z}}_s} \cdot \frac{\eta_t^N(\mathcal{L} \mathbb{I}_{\check{E}_t} \varphi)}{\eta_t^N(\mathcal{L} \mathbb{I}_{E_t})}. \quad (24)$$

Note that

$$\begin{aligned} \widehat{\mathcal{Z}}_t \frac{\eta_t^N(\mathcal{L} \mathbb{I}_{\check{E}_t} \varphi)}{\eta_t^N(\mathcal{L} \mathbb{I}_{E_t})} &= \widehat{\mathcal{P}}_t \underbrace{\eta_t^N(\mathcal{L} \mathbb{I}_{E_t})}_{\widehat{\mathcal{Z}}_t} \frac{\eta_t^N(\mathcal{L} \mathbb{I}_{\check{E}_t} \varphi)}{\eta_t^N(\mathcal{L} \mathbb{I}_{E_t})} = \widehat{\mathcal{P}}_t \eta_t^N(\mathcal{L} \mathbb{I}_{\check{E}_t} \varphi) \\ &= \sum_{k=1}^N \widehat{\mathcal{P}}_t W_t^k \mathcal{L}(\mathbf{X}_t^k) \mathbb{I}\{\mathbf{X}_t^k \in \check{E}_t\} \varphi(\mathbf{X}_t^k). \end{aligned} \quad (25)$$

The final equality above implies that reweighting with respect to the (full) posterior requires that each particle targeting η_t at iteration t is assigned the weight

$$\check{w}_t^k = \widehat{\mathcal{P}}_t W_t^k \mathcal{L}(\mathbf{X}_t^k) \mathbb{I}\{\mathbf{X}_t^k \in \check{E}_t\}.$$

In turn, we have that

$$\pi^N(\varphi) = \sum_{t=1}^T \sum_{k=1}^N \check{W}_t^k \varphi(\mathbf{X}_t^k), \quad \check{W}_t^k = \frac{\check{w}_t^k}{\sum_{t=1}^T \sum_{k=1}^N \check{w}_t^k}. \quad (26)$$

is an estimator of $\pi(\varphi)$.

Pseudocode for this version of NS-SMC is given in Algorithm 3. We call this version *Fixed* NS-SMC (as opposed to adaptive) as one specifies $\{l_t\}_{t=1}^T$ apriori. Note that resampling occurs at each iteration in order to avoid wasting computational effort moving particles with zero weight.

Algorithm 3: Fixed NS-SMC

input : population size N and thresholds $\{l_t\}_{t=1}^{T+1}$ satisfying (20).

$\hat{\mathcal{P}}_1 \leftarrow 1, t \leftarrow 1$

for $k = 1$ **to** N **do** draw $\mathbf{X}_1^k \sim \eta$ and $W_1^k \leftarrow 1/N$

while true **do**

$t \leftarrow t + 1$

for $k = 1$ **to** N **do**

$w_t^k \leftarrow W_{t-1}^k \mathbb{I}\{\mathcal{L}(\mathbf{X}_{t-1}^k) \geq l_t\}$ // weight update for $\eta_t \rightarrow \eta_{t+1}$

$\check{w}_{t-1}^k \leftarrow \hat{\mathcal{P}}_{t-1} W_{t-1}^k \mathcal{L}(\mathbf{X}_{t-1}^k) \mathbb{I}\{\mathcal{L}(\mathbf{X}_{t-1}^k) < l_t\}$ // weight for π

end

$\hat{\mathcal{P}}_t \leftarrow \hat{\mathcal{P}}_{t-1} \left(\sum_{k=1}^N w_t^k \right)$ and $\hat{\mathcal{Z}}_{t-1} \leftarrow \sum_{k=1}^N \check{w}_{t-1}^k$

if $\sum_{k=1}^N w_t^k = 0$ **then** $T \leftarrow t$ and **break**

$\{\tilde{\mathbf{X}}_{t-1}^k\}_{k=1}^N \leftarrow$ resample $\{\mathbf{X}_{t-1}^k\}_{k=1}^N$ according to $\{w_t^k\}_{k=1}^N$

for $k = 1$ **to** N **do** $W_t^k \leftarrow 1/N$

$\{\mathbf{X}_t^k\}_{k=1}^N \leftarrow$ move $\left(\{\tilde{\mathbf{X}}_{t-1}^k\}_{k=1}^N, K_t \right)$ — where K_t is an η_t -invariant MCMC kernel

end

$\hat{\mathcal{Z}} = \sum_{t=1}^T \hat{\mathcal{Z}}_t$

return weighted samples $\{\{\mathbf{X}_t^k, \check{w}_t^k\}_{k=1}^N\}_{t=1}^{T+1}$ and estimator of the marginal likelihood, $\hat{\mathcal{Z}}$.

The issue of how to appropriately set $\{l_t\}_{t=1}^{T+1}$ will be addressed shortly. However, for now we return to the concerns with NS outlined earlier in Section 2.1, and note how they are addressed by NS-SMC:

1. **Assumption of Independent Samples.** NS-SMC has no requirement that the

samples be independent. Moreover, the unbiasedness and consistency properties of Fixed NS–SMC are established in Appendix A via a straightforward application of Feynman–Kac formalism.

2. **Effect of Quadrature on Posterior Inferences.** All errors in NS–SMC are solely Monte Carlo errors. The analogous error to that of NS in estimating $\pi(\varphi)$ is more natural and occurs as the result of the error in the ratios $\widehat{\mathcal{Z}}_t/\widehat{\mathcal{Z}}$ for $t = 1, \dots, T$.
3. **Parallelization.** NS–SMC is easily parallelizable without any further modification. After resampling, the move step, which is often the most computationally intensive, can be parallelized at the particle level.
4. **Truncation Error.** NS–SMC commits no truncation error as the final density π_T accounts for the interval $[0, p_T]$ which is omitted from the NS quadrature. However, it is important to note that the choice of the final threshold l_T will still have an effect on the variance of NS–SMC. Nevertheless, the absence of truncation error is a key factor in allowing NS–SMC to obtain unbiased estimates of \mathcal{Z} .

4.1 Adaptive NS–SMC (ANS–SMC)

Generally one does not have a good idea of a choice of $\{l_t\}_{t=1}^T$ that will perform well. In a similar manner to adaptive TA–SMC, at each iteration t in Algorithm 3 we can replace l_t with a random threshold L_t that is chosen adaptively. Ensuring an estimated ESS for η_t that is at least $(1 - \rho)N$ simply reduces to choosing L_t to be the $(1 - \rho)$ quantile of the values $(\mathcal{L}(\mathbf{X}_{t-1}^k))_{k=1}^N$. Such a choice in NS–SMC results in the online specification of both η_t and π_{t-1} . While $(1 - \rho)$ is analogous to α , we use this notation as it is common in adaptive multilevel splitting algorithms (Botev and Kroese, 2008, 2012; Cérou et al, 2012), where ρ is interpreted as the proportion of particles that one desires to lie above each successive adaptively chosen threshold. For NS–SMC, $(1 - \rho)$ can be interpreted as the desired proportion of samples with non-zero weight for π_{t-1} .

As with NS, ANS–SMC also requires that the iteration at which termination occurs is determined online in some manner. The termination criterion/procedure we use compares the evidence estimate after an iteration with an estimate that would be obtained by instead terminating at that iteration. At each iteration, after computing L_t , we compare the ratio of the two evidence estimates, and if it is greater than $1 - \epsilon$, we instead set $L_t = \infty$, and declare $T = t - 1$. In our examples, we found that the choice $\epsilon = 10^{-2}$ was suitable.

Remark 2 For a given adaptive choice of the next threshold L_t , experiments indicate that

there is considerably less bias (particularly for small N) in the estimates of \mathcal{Z} if one sets $\eta_t \propto \eta \mathbb{I}_{\{\mathcal{L} > L_t\}}$ and $\pi_{t-1} \propto \gamma \mathbb{I}_{\{L_{t-1} < \mathcal{L} \leq L_t\}}$ instead of $\eta_t \propto \eta \mathbb{I}_{\{\mathcal{L} \geq L_t\}}$ and $\pi_{t-1} \propto \gamma \mathbb{I}_{\{L_{t-1} \leq \mathcal{L} < L_t\}}$.

4.2 Improved NS

In this section, we follow the original NS sampling scheme more closely and derive an SMC estimator using a similar two-branched importance sampling scheme as illustrated in Figure 1. Specifically, we choose the sequence of distributions adaptively so only one particle lies below the next threshold and conduct our move step in a similar manner to NS. Just as Algorithm 3 can be viewed as an extension to the rare-event SMC algorithm of Cérou et al (2012), the more direct variant of NS we describe here can be viewed as an extension of the static *Last Particle Method* (LPM) for rare-event simulation of Guyader et al (2011). Unfortunately, there is a lack of theoretical results for the LPM in the setting where MCMC is used (due mainly to the special type of move step, outlined shortly).

We call this method Improved Nested Sampling (INS). The sampling scheme is *identical* for NS and INS, and thus one can obtain both estimates from the same nested sampling run. Somewhat surprisingly, provided the filling in procedure is used, the NS and INS estimators of model evidence and posterior quantities *also* become identical as $N \rightarrow \infty$. This provides insight into why NS seems to perform well in practice despite a violation of the independence assumption that underlies its quadrature.

INS is a modified version of ANS-SMC with the following differences:

1. We enforce for iterations $t < T$ that a *single* particle has non-zero incremental weight for π_t . That is, like NS, we have only one particle that does not have support on the next constrained version of η .
2. We conduct the resampling and mutation step in a manner that ensures that MCMC is only required to replenish the “worst-performing particle”.

Unfortunately, setting $\rho = (N - 1)/N$ alone in ANS-SMC does not always ensure the first property above, which requires that all particles correspond to a *unique* value of \mathcal{L} . In discrete settings it is common for some particles to have the same value of \mathcal{L} . However, even if $\mathbb{P}_\eta(\mathcal{L}(\mathbf{X}) = l) = 0$ for all $l \in \mathbb{R}$, there may still be duplicate particles if there is a non-zero probability that the MCMC kernel will return the same point (as is the case in Metropolis-Hastings MCMC).

While technically involved, the solution is reasonably straightforward. We employ auxiliary

variables in a similar manner to (Murray, 2007, pgs. 96–98), who proposes a variant of NS that can be applied to discrete spaces. A similar approach is used in Cérou and Guyader (2016) to break ties in the theoretical analysis of adaptive multilevel splitting, see .

For brevity, we assume the aforementioned condition that $\mathbb{P}_\eta(\mathcal{L}(\mathbf{X}) = l) = 0$ for all $l \in \mathbb{R}$, which is typically the case for continuous E . However, this condition excludes certain cases of what Skilling (2006) refers to as “degenerate likelihoods”. Under this assumption, the approach about to be described is entirely *implicit* if one does not consider any auxiliary variables, ignores any duplicate particles, and just moves a single particle at each iteration, yielding the same L_t for multiple iterations. However, in discrete cases, one must consider the extended space *explicitly* and conduct the move step differently, see Murray et al (2005) and Murray (2007).

We extend the space from E to $E \times (0, 1)$ via a uniformly distributed variable U . That is, we have

$$\eta(\mathbf{x}, u) = \eta(\mathbf{x})\mathbb{I}\{0 < u < 1\}, \text{ and } \pi(\mathbf{x}, u) \propto \gamma(\mathbf{x})\mathbb{I}\{0 < u < 1\}.$$

In this setting, define the *augmented* threshold schedule:

$$(l_1, v_1) = (-\infty, 0) < (l_2, v_2) < \cdots < (l_T, v_T) < (l_{T+1}, v_{T+1}) = (\infty, 1),$$

where $(l, v) < (l', v')$ is to be understood as either $l' > l$, or that both $l' = l$ and $v' > v$.

Applying a similar derivation of NS–SMC to that given earlier in this section, we have the sets

$$\begin{aligned} E_t &= \{(\mathbf{x}, u) \in E \times (0, 1) : (\mathcal{L}(\mathbf{x}) > l_t, 0 < u < 1) \cup (\mathcal{L}(\mathbf{x}) = l_t, v_t < u < 1)\} \\ \check{E}_t &= \{(\mathbf{x}, u) \in E \times (0, 1) : (l_t < \mathcal{L}(\mathbf{x}) < l_{t+1}, 0 < u < 1) \cup (\mathcal{L}(\mathbf{x}) = l_t, v_t < u \leq v_{t+1})\}, \end{aligned} \tag{27}$$

and the densities

$$\eta_t(\mathbf{x}, u) \propto \eta(\mathbf{x}, u)\mathbb{I}\{(\mathbf{x}, u) \in E_t\}, \text{ and } \pi_t(\mathbf{x}, u) \propto \pi(\mathbf{x}, u)\mathbb{I}\{(\mathbf{x}, u) \in \check{E}_t\}, \text{ for } t = 1, \dots, T.$$

Note that this setup ensures that the E_t sets are nested and that the \check{E}_t sets form a partition.

Prior to demonstrating how INS relates to NS, we stress the following. Due to the special type of mutation step, the algorithm falls outside of the standard SMC sampler framework (which requires the same forward kernel for all particles).

Despite this, we continue to use incremental weight functions of the form (16). While this choice seems to be a natural one (and matches the approach used in the LPM), it implicitly

assumes that the INS procedure produces a population of particles that are marginally distributed according to (the adaptively chosen) η_t at each time step. This may only hold approximately in practice, and even establishing that the property holds as $N \rightarrow \infty$ is difficult due to the complicated combination of adaptively chosen distributions and non-standard mutation step.

Nevertheless, we present the method for the purpose of making clear the connection of NS with the NS-SMC framework. Moreover, we point out that while our assumption on the marginal distribution of the particles at each iteration is a strong one, it is substantially weaker than that required in the original formulation of NS, which assumes not only the same condition on the marginal distributions of the particles, but *also* that the particles are independent. Recall that both of these conditions are required for the property (8) to hold.

With the above in mind, we sketch the key aspects of the INS below.

Adaptive Choice of Densities. At each (non-final) iteration, we determine π_{t-1} and η_t adaptively (via the choice of the next threshold parameters L_t and V_t) as follows. First, we set $L_t = \min_{1 \leq k \leq N} \mathcal{L}(\mathbf{X}_{t-1}^k)$. Next, denote the indices of the particles satisfying $\mathcal{L}(\mathbf{x}) = L_t$ by \mathcal{I} . We “break ties” by choosing $V_t = \min_{k \in \mathcal{I}} \{U_{t-1}^k\}$.

Reweighting. Importance sampling takes place for η_t and π_{t-1} with the incremental weight functions \mathbb{I}_{E_t} and $\mathcal{L}\mathbb{I}_{\check{E}_{t-1}}$, respectively.

By construction, we will have $N-1$ samples with non-zero incremental weight for η_t , giving

$$\hat{\mathcal{P}}_t = \underbrace{\left(\frac{N-1}{N}\right)^{t-1}}_{\hat{\mathcal{P}}_{t-1}} \underbrace{\frac{N-1}{N}}_{\eta_{t-1}^N(\mathbb{I}_{E_t})} = \left(\frac{N-1}{N}\right)^t.$$

Similarly, only one particle, denoted $(\check{\mathbf{X}}_{t-1}, \check{U}_{t-1})$, will have non-zero incremental weight for π_{t-1} (and thus non-zero weight for π), so we have

$$\hat{\mathcal{Z}}_{t-1} = \underbrace{\left(\frac{N-1}{N}\right)^{t-1}}_{\hat{\mathcal{P}}_{t-1}} \underbrace{\frac{1}{N} \mathcal{L}(\check{\mathbf{X}}_{t-1})}_{\eta_{t-1}^N(\mathcal{L}\mathbb{I}_{\check{E}_{t-1}})}. \quad (28)$$

Note that (28) is not only $\hat{\mathcal{Z}}_{t-1}$ but also precisely the weight of $\check{\mathbf{X}}_{t-1}$ with respect to π as in (25). Recall that this is also the case with NS.

Resampling and Mutation. We resample according to a *residual* scheme, and reset all weights to $1/N$. As we have $N-1$ samples with equal (non-zero) weight, residual resampling will result in $N-1$ unique particle positions $\{\widetilde{\mathbf{X}}_{t-1}^k, \widetilde{U}_{t-1}^k\}_{k=1}^{N-1}$, as well as a final particle $(\widetilde{\mathbf{X}}_t^N, U_t^N)$ that is a copy of one of the others.

The mutation step is as follows. We begin by applying the identity map all particles except the N -th one, moving $\{\widetilde{\mathbf{X}}_{t-1}^k, \widetilde{U}_{t-1}^k\}_{k=1}^{N-1}$ to $\{\mathbf{X}_t^k, U_t^k\}_{k=1}^{N-1}$. Then, we perform the following two η_t -invariant moves in sequence to move $(\widetilde{\mathbf{X}}_{t-1}^N, \widetilde{U}_{t-1}^N)$ to (\mathbf{X}_t^N, U_t^N) .

First, we move $\widetilde{\mathbf{X}}_{t-1}^N$ to \mathbf{X}_t^N by applying some fixed number of iterations of an $\eta_t(\mathbf{x}|u)$ -invariant MCMC kernel. Note that

$$\eta_t(\mathbf{x}|u) \propto \begin{cases} \eta(\mathbf{x})\mathbb{I}\{\mathcal{L}(\mathbf{x}) \geq L_t\} & u > V_t \\ \eta(\mathbf{x})\mathbb{I}\{\mathcal{L}(\mathbf{x}) > L_t\} & u \leq V_t \end{cases},$$

so this is simply sampling from a constrained version of η as in standard NS or NS-SMC. Next, we draw from a new u position according to

$$\eta_t(u|\mathbf{x}) \propto \begin{cases} \mathbb{I}\{0 < u < 1\} & \mathcal{L}(\mathbf{x}) > L_t \\ \mathbb{I}\{V_t < u < 1\} & \mathcal{L}(\mathbf{x}) = L_t \end{cases}.$$

Final Iteration. The reweighting and mutation steps continue up until a termination criteria is satisfied. At this point, we declare $T = t-1$, and set the final threshold parameters $L_t = \infty$, $U_t = 1$. Here, all samples will have non-zero incremental weight for π_T , and we have

$$\widehat{\mathcal{Z}}_T = \left(\frac{N-1}{N}\right)^T \frac{1}{N} \sum_{k=1}^N \mathcal{L}(\mathbf{X}_T^k). \quad (29)$$

Note that the above normalizing constant estimator bears similarity to the “filling in” heuristic in NS. However, here it arises *naturally* as a final step, and uses $\left(\frac{N-1}{N}\right)^T$ to estimate p_T , as opposed to $\exp(-T/N)$.

Despite this similarity, it still appears that (28) is distinct from its analogous term in NS. However, by means of some simple algebraic manipulation, we obtain the identity

$$\left(\frac{N-1}{N}\right)^t = \left(\frac{N-1}{N}\right)^{t-1} \left(1 - \frac{1}{N}\right) = \left(\frac{N-1}{N}\right)^{t-1} - \left(\frac{N-1}{N}\right)^{t-1} \frac{1}{N},$$

which, after rearrangement, reveals that

$$\left(\frac{N-1}{N}\right)^{t-1} \frac{1}{N} = \left(\frac{N-1}{N}\right)^{t-1} - \left(\frac{N-1}{N}\right)^t, \quad (30)$$

resembling precisely the Riemann sum quadrature rule, with the choice $p_t = \left(\frac{N-1}{N}\right)^t$. The most notable aspect of this is that at no stage in the derivation of INS did we require the property given in (8), which would require samples to be independent.

We give this version of NS with the “improved” moniker as the alternative choices for p_t have been found to yield superior estimators of \mathcal{P}_t when compared to those proposed originally by Skilling (2006). Guyader et al (2011) show (under the same idealized sampling assumption as NS), that the LPM estimators $\hat{\mathcal{P}}_t = \left(\frac{N-1}{N}\right)^t$ are *unbiased* estimators of $\mathbb{P}_\eta(\mathcal{L}(\mathbf{X}) \geq L_t)$. Further to this, Walter (2017, Remark 1) shows that this estimator results in superior estimates over $\exp(-t/N)$ in terms of variance so long as $p_t > \exp(-1)$. In light of this, Walter suggests using Riemann sum quadrature using these alternative values for p_t as it will result in a superior NS estimator.

The final piece of the puzzle connecting NS with INS and the overall NS–SMC framework, is that as $N \rightarrow \infty$ we have that $\left(\left(\frac{N-1}{N}\right)^t - e^{-t/N}\right) \rightarrow 0$, so NS’s weights become equal to those of INS. We give a simple illustration of this convergence in Figure 2, where we plot the ratio of (30) to the standard NS Riemann sum / trapezoidal rule terms after $T/N = 10$ iterations of NS for different N (NS gives identical estimates for p_t for this choice, regardless of N). The convergence will be slower for larger T/N .

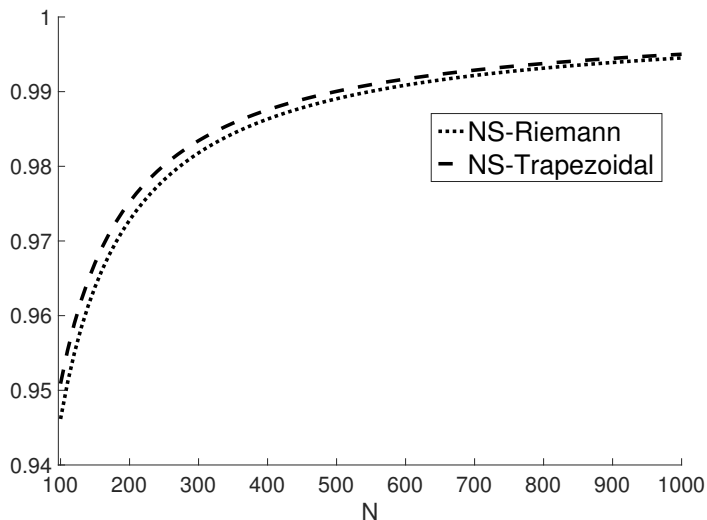


Figure 2: Ratio of weights for INS for $T/N = 10$, relative to NS estimators.

This provides some insight as to why NS seems to deliver correct results in practice, even when particles are far from independent (as will soon be demonstrated numerically). In essence, NS is an *adaptive* SMC algorithm on an extended space, where the choice of weights are, in a sense, sub-optimal.

4.3 Phase Transition Example

In order to compare the different variants of NS-SMC and NS, we shall use a phase transition example. Nested sampling is robust to models that contain phase transitions, i.e., models for which the graph of $\log p$ against $\log \mathcal{L}(F_{\mathcal{L}(\mathbf{x})}^{-1}(p))$ is not concave. For a full discussion of the challenges of phase transition phenomena, including why they can be challenging for temperature based methods such as TA-SMC and the *power posteriors* method of [Friel and Pettitt \(2008\)](#), we refer back to the original NS paper ([Skilling, 2006](#)). In a Bayesian context, a phase transition can be understood intuitively as having a likelihood function that is spiked and changes rapidly in certain regions. While this would seem to be a pathological type of behaviour restricted to problems in physics, it is known to occur in statistical settings, see for example [Brewer \(2014\)](#).

Similar to [Skilling \(2006\)](#), we consider the estimation of

$$\mathcal{Z} = \int_{\mathbb{R}^n} \underbrace{\left(\sum_{k=1}^2 a_k \phi_{\sigma_k}(\mathbf{x}) \right)}_{\mathcal{L}(\mathbf{x})} \underbrace{\frac{\mathbb{I}\{\|\mathbf{x}\| < 1\}}{V(\mathcal{B}_n)}}_{\eta(\mathbf{x})} d\mathbf{x},$$

where ϕ_{σ} denotes the pdf of a multivariate normal distribution with standard deviation σ for each component, centred at the origin, and $V(\mathcal{B}_n)$ denotes the volume of an n -dimension unit hypersphere. This problem can be viewed as estimating the model evidence of a model with uniform “prior” on the unit ball, and a mixture of two multivariate normals centered at the origin as a “likelihood function”. Despite the conceptual simplicity of this problem, it is still difficult computationally, and we can introduce a phase transition by varying parameters appropriately. In our case, we introduce a phase transition by specifying $\boldsymbol{\sigma} = (0.1, 0.01)$ and $\mathbf{a} = (0.25, 0.75)$, which introduces a large “spike” in \mathcal{L} due to the second mixture component. This particular example is also interesting as we are able to perform *exact sampling* from each η_t . This corresponds to using the *optimal* forward kernel.

In order to illustrate the effects of particle dependency, we also implement a version with MCMC. For an MCMC kernel, we perform ten iterations of a variant of the random walk sampler where we simply propose a movement along a randomly chosen coordinate axis. In order to ensure the sampler is well suited across progressively narrower densities, we choose h to be $1/10$ or $1/40$ with equal probability. We remark that this method strongly outperforms the obvious first choice of the standard random walk sampler. For NS and ANS-SMC, we use our knowledge of the problem to set the termination criterion to be $L_t/\mathcal{L}(\mathbf{0}) \geq 0.75$; i.e., we stop when the current threshold is higher than 75% of the maximum. While this ensures that the truncation error for NS is very small, we still use the filling-in procedure.

For (fixed) NS-SMC, we use the thresholds obtained via a pilot run of ANS-SMC.

In terms of simulation effort, it is worth noting that a choice of $\rho = 0.37$ ($\approx \exp(-1)$) for ANS-SMC yields around the same number of likelihood evaluations as NS. This is because for one iteration of NS-SMC, we spend an effort proportional to N (each particle is moved/generated), whereas for NS the effort is proportional to 1. As $\exp(-1) \approx (\frac{N-1}{N})^N$, we would expect roughly (discounting the effect of resampling and moving all the particles at once in the case of NS-SMC) that the two algorithms will have compressed a similar amount for prior mass for the same amount of likelihood evaluations. Thus, for purposes of a more direct comparison with NS / INS, we use this choice of ρ . We also implement adaptive TA-SMC for this example, where we use the conservative choice of $\alpha = 0.95$ and 20 MCMC repeats. Note that TA-SMC with $\alpha = 0.95$ will attempt to maintain an ESS of $0.95N$ between successive distributions, and thus will progress slower and allow the particles to move around the space more.

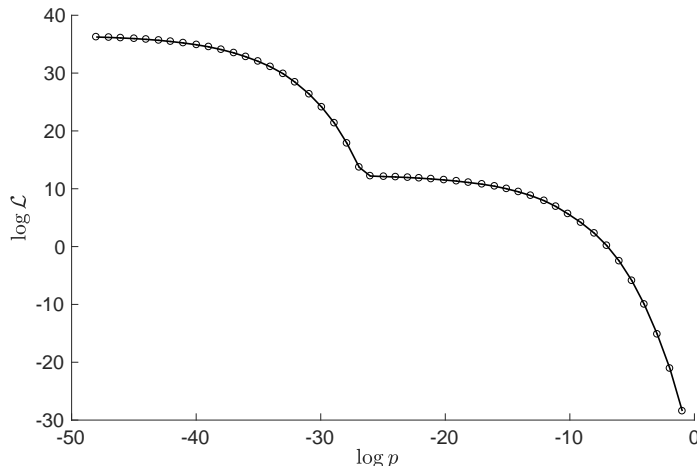


Figure 3: Phase transition diagnostic plot for the ten-dimensional sphere example. The phase transition appears around $\log p = -27$, corresponding to approximately 10^{-12} remaining prior mass.

The results given in Table 1 demonstrate several things. Firstly, we see that both the variance and bias in the integral estimate seems more pronounced for small N when MCMC is used. The observed (upward) bias for low N is a problem which seems to become more severe when samples are dependent.

Most notable is the exceptionally poor performance of TA-SMC, which fails on two accounts. Firstly, temperature-based methods are ill-suited to phase transitions. In the context of SMC, the nature of such problems means that the *actual* ESS will be low unless

successive temperatures are very close. Secondly, at each iteration, TA-SMC vastly overestimates its ESS for any given l as a result of there being no particles in the spike. That is, TA-SMC is unable to identify when it has a poor approximation of the current target. This results in the adaptive choice of levels failing to achieve its goal, and producing very poor estimates as a result. For example, notice how for $N = 10^2$ and $N = 10^3$, it appears to have missed three quarters of the integral belonging to the spike completely. Interestingly, we see TA-SMC occasionally successfully find the spike in the $N = 10^4$ case, now giving a more accurate result, but with enormous variance. In contrast, NS-SMC’s estimates of its own ESS will generally be better behaved, as the incremental weights along the η_t path will be either zero or one.

Table 1: Results for the 10-dimensional sphere example with phase transition. Results for $N = 10^2$ correspond to 1000 runs, while $N = 10^3$ and $N = 10^4$ correspond to 100 runs. We have that $\mathcal{Z} = 1$.

sampler	method	$N = 10^2$		$N = 10^3$		$N = 10^4$	
		$\widehat{\mathcal{Z}}$ (SE%)	evals	$\widehat{\mathcal{Z}}$ (SE%)	evals	$\widehat{\mathcal{Z}}$ (SE%)	evals
Exact	NS	1.14 (1.9)	5.0×10^3	1.01 (1.8)	5.0×10^4	1.000 (0.5)	5.0×10^5
	INS	0.99 (1.6)	5.0×10^3	1.00 (1.8)	5.0×10^4	0.999 (0.5)	5.0×10^5
	ANS-SMC	1.00 (2.2)	4.9×10^3	1.00 (2.1)	4.9×10^4	1.009 (0.7)	4.9×10^5
	NS-SMC	0.99 (2.4)	5.0×10^3	0.99 (2.1)	5.0×10^4	1.010 (0.6)	5.0×10^5
MCMC	NS	1.52 (9.8)	4.9×10^4	1.11 (5.2)	4.9×10^5	1.01 (1.6)	4.8×10^6
	INS	1.33 (8.6)	4.9×10^4	1.10 (5.1)	4.9×10^5	1.01 (1.5)	4.8×10^6
	ANS-SMC	1.19 (5.8)	4.8×10^4	1.06 (4.0)	4.8×10^5	1.02 (1.2)	4.8×10^6
	NS-SMC	1.01 (4.4)	4.8×10^4	0.94 (3.5)	4.8×10^5	1.00 (1.1)	4.8×10^6
	TA-SMC	0.24 (0.5)	4.7×10^4	0.25 (0.2)	4.8×10^5	1.03 (73)	4.8×10^6

While this example illustrates the similarity between NS and NS-SMC methods, showing how they can each handle phase transitions where TA-SMC has difficulty, the question arises as to how NS-SMC compares to TA-SMC on challenging and realistic problems. However, prior to conducting a comparative study, we first consider how one can attempt to ensure the best possible performance for both methods.

5 PRACTICAL CONSIDERATIONS

The implementation of SMC methods requires a specification of kernel parameters, and the number of MCMC iterations at *each* time step. As making a judicious choice of these parameters at each time step is a daunting task, it is common to use the same MCMC kernel parameters through the entire sequence. Likewise, it is common to use the same number of MCMC kernel iterations at each time step. Unfortunately, using a fixed scheme for kernel parameters and number of iterations does not take into account that the targets

can become more (or less) difficult to sample from in later iterations. While SMC methods retain their convergence properties regardless of these factors, one would ideally like to choose them in a way that is in some sense optimal at each iteration, especially if we aim to make a fair comparison between different SMC methods. In this section we present some novel ways of approximately achieving this goal in practice.

5.1 Choice of Kernel Parameters

One of the major advantages of SMC samplers over MCMC and NS is the ability to use the population of particles at each time step to inform the choice of MCMC kernel parameters. For example, it is common (see, for example, [Chopin and Ridgway \(2017\)](#)) to use the sample covariance matrix of the particles $\hat{\Sigma}$ (an estimator of the global covariance Σ) in local proposals. However, when it comes to more general kernel parameter selection, it unfortunately remains common practice to use the same fixed kernel parameter across all time steps.

For MCMC samplers, [Pasarica and Gelman \(2010\)](#) propose to select kernel parameters by maximizing the *expected square jump distance* (ESJD) for a single MCMC iteration, which is equivalent to minimizing the first order (lag-1) autocorrelation. For current state \mathbf{X}_{curr} and state after an MCMC iteration \mathbf{X}_{new} , the ESJD is

$$\text{ESJD} := \mathbb{E} \|\mathbf{X}_{\text{new}} - \mathbf{X}_{\text{curr}}\|^2,$$

where $\|\cdot\|$ denotes some norm, and \mathbf{X}_{curr} is distributed according to the target density. In the context of Metropolis–Hastings MCMC with proposal density $q(\mathbf{x}' | \mathbf{x})$, the ESJD is given by

$$\mathbb{E}[\|\mathbf{X}_{\text{prop}} - \mathbf{X}_{\text{curr}}\|^2 \alpha_{\text{MH}}(\mathbf{X}_{\text{curr}}, \mathbf{X}_{\text{prop}})],$$

where $\mathbf{X}_{\text{prop}} \sim q(\mathbf{x}' | \mathbf{x}_{\text{curr}})$, and $\alpha_{\text{MH}}(\mathbf{X}_{\text{curr}}, \mathbf{X}_{\text{prop}})$ is the Metropolis Hastings acceptance probability of moving from \mathbf{X}_{curr} to the proposal \mathbf{X}_{prop} . In this context, we can estimate the ESJD via

$$\|\mathbf{X}_{\text{prop}} - \mathbf{X}_{\text{curr}}\|^2 \alpha_{\text{MH}}(\mathbf{X}_{\text{curr}}, \mathbf{X}_{\text{prop}}).$$

[Fearnhead and Taylor \(2013\)](#) propose an adaptive SMC sampler that uses the estimated ESJD in its selection of MCMC kernel parameters. The method starts with an initial population of kernel parameters which is used in the first mutation step. After the first and all subsequent mutation steps, the population of kernel parameters is resampled according to ESJD and then jittered. Generally there will be many poor-performing kernel parameters in the early iterations, and this may lead to poor mixing that can affect later distributions. Moreover, the kernel parameters that were roughly optimal for the previous iteration are

used as a basis for those in the next iteration. If the targets change in a way that significantly affects the optimal tuning parameter (for example, the separation of modes due to a new threshold in NS-SMC), then poor results can be expected. To avoid the use of many poor choices of parameters in early iterations and to make selection robust to changes in the optimal tuning parameter between iterations, we opt to select a single optimal tuning parameter per target based on a single MCMC iteration on all the particles.

Specifically, to automate the selection of a single optimal tuning parameter, we do the following. We specify a finite set of values for the tuning parameter and at each $t > 1$, each particle is randomly assigned one of these choices. We then perform a single MCMC iteration per particle and record the corresponding estimate of the ESJD. We follow both [Pasarica and Gelman \(2010\)](#) and [Fearnhead and Taylor \(2013\)](#) in the choice of *Mahalanobis distance* as a norm, i.e., $\|\mathbf{y}\|_{\hat{\Sigma}} := \sqrt{\mathbf{y}^T \hat{\Sigma}^{-1} \mathbf{y}}$, where $\hat{\Sigma}$ is an estimate of the global covariance matrix obtained from the particle positions.

The kernel parameter that produces the highest median estimated ESJD per target evaluation is selected and the remaining MCMC repeats are subsequently performed. Our method works well in combination with the method for tuning the number of MCMC repeats (which is explained shortly) and we illustrate how these methods work in an example in [Figure 4](#).

Remark 3 When sampling from $\{\eta_t\}_{t=1}^T$ in NS-SMC using Metropolis-Hastings with proposal density $q(\mathbf{x}' | \mathbf{x})$, the acceptance probability becomes

$$\alpha_{\text{MH}} = \min \left\{ 1, \frac{\eta(\mathbf{x}')q(\mathbf{x}' | \mathbf{x})}{\eta(\mathbf{x})q(\mathbf{x} | \mathbf{x}')} \mathbb{I}\{\mathbf{x}' \in E_t\} \right\}.$$

While computing this quantity explicitly is required for estimating ESJD in the pilot run, we remark that when this is not required, it is more efficient to accept a proposal in two stages as follows. We *conditionally* accept with probability $\min \{1, \eta(\mathbf{x}')q(\mathbf{x}' | \mathbf{x})/\eta(\mathbf{x})q(\mathbf{x} | \mathbf{x}')\}$. Then, if a proposal has been conditionally accepted, we accept the proposal iff $\mathbf{x}' \in E_t$. This approach reduces the number of likelihood evaluations required for the same number of iterations, and is an additional benefit of NS-SMC samplers.

5.2 Choosing the Number of MCMC Iterations

Choosing the number of MCMC iterations per particle at each iteration in an efficient manner remains a challenging open problem. Computational effort aside, one would like the particles to be close to independent. However, in practice, we consider this too lofty a goal. For example, in the case of temperature annealing, at the final move step, achieving this is equivalent to ensuring burn-in for N standard MCMC samplers for π . Alternatively,

one could focus less on attempting to ensure particle independence and instead try to ensure that there are N *unique* particles after the move step. For example, one could perform a single iteration of Slice Sampling to guarantee unique particles, but the average distance moved may be extremely small. In practice, a balance must be struck.

[Drovandi and Pettitt \(2011\)](#) propose a formula to estimate the number of repeats required to move particles at least once with a specified probability in the context of a Metropolis–Hastings MCMC move step. The formula uses an average acceptance probability which can be estimated from the previous SMC iteration, or calculated with a single MCMC repeat for the current target as in [South et al \(2016\)](#). Although this method is relatively simple to implement, it does not consider the quality of the proposed moves in terms of jumping distance. Large proposals that are accepted with small probability are given more repeats than small proposals that are accepted with high probability. In practice, this method is effective at ensuring a collection of unique particles, but the uniqueness of particles does not guarantee quality of the particle approximation. Furthermore, this method is not sensible in the context of moves with guaranteed acceptance, such as Slice Sampling. A second approach, given by [Ridgway \(2016\)](#), is to check for convergence or stabilization of the moves. The sum (over the particles) of the absolute move distances at each MCMC is recorded and one should iterate until this quantity stabilizes. However, suggestions are not given as to precisely what defines stabilization of this quantity, or how to check for this in an automatic manner. Furthermore, we find that if the resampled particles already represent a reasonable approximation to the target, as they do in the context of NS–SMC and TA–SMC with a sufficiently large ρ/α , then stabilization becomes even more difficult to determine.

In light of this, we propose an approach that allows the particles to perform a reasonable level of exploration. Define the *expected jump distance* for a single MCMC repeat and particle to be $J := \mathbb{E} \|\mathbf{X}_{\text{new}} - \mathbf{X}_{\text{curr}}\|_{\hat{\Sigma}}$. In Metropolis–Hastings (MH) MCMC, we can estimate J via

$$\|\mathbf{X}_{\text{prop}} - \mathbf{X}_{\text{curr}}\|_{\hat{\Sigma}} \alpha_{\text{MH}}(\mathbf{X}_{\text{curr}}, \mathbf{X}_{\text{prop}}).$$

For R iterations of an MCMC kernel, we have

$$\hat{J}(R) := \sum_{r=1}^R \hat{J}_r, \tag{31}$$

where \hat{J}_r denotes the estimate of expected jump distance obtained from the r -th iteration. For some specified quantity J_{desired} , we propose to continue iterating the MCMC kernel over all particles until a specified proportion of the particles satisfies $\hat{J}(R) > J_{\text{desired}}$. Note that the sum in (31) is over the MCMC iterations, rather than over the particles as was

the case in [Ridgway \(2016\)](#).

Our proposed method requires a choice of the proportion of particles as well as a choice of J_{desired} . Both can be chosen based on how conservative the move step should be. In all of our examples, we choose J_{desired} online by using the (weighted) mean Mahalanobis distance between particles before resampling. We continue to perform repeats until 50% of particles satisfy $\hat{J}(R) > J_{\text{desired}}$.

This method can be implemented to tune the repeats online and is well-suited to comparing/selecting different MCMC kernels that differ in terms of acceptance rate and jumping distance. Our experiments also find it to be more robust to sub-optimal tuning parameters than the acceptance probability based method of [Drovandi and Pettitt \(2011\)](#). While the method does not account for possible back-and-forth behaviour of the sampler, we find that it works well for all samplers across both SMC methods. We note that the underlying method of iterating until a desired criteria is observed for a specified number of particles is quite general and can encompass a wide range of goals for the move step. For example, using other measures of distance is possible, as is considering the sum of *actual* distances moved.

It is important to note that choosing the number of MCMC repeats online is a form of adaptivity. Thus, we recommend this approach is only used in the pilot run (where the sequence of distributions and kernel parameters are also chosen adaptively), in order to determine the number of repeats for fixed SMC runs.

6 COMPARISON WITH TA-SMC

While we saw in [Section 4.3](#) that variants of NS-SMC are capable of handling phase transitions as well as NS, with less bias, the question of how NS-SMC compares with TA-SMC arises. In this section, we compare the different SMC approaches on two challenging Bayesian statistical inference problems.

Our intention is not necessarily to demonstrate the superiority of our proposed method over TA-SMC. Given the variety of possible parameters for SMC (i.e., N and ρ/α) as well as many possible MCMC kernels, methods of tuning them, and choices for number of MCMC repeats at each iteration, there most likely exists an appropriate choice of these factors for any given problem that will allow one method to outperform the other. Thus, we instead aim to simply make our best efforts using our experience with SMC to get the best out of both algorithms in an automated manner, and observe the results.

Since we have introduced NS-SMC as a way to overcome theoretical issues in NS, particularly when there is particle dependency, in our experiments we restrict ourselves to MCMC kernels. We point out that region samplers are only a valid η_t -invariant kernel at each t if they are able to make proposals on *all* of E_t . However, as discussed earlier, one cannot guarantee this.

To give a fair comparison, we considered three choices of MCMC kernels:

1. The classic Random Walk (RW) sampler, where proposals take the form $\mathbf{Y} \sim \mathcal{N}(\mathbf{X}, h^2\hat{\Sigma})$. This sampler was selected as it a widely applicable and common sampler.
2. The Metropolis Adjusted Langevin Algorithm (MALA), with proposals $\mathbf{Y} \sim \mathcal{N}(\mathbf{X} + \nabla_{\mathbf{x}} \log \tilde{\pi}(\mathbf{X}), h^2\hat{\Sigma})$, where $\tilde{\pi}$ is the target distribution. MALA is applicable when the derivatives of the log target with respect to the parameters are available in analytic form or can be unbiasedly estimated. One of the strengths of TA-SMC is that there is rich literature of samplers that are straightforward to apply (see [Kroese et al \(2011, Ch.6\)](#) for example). We feel it important to include MALA because it is more suited to unconstrained targets. The derivatives of the log likelihood are not used in NS-SMC MALA because the likelihood is only used in defining the constraints.
3. Slice Sampling ([Neal \(2003\)](#)), specifically the `slicesample` function from the Statistics and Machine Learning Toolbox in MATLAB . This implementation is based on the basic stepping out and shrinkage implementation described in [Neal \(2003\)](#). The step out distance in each dimension is chosen to be $w\hat{\sigma}_i$ where $\hat{\sigma}_i$ is the standard deviation of the i th parameter estimated from the population of particles. Unlike RW and MALA, Slice Sampling is not disadvantaged by working in a constrained space, as it requires constrained sampling regardless of the underlying distribution.

In this section, we wish to compare the two SMC methods in a setting resembling what is typically used in practice. Thus, we use the stratified resampling scheme of [Kitagawa \(1996\)](#) for both methods as this results in lower variance over that of the simpler multinomial scheme. For recent convergence results and justification for choosing this scheme over other alternatives, see [Gerber et al \(2017\)](#).

We conduct an initial pilot run to determine the sequence of distributions, before executing 100 runs with fixed choice of distributions and MCMC parameters/repeats determined from the pilot run (details of how these were chosen are given in the next section). In the pilot run, we used $\rho = 0.5$ for NS-SMC, and $\alpha = 0.5$ for TA-SMC, as this leads to the same proportion of ESS out of the total and in our experience these choices tend to generally perform well. As the same choice of ρ and α typically yields more iterations (i.e., target

distributions) for NS-SMC, in the pilot run we used $N = 4 \cdot 10^4$ and $N = 10^4$ samples for TA-SMC and NS-SMC, respectively, in order to keep the number of likelihood evaluations roughly equivalent across methods. For 100 fixed runs, we used $N = 4 \cdot 10^3$ and $N = 10^3$ samples for TA-SMC and NS-SMC, respectively. We felt it was important to use a larger number of samples in the pilot run in order to tune the MCMC kernels better, and to ensure differences in performance were not simply due to poor selection of tuning parameters.

Figure 4 illustrates the typical behaviour of our methods for the selection of kernel parameter and repeats is what one should expect for TA-SMC. Specifically, as the target becomes increasingly complex, the number of repeats increases, and smaller step sizes are made. Furthermore, TA-SMC MALA makes larger steps and therefore uses fewer repeats. See Appendix C and ?? for further plots.

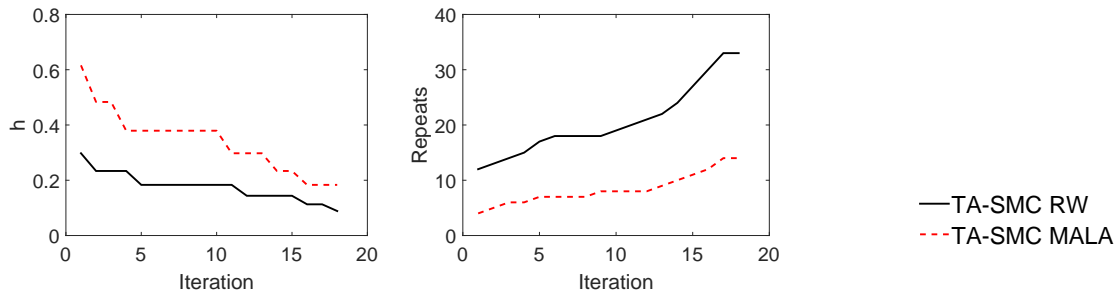


Figure 4: Tuning parameter and repeats selection for TA-SMC RW and TA-SMC MALA for the challenging three component factor analysis model considered in Section 6.

When comparing results, we examine estimates of posterior means, posterior lower (2.5%) and upper (97.5%) quantiles, and model evidence. We measure efficiency in terms of *work-normalized variance* (WNV), specifically the variance of the quantity of interest (a measure of statistical efficiency) multiplied by the number of likelihood evaluations (a measure of computational efficiency). The relative WNV measure shown in some tables is the WNV for that method divided by the WNV for TA-SMC RW. Thus, smaller values are considered evidence of superior performance.

6.1 Example 1: Factor Analysis

This model choice example demonstrates how NS-SMC and TA-SMC perform on three different posterior distributions of varying complexity. We consider the monthly exchange rate dataset used in West and Harrison (1997), where exchange rates (relative to the British Pound) of six different currencies were collected from January 1975 to December 1986, for a

total of $n = 143$ observations. As in [Lopes and West \(2004\)](#), we model the covariance of the (standardized) monthly-differenced exchange rates, using a factor analysis model, i.e., for $k \leq d$ factors, our data is assumed to be drawn independently from a $\mathcal{N}(\mathbf{0}, \Omega)$ distribution, where Ω can be factorized as $\Omega = \mathbf{L}\mathbf{L}^\top + \Lambda$, for $\mathbf{L} \in \mathbb{R}^{d \times k}$ lower triangular with positive diagonal elements, and Λ a diagonal matrix with diagonal given by $\boldsymbol{\lambda} \in \mathbb{R}_+^d$. Thus, we have that for each additional factor in the model, we introduce $6(k+1) - k(k-1)/2$ additional parameters, giving 12, 17, and 21 parameters for one, two and three factors, respectively. For priors, we follow [Lopes and West \(2004\)](#) and specify

$$\begin{aligned} L_{ij} &\sim \mathcal{N}(0, 1), \quad i < j, i = 1, \dots, k, j = 1, \dots, d \\ L_{ii} &\sim \mathcal{TN}_{(0, \infty)}(0, 1), \quad i = 1, \dots, k \\ \lambda_i &\sim \mathcal{IG}(1.1, 0.05), \quad i = 1, \dots, d, \end{aligned}$$

where $\mathcal{TN}_{(0, \infty)}(\boldsymbol{\mu}, \Sigma)$ denotes a $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ distribution truncated to the interval $(0, \infty)$, and $\mathcal{IG}(\alpha, \beta)$ denotes the Inverse-Gamma distribution with probability density function

$$f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp(-\beta/x), \quad x > 0.$$

In order to facilitate improved sampling, we take a log transform of L_{ii} for $i = 1, \dots, k$, which obviates the need to deal with any constraints at all in TA-SMC.

The one factor posterior (FA1) is relatively easy to sample from in that the marginal densities are all unimodal. The two factor (FA2) posterior possesses highly separated modes that are challenging to capture for standard MCMC methods (for example, the reversible jump sampler of [Lopes and West \(2004\)](#) failed to capture this). Finally, the three factor posterior (FA3) contains an exceptionally complex landscape, as shown by [Figure 6](#).

We also include results from an extended ‘‘gold standard’’ run of TA-SMC, for which we used $N = 5 \times 10^4$, and the extremely conservative $\alpha = 0.999$ to ensure the particles adequately explored the space. The $\log p$ vs. $\log \mathcal{L}$ plots did not indicate the presence of any phase transitions in any of the three cases. Results for evidence estimation are shown in [Table 2](#) and [Figure 8](#), and results for posterior inference are given in [Appendix B](#). It appears that RW and slice kernels are more efficient for NS-SMC than TA-SMC for both of the challenging models. Given the earlier discussion on using MALA in a constrained space, it is not surprising to see that NS-SMC has performed poorly.

In the 2 and 3 component models, multimodality introduces additional difficulty. In SMC, the main issues with multimodality are that (a) resampling can change the proportion of particles in each mode and (b) many MCMC kernels do not correct for this by moving between modes. In NS-SMC, the constraints mean that modes become well separated

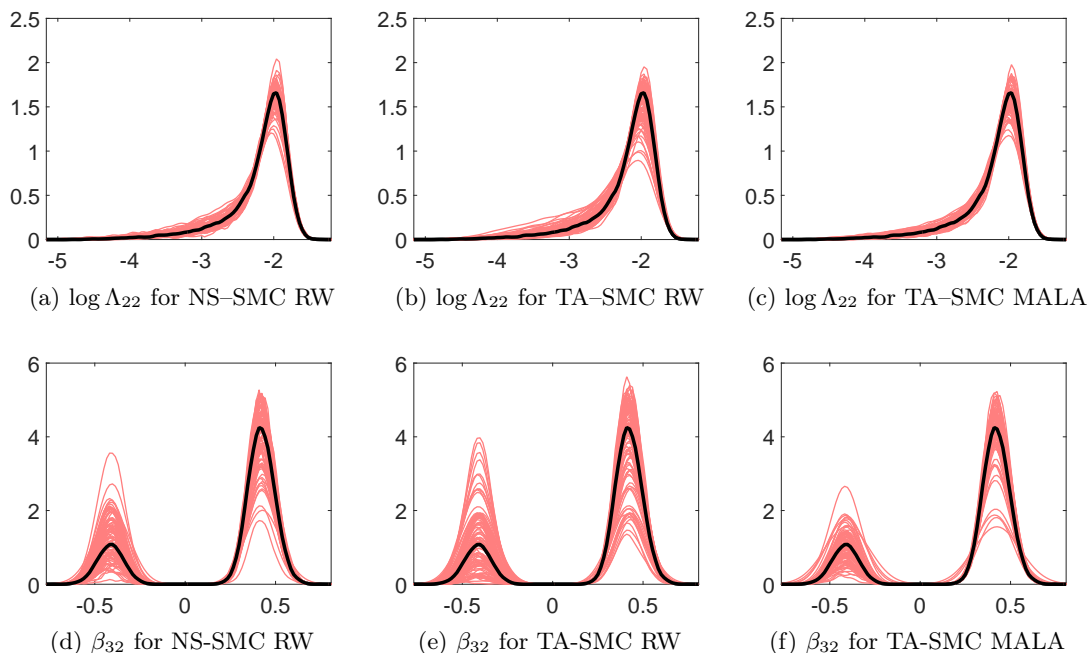


Figure 5: FA2 posterior marginal estimates for the gold standard and for 100 runs of NS-SMC and TA-SMC. Shown are parameters (a,b) $\log \Lambda_{22}$ which is highly skewed and (c,d) β_{32} which has well separated modes. Note that the proportion in each mode is more consistent in NS-SMC RW than in TA-SMC RW.

quickly which compounds problem (b). On the other hand, TA-SMC suffers more from problem (a) because if resampling removes all samples from a given mode, then unless the unlikely event that the mode is rediscovered occurs, it will not be captured at all by the particles on the final target (even with recycling methods as described in [Nguyen et al \(2014\)](#), the highest weights will be on the final few targets). This may explain why one method does not seem to significantly outperform the other in the 2 and 3 component models.

6.2 Example 2: Ordinary Differential Equation

Models for which the posterior density exhibits strong and complicated tail dependencies present a unique challenge for samplers. Thus, it is natural to consider to what extent NS-SMC is robust to these issues by testing it on such an example.

We consider a system of ordinary differential equations for modelling biochemical pathways [Girolami \(2008\)](#), specifically the following system of coupled ordinary differential equations

Table 2: Factor Analysis model evidence results for 100 runs. Efficiency factor is relative to TA-SMC RW.

Factors	Method	Sampler	$\log \widehat{\mathcal{Z}}$	avg. evals	relative WNV
One	TA-SMC	RW	-1014.28	7.3×10^5	1.0
		MALA	-1014.28	3.0×10^5	0.1
		Slice	-1014.27	9.5×10^5	2.6
	NS-SMC	RW	-1014.27	5.3×10^5	1.1
		MALA	-1014.24	8.2×10^5	2.2
		Slice	-1014.32	8.2×10^5	2.5
Two	TA-SMC	RW	-903.21	1.3×10^6	1.0
		MALA	-903.24	7.5×10^5	0.1
		SLICE	-903.38	1.3×10^6	2.5
	NS-SMC	RW	-903.23	1.2×10^6	0.3
		MALA	-903.02	1.9×10^5	1.9
		SLICE	-903.18	1.3×10^6	2.1
Three	TA-SMC	RW	-905.29	1.5×10^6	1.0
		MALA	-905.36	6.1×10^5	0.1
		SLICE	-905.02	1.8×10^6	11.8
	NS-SMC	RW	-905.39	1.7×10^6	0.4
		MALA	-905.40	1.4×10^6	0.7
		SLICE	-905.30	2.2×10^6	1.2

(ODEs)

$$\begin{aligned}
 \frac{dS}{dt} &= -k_1 S \\
 \frac{dD}{dt} &= k_1 S \\
 \frac{dR}{dt} &= \frac{-V_1 R S}{K m_1 + R} + \frac{V_2 R_{pp}}{K m_2 + R_{pp}} \\
 \frac{dR_{pp}}{dt} &= \frac{V_1 R S}{K m_1 + R} - \frac{V_2 R_{pp}}{K m_2 + R_{pp}}.
 \end{aligned}$$

Following [Girolami \(2008\)](#), Gamma priors are specified for all parameters,

$$\begin{aligned}
 k_1, V_1, K_{m_1}, V_2, K_{m_2}, \sigma &\sim \mathcal{G}(1, 1) \\
 S(0), R(0) &\sim \mathcal{G}(5, 0.2) \\
 D(0), R_{pp}(0) &\sim \mathcal{G}(1, 0.1).
 \end{aligned}$$

where $\mathcal{G}(\alpha, \beta)$ has the density

$$f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x), \quad x \geq 0.$$

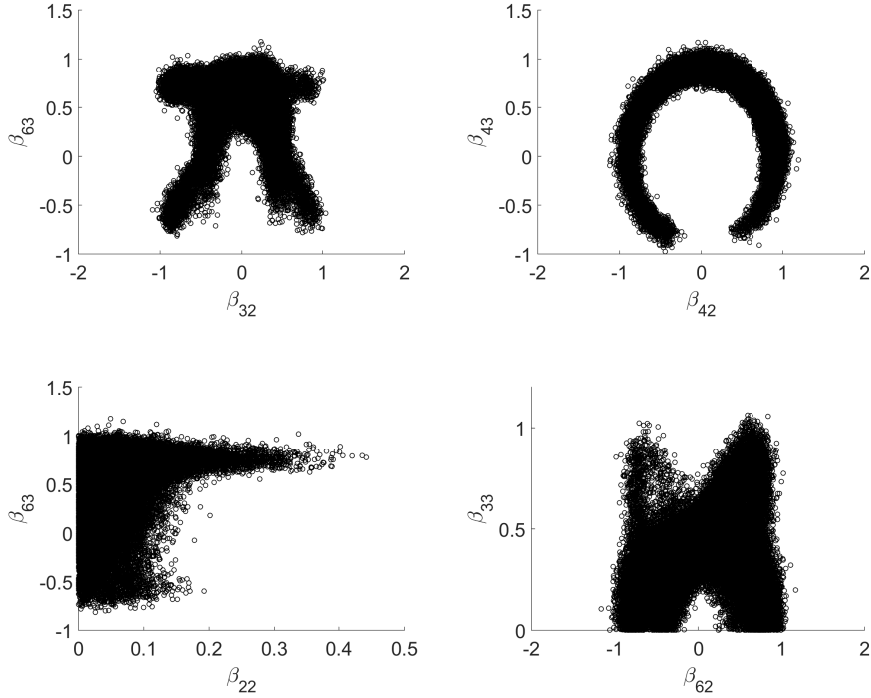


Figure 6: A selection of the most challenging bivariate distributions. Plots are FA3 bivariate posterior scatterplots from the gold standard run.

As in [Girolami \(2008\)](#), we generate a synthetic dataset using

$$y(t) \sim \mathcal{N}(R_{pp}(t), \sigma^2), \quad t = 0, 3, 6, \dots, 57,$$

where $\sigma = 0.02$, and $R_{pp}(t)$ is obtained via forward simulation of the model (this is a stiff system, so MATLAB's ODE15s solver is used) with

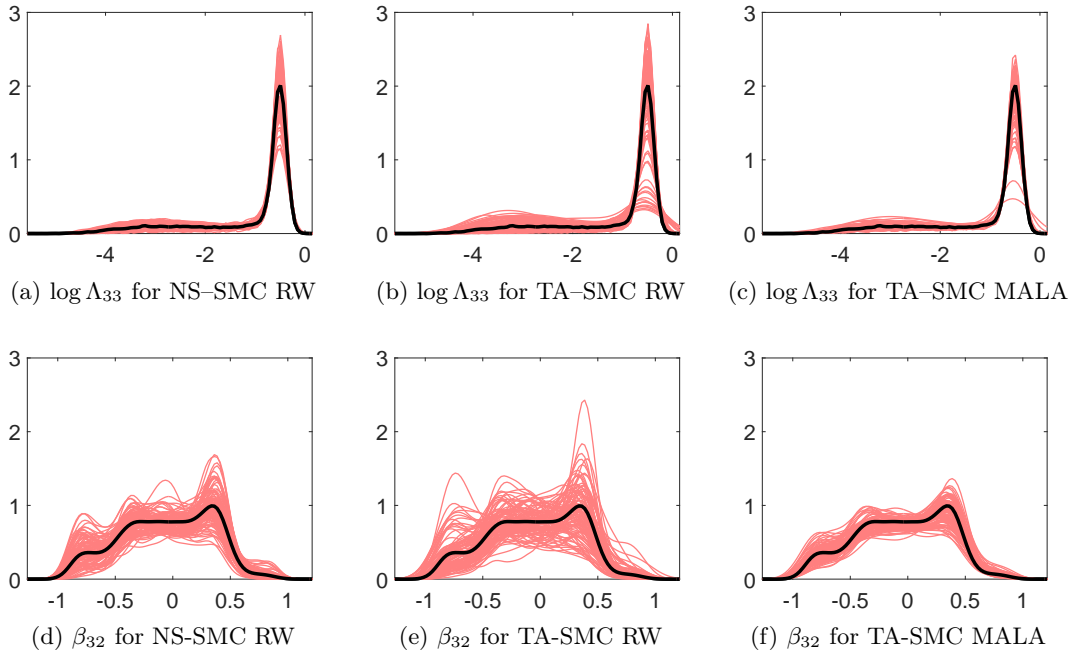


Figure 7: FA3 posterior marginal estimates for the gold standard (thick line) and for 100 runs of NS-SMC and TA-SMC (thin lines). Shown are parameters (a,b) $\log \Lambda_{33}$ which is highly skewed and (c,d) β_{32} which is multimodal.

$$\begin{bmatrix} k_1 \\ V_1 \\ K_{m_1} \\ K_{m_2} \\ V_2 \\ S(0) \\ D(0) \\ R(0) \\ R_{pp}(0) \end{bmatrix} = \begin{bmatrix} 0.05 \\ 0.2 \\ 0.1 \\ 0.1 \\ 0.1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} .$$

We perform all sampling on a transformed space where the natural logarithm is applied element-wise to each parameter, in order to remove the need to sample on a constrained space. Despite being only a nine-dimensional parameter space, sampling from the posterior density in this example is challenging due to complex tail dependencies.

The “gold standard” for this example is a 10^7 iteration random walk MCMC run, with a burn-in of 10^5 iterations and thinning by taking every 10^3 -th sample. This extended run

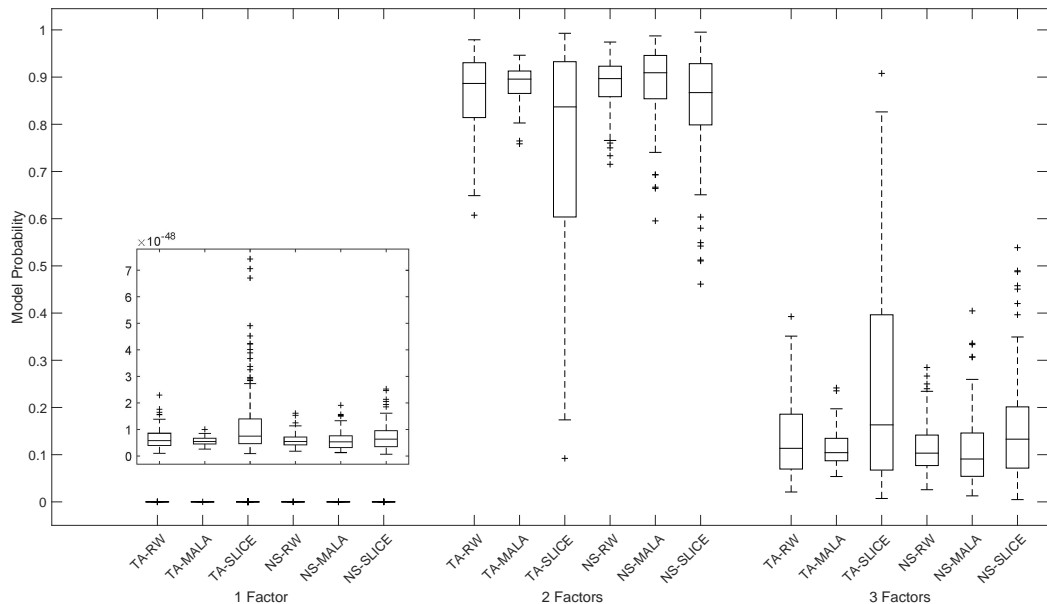


Figure 8: FA model probabilities based on 100 runs.

Table 3: ODE model evidence results for 100 runs.

Method	Sampler	$\log \hat{\mathcal{Z}}$	avg. evals	relative WNV
TA-SMC	RW	21.98	1.3×10^6	1.0
	MALA	21.85	9.6×10^5	8.1
	SLICE	22.20	2.2×10^6	13.0
NS-SMC	RW	22.15	2.1×10^6	3.1
	MALA	22.00	8.8×10^5	1.2
	SLICE	21.97	2.0×10^6	3.8

uses roughly 5-10 times the number of likelihood evaluations as any of the SMC samplers considered here.

From Figure 9 and Table 13 in Appendix B, it can be seen that both TA-SMC and NS-SMC fall short, somewhat surprisingly, in a very similar manner with respect to tail coverage for parameters $\log k_1$, $\log Km_2$ and $\log V_2$. Observe in Figure 9 that the occasional run produces a disproportionate amount of samples in the tails, indicating that the failure to obtain representative samples in the tails is a largely a manifestation of high variance.

Once again, the choice of MCMC kernel has more of an impact on evidence and posterior estimation than the choice of SMC method. An interesting case here is TA-SMC MALA which performs poorly both in terms of evidence estimation and posterior approximation.

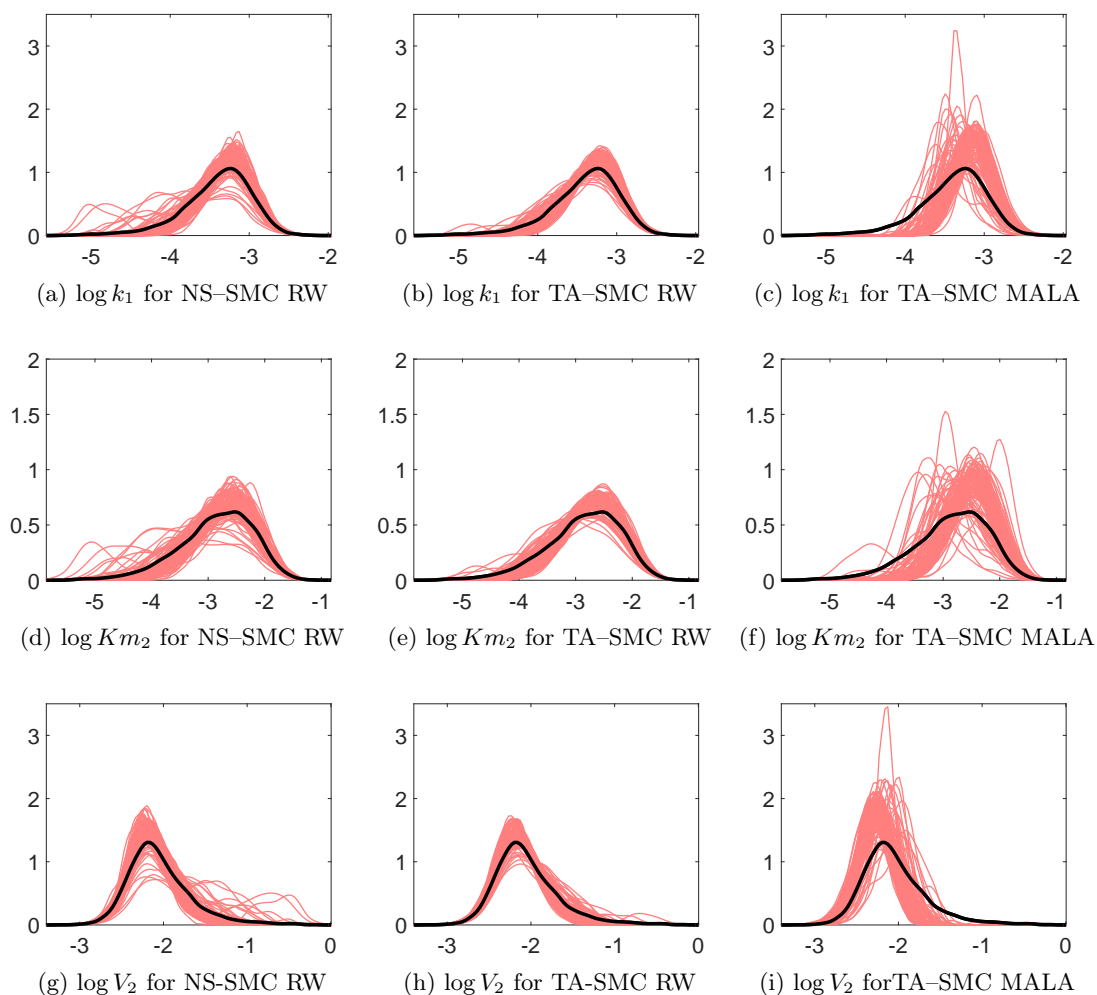


Figure 9: ODE posterior marginal estimates for the gold standard and for 100 runs of NS-SMC RW, TA-SMC RW and TA-SMC MALA. Shown are parameters (a,b,c) $\log k_1$ where lower tail coverage is an issue, (d,e,f) $\log Km_2$ where lower tail coverage is an issue, and (g,h,i) $\log V_2$ where upper tail coverage is an issue.

TA-SMC MALA makes proposals which are guided by the (estimated) global covariance and the derivatives of the log target. As this does not take local dependencies into account, the use of derivative information here actually results in something that performs *worse* than the RW sampler by a significant factor. In general, one must keep in mind that the use of additional derivative information does not necessarily translate into superior performance.

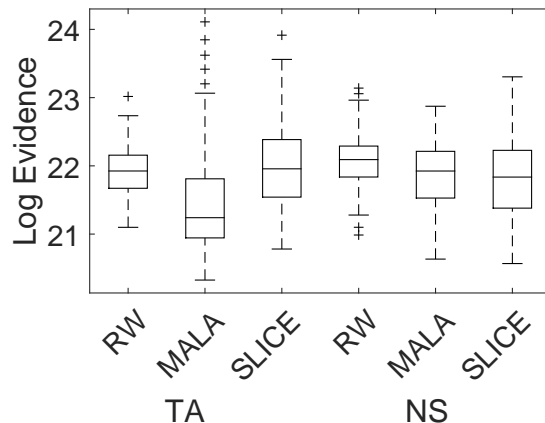


Figure 10: Boxplots of the log evidence for the ODE example based on 100 runs.

7 DISCUSSION

The results of our numerical study demonstrate that the NS-SMC approach is capable of performing well on very difficult problems. The results in Section 6 indicate that the performance of the model evidence estimator is more a product of the performance of an MCMC kernel than of the overarching SMC method. However, as illustrated by the phase transition example in Section 4.3, there are problems for which NS-SMC is preferable. Such cases asides, the question whether SMC is preferable using the TA or NS approach is really one of whether it is preferable to sample (relatively) easy distributions subject to a constraint or to sample potentially difficult distributions. Overall, our results provide evidence that the NS approach to SMC has its merits and deserves further attention.

In terms of extensions and variants to NS-SMC from the NS literature, we identify several promising directions. An analogous SMC method to the Diffusive Nested Sampling of Brewer et al (2011) may be possible through the use of specifying a sequence of mixtures of densities of the form (21). Such an approach may increase robustness to the tail coverage issues such as those in the ODE example, and would perhaps improve the performance of NS-SMC in multimodal settings. A NS-SMC version of the ellipsoidal nested importance sampling method of Chopin and Robert (2010) is straightforward. We note however that as $\mathbb{P}_\eta(\mathbf{X} \in \check{E}_t)$ is easily computed in this setting, and as exactly sampling from η constrained to the shells \check{E}_t is possible (as Nested Importance Sampling reformulates the problem so that η is Gaussian and \check{E}_t are ellipsoidal regions), the method reduces to a stratified form of importance sampling.

Conversely, improvements to NS-SMC may also be made by borrowing from the SMC literature. Again, there are several exciting possible directions in this regard. For example, use of the particle population at each stage to construct independence samplers as in [South et al \(2016\)](#), which not only are capable of providing highly effective MCMC kernels at each iteration, but have the added advantage of allowing one to recycle proposals to further improve estimates. Furthermore, with the absence of a deterministic quadrature rule, and Monte Carlo estimators in their place, NS-SMC may be improved further by control variate techniques such as zero-variance control variates ([Mira et al \(2013\)](#)) or control functionals ([Oates et al \(2017\)](#)).

In terms of theoretical developments, convergence results for ANS-SMC may be possible by extending the results of [Cérou and Guyader \(2016\)](#) for adaptive multilevel splitting, and would require taking into account the dual importance sampling at each iteration, as well as the random termination condition. Convergence results for INS (and in turn NS with MCMC) remain difficult due to the combined adaptivity and special choice of move step, however the connection of NS to SMC provides a new way of looking at the problem.

Finally, as the performance of NS-SMC largely depends on the performance of the MCMC kernel used in the move step, further research on how to best sample from distributions subject to complicated constraints is also of interest. Such samplers are also of interest for SMC methods for Approximate Bayesian Computation (see for example, [Del Moral et al \(2012a\)](#)).

ACKNOWLEDGEMENTS

This work has been supported by the Australian Research Council Centre of Excellence for Mathematical & Statistical Frontiers (ACEMS), under grant number CE140100049. Christopher Drovandi is also supported by the Discovery Program of the Australian Research Council. We thank Michael Betancourt, Brandon Brewer, and Zdravko Botev for correspondence and discussions related to NS. We also extend a special thanks to Christian Robert, whose many blog posts on NS helped influence this work, and played a large part in inspiring it.

References

- Baldock RJN (2017) Classical Statistical Mechanics with Nested Sampling. Springer theses, Springer, Cham
- Beskos A, Jasra A, Kantas N, Thiery A (2016) On the convergence of adaptive sequential monte carlo methods. *The Annals of Applied Probability* 26(2):1111–1146

- Botev Z, Kroese D (2008) An efficient algorithm for rare-event probability estimation, combinatorial optimization, and counting. *Methodology and Computing in Applied Probability* 10(4):471–505
- Botev ZI, Kroese DP (2012) Efficient Monte Carlo simulation via the generalized splitting method. *Statistics and Computing* 22(1):1–16
- Brewer BJ (2014) Inference for trans-dimensional Bayesian models with diffusive nested sampling. arXiv:14113921
- Brewer BJ, Pártay LB, Csányi G (2011) Diffusive nested sampling. *Statistics and Computing* 21(4):649–656
- Buchner J (2016) A statistical test for nested sampling algorithms. *Statistics and Computing* 26(1):383–392
- Burkoff N, Várnai C, Wells SA, Wild DL (2012) Exploring the energy landscapes of protein folding simulations with Bayesian computation. *Biophysical Journal* 102(4):878–886
- Cérou F, Guyader A (2016) Fluctuation analysis of adaptive multilevel splitting. *Ann Appl Probab* 26(6):3319–3380
- Cérou F, Moral P, Furon T, Guyader A (2012) Sequential Monte Carlo for rare event estimation. *Statistics and Computing* 22(3):795–808
- Chopin N (2002) A sequential particle filter method for static models. *Biometrika* 89(3):539–552
- Chopin N, Ridgway J (2017) Leave Pima indians alone: Binary regression as a benchmark for Bayesian computation. *Statist Sci* 32(1):64–87
- Chopin N, Robert CP (2010) Properties of nested sampling. *Biometrika* 97(3):741–755
- Del Moral P (2004) *Feynman-Kac Formulae Genealogical and Interacting Particle Systems with Applications*. Probability and its Applications, A Series of the Applied Probability Trust, Springer New York, New York, NY
- Del Moral P (2013) *Mean field simulation for Monte Carlo integration*. Monographs on statistics & applied probability ; 126, Taylor & Francis, Boca Raton
- Del Moral P, Doucet A (2014) *Particle methods: an introduction with applications*. ESAIM: Proceedings 44:1–46
- Del Moral P, Doucet A, Jasra A (2006) Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68(3):411–436
- Del Moral P, Doucet A, Jasra A (2012a) An adaptive sequential monte carlo method for approximate bayesian computation. *Statistics and Computing* 22(5):1009–1020

- Del Moral P, Doucet A, Jasra A (2012b) On adaptive resampling strategies for sequential Monte Carlo methods. *Bernoulli* 18(1):252–278
- Doucet A, Johansen AM (2011) A Tutorial on Particle filtering and smoothing: Fifteen years later. *The Oxford handbook of nonlinear filtering* (December 2008):656–705
- Drovandi CC, Pettitt AN (2011) Likelihood-free Bayesian estimation of multivariate quantile distributions. *Computational Statistics and Data Analysis* 55(9):2541–2556
- Evans M (2007) Bayesian statistics 8. Oxford University Press, New York, chap Discussion of Nested sampling for Bayesian computations by John Skilling, pp 491–524
- Fearnhead P, Taylor BM (2013) An adaptive sequential Monte Carlo sampler. *Bayesian analysis* 8(2):411–438
- Feroz F, Hobson MP (2008) Multimodal nested sampling: an efficient and robust alternative to Markov Chain Monte Carlo methods for astronomical data analyses. *Monthly Notices of the Royal Astronomical Society* 384(2):449–463
- Friel N, Pettitt AN (2008) Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70(3):589–607
- Gelman A, Roberts GO, Gilks WR, et al (1996) Efficient Metropolis jumping rules. *Bayesian statistics* 5(599-608):42
- Gerber M, Chopin N, Whiteley N (2017) Negative association, ordering and convergence of resampling methods. *arXiv:170701845* pp 1–24
- Girolami M (2008) Bayesian inference for differential equations. *Theoretical Computer Science* 408(1):4–16
- Gordon NJ, Salmond DJ, Smith AF (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In: *IEE Proceedings F (Radar and Signal Processing)*, IET, vol 140, pp 107–113
- Guyader A, Hengartner N, Matzner-Løber E (2011) Simulation and estimation of extreme quantiles and extreme probabilities. *Applied Mathematics & Optimization* 64(2):171–196
- Higson E, Handley W, Hobson M, Lasenby A (2018) Sampling errors in nested sampling parameter estimation. *Bayesian Analysis* pp 1–24, DOI 10.1214/17-BA1075, (Advance Publication)
- Jasra A, Stephens DA, Doucet A, Tsagaris T (2011) Inference for Lévy-Driven Stochastic Volatility Models via Adaptive Sequential Monte Carlo. *Scandinavian Journal of Statistics* 38(1):1–22

- Kahn H, Harris TE (1951) Estimation of particle Transmission by Random Sampling. National Bureau of Standards Applied Mathematics Series 12:27–30
- Kitagawa G (1996) Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics* 5(1):1–25
- Kroese DP, Taimre T, Botev ZI (2011) *Handbook of Monte Carlo methods*. John Wiley and Sons, New York
- Liu JS (2001) *Monte Carlo Strategies in Scientific Computing* by Jun S. Liu. Springer Series in Statistics, Springer New York : Imprint: Springer, New York, NY
- Lopes HF, West M (2004) Bayesian model assessment in factor analysis. *Statistica Sinica* 14(1):41–67
- Mira A, Solgi R, Imparato D (2013) Zero variance Markov chain Monte Carlo for Bayesian estimators. *Statistics and Computing* 23(5):653–662
- Murray I (2007) *Advances in Markov chain Monte Carlo methods*. PhD thesis, Gatsby computational neuroscience unit, University College London
- Murray I, MacKay DJC, Ghahramani Z, Skilling J (2005) Nested sampling for Potts models. pp 947–954
- Neal RM (2001) Annealed importance sampling. *Statistics and Computing* 11(2):125–139
- Neal RM (2003) Slice sampling. *The Annals of Statistics* 31(3):705–741
- Nguyen TLT, Septier F, Peters GW, Delignon Y (2014) Improving SMC sampler estimate by recycling all past simulated particles. In: *Statistical Signal Processing (SSP), 2014 IEEE Workshop on, IEEE*, pp 117–120
- Oates CJ, Girolami M, Chopin N (2017) Control functionals for Monte Carlo integration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79(3):695–718
- Pasarica C, Gelman A (2010) Adaptively scaling the Metropolis Hastings algorithm using expected squared jumped distance. *Statistica Sinica* 20(1):343–364
- Přartay LB, Bartřšk AP, Csřny G (2014) Nested sampling for materials: the case of hard spheres. *Physical review E, Statistical, nonlinear, and soft matter physics* 89(2)
- Ridgway J (2016) Computation of Gaussian orthant probabilities in high dimension. *Statistics and Computing* 26(4):899–916
- Robert C, Casella G (2004) *Monte Carlo Statistical Methods*, 2nd edn. Springer-Verlag, New York

- Rubinstein RY, Kroese DP (2017) *Simulation and the Monte Carlo Method*, 3rd edn. John Wiley & Sons, New York
- Skilling J (2006) Nested sampling for general Bayesian computation. *Bayesian Analysis* 1(4):833–859
- South L, Pettitt AN, Drovandi C (2016) Sequential Monte Carlo for Static Bayesian Models with Independent MCMC Proposals. QUT ePrints pp 1–52, URL <https://eprints.qut.edu.au/101729/>
- Vegetti S, Koopmans LVE (2009) Bayesian strong gravitational-lens modelling on adaptive grids: objective detection of mass substructure in galaxies. *Monthly Notices of the Royal Astronomical Society* 392(3):945–963
- Veitch J (2015) Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library. *Physical Review D (Particles, Fields, Gravitation and Cosmology)* 91(4)
- Walter C (2017) Point process-based Monte Carlo estimation. *Statistics and Computing* 27(1):219–236
- West M, Harrison J (1997) *Bayesian forecasting and dynamic models* / Mike West, Jeff Harrison., 2nd edn. Springer series in statistics, Springer, New York

A Theoretical Properties

For the purposes of theoretical analysis, SMC algorithms can be interpreted as interacting particle approximations to a flow of associated Feynman–Kac measures. We proceed using the convention in the main text and [Del Moral et al \(2006\)](#) that $t \geq 1$, as opposed to $t \geq 0$. We note this point as the latter is typically used in the analysis of Feynman–Kac flows. However, the difference is simply one of presentation.

Consider the sequence of densities η_1, \dots, η_T defined in the nested manner described at the beginning of Section 4. [Cérou et al \(2012, Proposition 2\)](#) show that the associated measures have the Feynman–Kac representation

$$\eta_t(f) = \mathbb{E}_{\eta_t}[f(\mathbf{X})] = \frac{\mathbb{E}[f(\mathbf{X}_t) \prod_{p=1}^{t-1} \mathbb{I}\{\mathbf{X}_p \in E_{p+1}\}]}{\mathbb{E}[\prod_{p=1}^{t-1} \mathbb{I}\{\mathbf{X}_p \in E_{p+1}\}]},$$

where f is any test function, and $(\mathbf{X}_p)_{p=1}^t$ is a Markov chain such that $\mathbf{X}_1 \sim \eta$. Precise details regarding the transition kernel of this time-inhomogeneous chain can be found in

Cérou et al (2012, Section 2). However, the key aspect is that the kernel K_t that governs transitions from \mathbf{X}_{t-1} to \mathbf{X}_t is η_t -invariant.

For $t = 1, \dots, T$, we thus have the *unnormalized* and *normalized Feynman–Kac measures*, given by

$$\gamma_t(\varphi) := \mathbb{E} \left[f(\mathbf{X}_t) \prod_{p=1}^{t-1} \mathbb{I}\{\mathbf{X}_p \in E_{p+1}\} \right]$$

and $\eta_t(f) := \gamma_t(f)/\gamma_t(1)$,

respectively.

The population of particles in NS–SMC (equivalently, the fixed levels algorithm of Cérou et al (2012)) approximate these measures with the *particle approximation* measures

$$\gamma_t^N(f) := \underbrace{\left(\prod_{p=1}^{t-1} \eta_p^N(\mathbb{I}_{E_{p+1}}) \right)}_{\gamma_t^N(1) = \widehat{\mathcal{P}}_t} \underbrace{\left(\frac{1}{N} \sum_{k=1}^N f(\mathbf{X}_t^k) \right)}_{\eta_t^N(f)}, \quad (32)$$

and

$$\eta_t^N(\varphi) := \gamma_t^N(f)/\gamma_t^N(1).$$

Feynman–Kac particle approximation measures have the well-known properties (see for example, Del Moral (2013)) that for all bounded measurable f : (1) $\mathbb{E}[\gamma_t^N(f)] = \gamma_t(f)$, and (2) as $N \rightarrow \infty$, $\gamma_t^N(f) \xrightarrow{\text{a.s.}} \gamma_t(f)$ and $\eta_t^N(f) \xrightarrow{\text{a.s.}} \eta_t(f)$. These properties are often presented in the context of multinomial resampling. However, they also hold for other resampling schemes that satisfy certain conditions; see Chapter 11.8 of Del Moral (2004).

We have that

$$\mathcal{P}_t = \gamma_t(1) = \prod_{p=1}^{t-1} \eta_p(\mathbb{I}_{E_{p+1}}) \quad \text{and} \quad \mathcal{Z}_t = \gamma_t(\mathcal{L}\mathbb{I}_{\check{E}_t}) = \mathcal{P}_t \underbrace{\eta_t(\mathcal{L}\mathbb{I}_{\check{E}_t})}_{\widehat{\mathcal{Z}}_t/\widehat{\mathcal{P}}_t}.$$

Henceforth, we proceed under the assumption that \mathcal{L} is bounded. As a result of Property

(1) it follows that the estimators

$$\widehat{\mathcal{Z}}_t = \gamma_t^N(\mathcal{L}\mathbb{I}_{\check{E}_t}) = \underbrace{\left(\prod_{p=1}^{t-1} \eta_p^N(\mathbb{I}_{E_{p+1}}) \right)}_{\widehat{\mathcal{P}}_t} \underbrace{\eta_t^N(\mathcal{L}\mathbb{I}_{\check{E}_t})}_{\mathcal{Z}_t/\mathcal{P}_t}, \quad \text{for } t = 1, \dots, T,$$

are unbiased. By linearity of expectation, it follows that $\widehat{\mathcal{Z}} = \sum_{t=1}^T \widehat{\mathcal{Z}}_t$ is an unbiased estimator of $\mathcal{Z} = \sum_{t=1}^T \mathcal{Z}_t$. By Property (2), we have that $\widehat{\mathcal{Z}}_t \xrightarrow{\text{a.s.}} \mathcal{Z}_t$, for $t = 1, \dots, T$, and thus $\widehat{\mathcal{Z}} \xrightarrow{\text{a.s.}} \mathcal{Z}$.

The NS-SMC estimator for $\pi(\varphi)$ is based on the simple identity:

$$\pi(\varphi) = \sum_{t=1}^T \frac{\mathcal{Z}_t}{\mathcal{Z}} \pi_t(\varphi) = \sum_{t=1}^T \frac{\mathcal{Z}_t}{\sum_{s=1}^T \mathcal{Z}_s} \cdot \frac{\eta_t(\mathcal{L}\mathbb{I}_{\check{E}_t}\varphi)}{\eta_t(\mathcal{L}\mathbb{I}_{\check{E}_t})},$$

which we approximate via

$$\pi^N(\varphi) = \sum_{t=1}^T \frac{\widehat{\mathcal{Z}}_t}{\sum_{s=1}^T \widehat{\mathcal{Z}}_s} \cdot \frac{\eta_t^N(\mathcal{L}\mathbb{I}_{\check{E}_t}\varphi)}{\eta_t^N(\mathcal{L}\mathbb{I}_{\check{E}_t})}.$$

Combining the almost-sure convergence of $\widehat{\mathcal{Z}}_1, \dots, \widehat{\mathcal{Z}}_T$ with Property (2), as $N \rightarrow \infty$ we have that $\pi^N(\varphi) \xrightarrow{\text{a.s.}} \pi(\varphi)$ for any bounded measurable function φ .

B Inference Results

We present posterior expectation and quantile results from 100 runs, $N = 4 \cdot 10^3$ for TA-SMC and $N = 10^3$ for NS-SMC for the Factor Analysis and ODE examples described in Section 6.

In brackets we report the ratio of (sample variance \times average number of evaluations of \mathcal{L}) for the associated method to that of TA-SMC with the Random Walk Sampler. Thus, lower values indicate lower work-normalized variance.

Table 4: Results for FA1 — Part 1 of 2

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$
$\log \Lambda_{11}$	Gold standard TA-SMC	RW	-0.23	-0.46	0.01
		MALA	-0.23(1.0)	-0.47(1.0)	0.01(1.0)
		SLICE	-0.23(0.2)	-0.46(0.3)	0.01(0.4)
	NS-SMC	RW	-0.23(4.5)	-0.47(1.4)	0.01(2.1)
		MALA	-0.23(1.3)	-0.47(0.5)	0.01(1.1)
		SLICE	-0.24(3.7)	-0.47(1.3)	0.01(2.2)
			-0.23(3.2)	-0.47(1.2)	0.01(1.7)
$\log \Lambda_{22}$	Gold standard TA-SMC	RW	-0.25	-0.48	-0.00
		MALA	-0.25(1.0)	-0.48(1.0)	-0.00(1.0)
		SLICE	-0.25(0.2)	-0.48(0.3)	-0.00(0.2)
	NS-SMC	RW	-0.25(3.4)	-0.48(1.3)	-0.00(1.3)
		MALA	-0.24(1.3)	-0.48(0.7)	-0.00(0.7)
		SLICE	-0.25(1.7)	-0.48(1.0)	-0.01(1.0)
			-0.25(4.6)	-0.48(1.2)	-0.00(1.5)
$\log \Lambda_{33}$	Gold standard TA-SMC	RW	-0.43	-0.66	-0.19
		MALA	-0.43(1.0)	-0.66(1.0)	-0.18(1.0)
		SLICE	-0.43(0.2)	-0.66(0.3)	-0.18(0.2)
	NS-SMC	RW	-0.43(3.0)	-0.66(1.6)	-0.18(1.5)
		MALA	-0.43(1.0)	-0.66(0.6)	-0.19(0.6)
		SLICE	-0.43(3.7)	-0.66(1.7)	-0.19(1.2)
			-0.43(4.0)	-0.66(1.4)	-0.18(1.7)
$\log \Lambda_{44}$	Gold standard TA-SMC	RW	-2.65	-3.51	-2.05
		MALA	-2.65(1.0)	-3.51(1.0)	-2.04(1.0)
		SLICE	-2.66(0.1)	-3.52(0.2)	-2.05(0.3)
	NS-SMC	RW	-2.65(4.4)	-3.51(2.2)	-2.05(2.2)
		MALA	-2.66(1.2)	-3.52(0.9)	-2.05(0.7)
		SLICE	-2.66(3.5)	-3.52(2.4)	-2.05(1.3)
			-2.65(3.6)	-3.51(2.4)	-2.05(1.4)
$\log \Lambda_{55}$	Gold standard TA-SMC	RW	-1.45	-1.73	-1.17
		MALA	-1.45(1.0)	-1.73(1.0)	-1.16(1.0)
		SLICE	-1.45(0.2)	-1.73(0.4)	-1.16(0.3)
	NS-SMC	RW	-1.45(3.1)	-1.73(1.8)	-1.16(1.9)
		MALA	-1.45(1.2)	-1.73(0.9)	-1.17(0.8)
		SLICE	-1.45(2.6)	-1.73(1.7)	-1.16(1.1)
			-1.45(3.4)	-1.73(1.9)	-1.16(1.4)
$\log \Lambda_{66}$	Gold standard TA-SMC	RW	-1.44	-1.73	-1.16
		MALA	-1.43(1.0)	-1.72(1.0)	-1.15(1.0)
		SLICE	-1.44(0.2)	-1.73(0.4)	-1.15(0.3)
	NS-SMC	RW	-1.44(4.5)	-1.73(1.5)	-1.15(1.7)
		MALA	-1.44(1.5)	-1.73(1.0)	-1.15(0.8)
		SLICE	-1.44(3.6)	-1.73(1.5)	-1.16(1.5)
			-1.44(5.3)	-1.72(1.5)	-1.15(1.7)

Table 5: Results for FA1 — Part 2 of 2

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$	
$\log \beta_{11}$	Gold standard		-0.81	-1.24	-0.48	
		TA-SMC	RW	-0.81(1.0)	-1.24(1.0)	-0.48(1.0)
			MALA	-0.81(0.2)	-1.23(0.3)	-0.48(0.3)
	SLICE		-0.81(4.1)	-1.23(2.5)	-0.48(2.0)	
	NS-SMC	RW	-0.82(1.1)	-1.24(1.2)	-0.48(0.7)	
		MALA	-0.81(3.2)	-1.23(2.3)	-0.48(1.3)	
SLICE		-0.81(4.2)	-1.24(2.5)	-0.48(1.3)		
β_{21}	Gold standard		0.46	0.30	0.63	
		TA-SMC	RW	0.46(1.0)	0.30(1.0)	0.63(1.0)
			MALA	0.46(0.3)	0.30(0.3)	0.63(0.4)
	SLICE		0.46(3.4)	0.30(1.6)	0.63(2.1)	
	NS-SMC	RW	0.46(1.8)	0.30(0.8)	0.63(1.0)	
		MALA	0.46(3.6)	0.30(1.1)	0.63(1.5)	
SLICE		0.46(4.5)	0.30(1.3)	0.63(2.3)		
β_{31}	Gold standard		0.59	0.44	0.75	
		TA-SMC	RW	0.59(1.0)	0.44(1.0)	0.75(1.0)
			MALA	0.59(0.2)	0.44(0.3)	0.75(0.3)
	SLICE		0.59(4.9)	0.44(2.2)	0.75(2.2)	
	NS-SMC	RW	0.59(1.6)	0.44(0.7)	0.75(1.2)	
		MALA	0.59(4.1)	0.44(1.2)	0.75(2.0)	
SLICE		0.59(7.3)	0.44(2.0)	0.75(3.2)		
β_{41}	Gold standard		0.97	0.86	1.10	
		TA-SMC	RW	0.97(1.0)	0.85(1.0)	1.10(1.0)
			MALA	0.97(0.2)	0.86(0.4)	1.11(0.2)
	SLICE		0.97(5.7)	0.86(2.4)	1.10(1.9)	
	NS-SMC	RW	0.97(1.3)	0.86(0.8)	1.10(0.7)	
		MALA	0.97(3.2)	0.86(1.4)	1.11(1.3)	
SLICE		0.97(8.4)	0.86(2.9)	1.10(2.5)		
β_{51}	Gold standard		0.88	0.76	1.02	
		TA-SMC	RW	0.88(1.0)	0.76(1.0)	1.02(1.0)
			MALA	0.88(0.2)	0.76(0.3)	1.02(0.3)
	SLICE		0.88(6.3)	0.76(2.5)	1.02(1.8)	
	NS-SMC	RW	0.88(1.7)	0.76(0.8)	1.02(0.9)	
		MALA	0.88(2.7)	0.76(1.2)	1.02(1.6)	
SLICE		0.88(8.4)	0.76(2.4)	1.02(2.7)		
β_{61}	Gold standard		0.88	0.76	1.02	
		TA-SMC	RW	0.88(1.0)	0.75(1.0)	1.02(1.0)
			MALA	0.88(0.2)	0.75(0.4)	1.02(0.4)
	SLICE		0.88(5.2)	0.75(2.5)	1.02(2.3)	
	NS-SMC	RW	0.88(1.2)	0.75(0.9)	1.02(0.8)	
		MALA	0.88(3.3)	0.76(2.0)	1.02(1.4)	
SLICE		0.88(7.6)	0.75(2.7)	1.02(3.3)		

Table 6: Results for FA2 — Part 1 of 3

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$
$\log \Lambda_{11}$	Gold standard TA-SMC	RW	-3.08	-4.41	-1.91
		MALA	-3.05(1.0)	-4.38(1.0)	-1.91(1.0)
		SLICE	-3.07(0.3)	-4.39(0.3)	-1.91(0.3)
	NS-SMC	RW	-2.94(7.4)	-4.10(6.6)	-1.97(13.0)
		MALA	-3.07(0.4)	-4.41(0.4)	-1.90(0.5)
		SLICE	-3.05(4.1)	-4.30(3.9)	-1.93(5.6)
			-3.03(3.6)	-4.25(3.2)	-1.95(6.6)
$\log \Lambda_{22}$	Gold standard TA-SMC	RW	-2.20	-3.50	-1.70
		MALA	-2.22(1.0)	-3.49(1.0)	-1.70(1.0)
		SLICE	-2.21(0.2)	-3.48(0.4)	-1.70(0.3)
	NS-SMC	RW	-2.29(10.8)	-3.31(5.5)	-1.73(22.6)
		MALA	-2.22(0.4)	-3.53(0.6)	-1.70(0.6)
		SLICE	-2.23(3.5)	-3.48(5.1)	-1.71(9.4)
			-2.23(3.5)	-3.38(3.9)	-1.72(8.0)
$\log \Lambda_{33}$	Gold standard TA-SMC	RW	-0.48	-0.70	-0.24
		MALA	-0.48(1.0)	-0.71(1.0)	-0.24(1.0)
		SLICE	-0.48(0.6)	-0.71(0.4)	-0.24(0.6)
	NS-SMC	RW	-0.48(56.6)	-0.71(25.3)	-0.23(34.0)
		MALA	-0.48(1.3)	-0.71(1.1)	-0.24(1.4)
		SLICE	-0.47(11.7)	-0.70(8.4)	-0.23(7.8)
			-0.48(23.0)	-0.70(18.9)	-0.24(18.1)
$\log \Lambda_{44}$	Gold standard TA-SMC	RW	-3.45	-4.49	-2.54
		MALA	-3.44(1.0)	-4.49(1.0)	-2.53(1.0)
		SLICE	-3.44(0.3)	-4.47(0.4)	-2.53(0.4)
	NS-SMC	RW	-3.36(16.6)	-4.26(14.8)	-2.52(22.2)
		MALA	-3.44(0.3)	-4.49(0.5)	-2.54(0.6)
		SLICE	-3.45(7.6)	-4.43(9.7)	-2.56(7.3)
			-3.41(7.5)	-4.38(8.7)	-2.53(9.3)
$\log \Lambda_{55}$	Gold standard TA-SMC	RW	-1.39	-1.65	-1.13
		MALA	-1.39(1.0)	-1.65(1.0)	-1.12(1.0)
		SLICE	-1.39(0.6)	-1.65(0.5)	-1.12(0.6)
	NS-SMC	RW	-1.40(39.1)	-1.66(23.8)	-1.13(21.2)
		MALA	-1.38(1.4)	-1.64(1.3)	-1.12(1.8)
		SLICE	-1.39(8.9)	-1.65(5.9)	-1.13(7.1)
			-1.39(16.6)	-1.65(11.2)	-1.13(13.4)
$\log \Lambda_{66}$	Gold standard TA-SMC	RW	-1.37	-1.63	-1.10
		MALA	-1.37(1.0)	-1.63(1.0)	-1.10(1.0)
		SLICE	-1.37(0.4)	-1.63(0.3)	-1.11(0.6)
	NS-SMC	RW	-1.37(28.0)	-1.64(18.6)	-1.11(18.3)
		MALA	-1.37(1.0)	-1.63(1.2)	-1.11(1.3)
		SLICE	-1.37(7.1)	-1.63(4.6)	-1.10(7.0)
			-1.37(15.2)	-1.63(10.6)	-1.11(14.4)

Table 7: Results for FA2 — Part 2 of 3

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$
$\log \beta_{11}$	Gold standard		-0.02	-0.15	0.11
	TA-SMC	RW	-0.02(1.0)	-0.15(1.0)	0.11(1.0)
		MALA	-0.02(0.3)	-0.15(0.3)	0.11(0.5)
		SLICE	-0.02(29.2)	-0.14(16.4)	0.11(32.8)
	NS-SMC	RW	-0.02(0.7)	-0.15(0.8)	0.11(1.6)
		MALA	-0.02(6.0)	-0.15(7.3)	0.11(7.8)
SLICE		-0.02(14.0)	-0.14(6.8)	0.11(22.1)	
β_{21}	Gold standard		0.95	0.83	1.09
	TA-SMC	RW	0.95(1.0)	0.83(1.0)	1.09(1.0)
		MALA	0.95(0.4)	0.83(0.4)	1.09(0.4)
		SLICE	0.96(25.3)	0.84(27.1)	1.09(24.3)
	NS-SMC	RW	0.95(0.8)	0.83(0.9)	1.09(1.2)
		MALA	0.95(4.9)	0.83(6.1)	1.09(6.3)
SLICE		0.95(15.6)	0.83(12.5)	1.09(13.5)	
β_{31}	Gold standard		0.45	0.30	0.62
	TA-SMC	RW	0.45(1.0)	0.30(1.0)	0.62(1.0)
		MALA	0.45(0.4)	0.30(0.4)	0.62(0.4)
		SLICE	0.46(28.7)	0.30(26.3)	0.62(24.6)
	NS-SMC	RW	0.46(0.8)	0.30(1.0)	0.62(1.2)
		MALA	0.45(8.3)	0.30(6.5)	0.62(7.2)
SLICE		0.46(20.9)	0.30(14.1)	0.62(18.8)	
β_{41}	Gold standard		0.39	0.23	0.56
	TA-SMC	RW	0.39(1.0)	0.23(1.0)	0.56(1.0)
		MALA	0.39(0.4)	0.23(0.4)	0.56(0.3)
		SLICE	0.40(22.9)	0.23(23.0)	0.56(22.4)
	NS-SMC	RW	0.40(0.7)	0.23(1.0)	0.56(0.9)
		MALA	0.39(5.7)	0.23(6.1)	0.56(4.9)
SLICE		0.39(18.6)	0.24(15.3)	0.56(15.3)	
β_{51}	Gold standard		0.41	0.25	0.58
	TA-SMC	RW	0.41(1.0)	0.25(1.0)	0.58(1.0)
		MALA	0.41(0.4)	0.25(0.5)	0.58(0.3)
		SLICE	0.42(22.6)	0.25(22.3)	0.58(17.6)
	NS-SMC	RW	0.41(0.8)	0.25(1.1)	0.58(0.8)
		MALA	0.41(5.3)	0.25(5.3)	0.58(3.2)
SLICE		0.41(18.6)	0.26(14.4)	0.58(13.0)	
β_{61}	Gold standard		0.41	0.25	0.57
	TA-SMC	RW	0.41(1.0)	0.25(1.0)	0.58(1.0)
		MALA	0.41(0.4)	0.25(0.3)	0.57(0.4)
		SLICE	0.41(25.6)	0.25(22.4)	0.57(27.1)
	NS-SMC	RW	0.41(0.6)	0.25(1.0)	0.58(1.0)
		MALA	0.41(5.9)	0.25(6.2)	0.57(5.3)
SLICE		0.41(19.3)	0.25(14.9)	0.57(19.0)	

Table 8: Results for FA2 — Part 3 of 3

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$	
$\log \beta_{22}$	Gold standard		-3.54	-6.34	-2.21	
		TA-SMC	RW	-3.57(1.0)	-6.06(1.0)	-2.23(1.0)
			MALA	-3.53(0.3)	-6.17(0.6)	-2.22(0.3)
	SLICE		-3.46(6.0)	-5.15(2.7)	-2.26(11.4)	
	NS-SMC	RW	-3.57(0.4)	-6.34(1.0)	-2.22(0.4)	
		MALA	-3.50(4.6)	-5.49(2.5)	-2.26(11.3)	
		SLICE	-3.49(3.8)	-5.39(1.9)	-2.25(7.5)	
	β_{32}	Gold standard		0.25	-0.50	0.56
			TA-SMC	RW	0.22(1.0)	-0.47(1.0)
MALA				0.25(0.2)	-0.49(0.0)	0.56(0.2)
SLICE		0.16(3.1)		-0.37(6.5)	0.54(75.5)	
NS-SMC		RW	0.22(0.5)	-0.49(0.4)	0.56(0.5)	
		MALA	0.23(2.2)	-0.42(5.6)	0.56(4.1)	
		SLICE	0.22(1.5)	-0.41(4.6)	0.56(1.7)	
β_{42}		Gold standard		0.55	-0.97	1.03
			TA-SMC	RW	0.47(1.0)	-0.93(1.0)
	MALA			0.54(0.2)	-0.96(0.0)	1.03(0.2)
	SLICE	0.35(3.0)		-0.68(7.1)	0.99(658.8)	
	NS-SMC	RW	0.47(0.5)	-0.95(0.5)	1.03(0.3)	
		MALA	0.49(2.2)	-0.80(6.1)	1.03(4.3)	
		SLICE	0.46(1.5)	-0.77(5.1)	1.03(2.4)	
	β_{52}	Gold standard		0.46	-0.84	0.90
			TA-SMC	RW	0.40(1.0)	-0.80(1.0)
MALA				0.46(0.2)	-0.83(0.0)	0.90(0.2)
SLICE		0.29(3.1)		-0.60(7.0)	0.86(482.9)	
NS-SMC		RW	0.40(0.5)	-0.82(0.5)	0.90(0.7)	
		MALA	0.42(2.2)	-0.69(6.0)	0.90(4.9)	
		SLICE	0.39(1.5)	-0.67(5.0)	0.90(2.4)	
β_{62}		Gold standard		0.46	-0.84	0.90
			TA-SMC	RW	0.40(1.0)	-0.80(1.0)
	MALA			0.46(0.2)	-0.83(0.0)	0.90(0.2)
	SLICE	0.29(3.1)		-0.60(7.0)	0.86(405.4)	
	NS-SMC	RW	0.39(0.5)	-0.82(0.5)	0.89(0.5)	
		MALA	0.41(2.2)	-0.69(6.0)	0.89(3.5)	
		SLICE	0.39(1.5)	-0.67(5.0)	0.90(2.0)	

Table 9: FA3 Inference Results — Part 1 of 4

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$
$\log \Lambda_{11}$	Gold standard	RW	-2.89	-4.31	-1.86
		TA-SMC	-2.89(1.0)	-4.25(1.0)	-1.86(1.0)
		MALA	-2.88(0.1)	-4.28(0.1)	-1.86(0.1)
	NS-SMC	SLICE	-2.84(5.0)	-4.14(3.4)	-1.89(11.0)
		RW	-2.88(0.4)	-4.30(0.5)	-1.86(0.6)
		MALA	-2.87(0.9)	-4.28(0.8)	-1.86(1.1)
SLICE	-2.87(2.6)	-4.23(2.0)	-1.86(4.0)		
$\log \Lambda_{22}$	Gold standard	RW	-2.38	-3.88	-1.73
		TA-SMC	-2.38(1.0)	-3.82(1.0)	-1.73(1.0)
		MALA	-2.38(0.1)	-3.84(0.1)	-1.73(0.1)
	NS-SMC	SLICE	-2.43(6.8)	-3.71(4.7)	-1.76(10.7)
		RW	-2.39(0.5)	-3.85(0.8)	-1.73(0.4)
		MALA	-2.41(1.1)	-3.84(1.1)	-1.74(1.0)
SLICE	-2.40(3.2)	-3.84(3.4)	-1.74(3.9)		
$\log \Lambda_{33}$	Gold standard	RW	-1.08	-3.84	-0.27
		TA-SMC	-1.15(1.0)	-3.63(1.0)	-0.28(1.0)
		MALA	-1.12(0.1)	-3.81(0.1)	-0.27(0.1)
	NS-SMC	SLICE	-1.09(2.3)	-3.21(3.4)	-0.28(4.9)
		RW	-1.10(0.4)	-3.75(0.4)	-0.27(0.3)
		MALA	-1.11(0.6)	-3.63(1.1)	-0.28(1.1)
SLICE	-1.11(1.0)	-3.68(1.2)	-0.27(1.0)		
$\log \Lambda_{44}$	Gold standard	RW	-3.17	-4.42	-1.96
		TA-SMC	-3.19(1.0)	-4.41(1.0)	-2.01(1.0)
		MALA	-3.18(0.1)	-4.41(0.1)	-1.99(0.1)
	NS-SMC	SLICE	-3.13(4.9)	-4.28(6.6)	-2.05(4.2)
		RW	-3.18(0.5)	-4.42(0.9)	-1.99(0.4)
		MALA	-3.20(1.0)	-4.43(0.8)	-2.05(2.1)
SLICE	-3.16(2.1)	-4.37(4.1)	-2.00(1.8)		
$\log \Lambda_{55}$	Gold standard	RW	-1.77	-3.80	-1.16
		TA-SMC	-1.77(1.0)	-3.68(1.0)	-1.16(1.0)
		MALA	-1.76(0.1)	-3.74(0.1)	-1.16(0.1)
	NS-SMC	SLICE	-1.80(4.6)	-3.40(3.7)	-1.17(6.0)
		RW	-1.75(0.4)	-3.69(0.5)	-1.16(0.3)
		MALA	-1.73(1.6)	-3.35(2.7)	-1.16(2.3)
SLICE	-1.77(2.1)	-3.62(2.1)	-1.16(2.1)		
$\log \Lambda_{66}$	Gold standard	RW	-1.75	-3.73	-1.14
		TA-SMC	-1.74(1.0)	-3.56(1.0)	-1.14(1.0)
		MALA	-1.73(0.1)	-3.64(0.1)	-1.15(0.1)
	NS-SMC	SLICE	-1.75(3.9)	-3.28(3.3)	-1.15(5.5)
		RW	-1.74(0.5)	-3.64(0.5)	-1.14(0.5)
		MALA	-1.74(1.7)	-3.47(2.5)	-1.14(1.6)
SLICE	-1.74(1.8)	-3.45(2.0)	-1.15(1.7)		

Table 10: Results for FA3 — Part 2 of 4

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$	
$\log \beta_{11}$	Gold standard		-0.02	-0.16	0.11	
		TA-SMC	RW	-0.02(1.0)	-0.16(1.0)	0.11(1.0)
			MALA	-0.02(0.1)	-0.16(0.1)	0.11(0.2)
	SLICE		-0.03(10.7)	-0.16(7.5)	0.10(11.9)	
	NS-SMC	RW	-0.02(1.0)	-0.16(1.3)	0.11(1.4)	
		MALA	-0.02(1.3)	-0.16(1.4)	0.11(1.3)	
		SLICE	-0.03(7.3)	-0.16(6.7)	0.10(8.2)	
	β_{21}	Gold standard		0.96	0.83	1.10
			TA-SMC	RW	0.96(1.0)	0.83(1.0)
MALA				0.96(0.1)	0.83(0.2)	1.09(0.2)
SLICE		0.96(10.2)		0.84(9.2)	1.09(10.5)	
NS-SMC		RW	0.96(1.0)	0.83(1.4)	1.10(1.6)	
		MALA	0.96(1.0)	0.83(1.4)	1.10(1.2)	
		SLICE	0.96(5.4)	0.83(6.6)	1.09(9.6)	
β_{31}		Gold standard		0.46	0.30	0.63
			TA-SMC	RW	0.46(1.0)	0.30(1.0)
	MALA			0.46(0.1)	0.30(0.2)	0.63(0.2)
	SLICE	0.46(15.0)		0.30(11.0)	0.62(10.3)	
	NS-SMC	RW	0.46(1.6)	0.30(2.1)	0.63(1.9)	
		MALA	0.46(1.3)	0.30(1.5)	0.63(1.2)	
		SLICE	0.46(8.3)	0.30(8.4)	0.62(7.9)	
	β_{41}	Gold standard		0.40	0.23	0.57
			TA-SMC	RW	0.40(1.0)	0.23(1.0)
MALA				0.40(0.1)	0.23(0.2)	0.57(0.2)
SLICE		0.40(16.9)		0.23(17.7)	0.56(12.6)	
NS-SMC		RW	0.40(1.1)	0.23(1.9)	0.57(1.3)	
		MALA	0.40(1.0)	0.23(1.5)	0.57(1.4)	
		SLICE	0.40(8.0)	0.23(13.2)	0.56(7.3)	
β_{51}		Gold standard		0.42	0.25	0.58
			TA-SMC	RW	0.42(1.0)	0.25(1.0)
	MALA			0.41(0.1)	0.25(0.2)	0.58(0.2)
	SLICE	0.41(18.3)		0.25(16.2)	0.58(12.3)	
	NS-SMC	RW	0.41(1.5)	0.25(1.9)	0.58(1.5)	
		MALA	0.42(1.1)	0.25(1.2)	0.58(1.7)	
		SLICE	0.41(8.3)	0.25(11.8)	0.58(7.7)	
	β_{61}	Gold standard		0.41	0.25	0.58
			TA-SMC	RW	0.41(1.0)	0.25(1.0)
MALA				0.41(0.1)	0.25(0.2)	0.58(0.3)
SLICE		0.41(16.1)		0.25(14.8)	0.58(14.1)	
NS-SMC		RW	0.41(1.1)	0.25(1.4)	0.58(1.8)	
		MALA	0.41(0.8)	0.25(0.9)	0.58(1.2)	
		SLICE	0.41(8.3)	0.25(11.7)	0.58(9.3)	

Table 11: Results for FA3 — Part 3 of 4

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$	
$\log \beta_{22}$	Gold standard		-3.23	-6.06	-1.63	
		TA-SMC	RW	-3.22(1.0)	-5.83(1.0)	-1.67(1.0)
			MALA	-3.23(0.1)	-5.99(0.3)	-1.65(0.1)
	SLICE		-3.20(5.7)	-5.42(2.8)	-1.73(2.5)	
	NS-SMC	RW	-3.24(0.6)	-6.01(0.8)	-1.64(0.6)	
		MALA	-3.10(2.5)	-5.19(2.1)	-1.64(1.2)	
		SLICE	-3.22(1.8)	-5.74(2.1)	-1.68(1.3)	
	β_{32}	Gold standard		-0.06	-0.83	0.58
			TA-SMC	RW	-0.06(1.0)	-0.79(1.0)
MALA				-0.06(0.1)	-0.83(0.1)	0.59(0.2)
SLICE		-0.06(4.0)		-0.69(4.8)	0.55(3.5)	
NS-SMC		RW	-0.06(0.6)	-0.81(0.7)	0.58(0.9)	
		MALA	-0.08(1.2)	-0.76(1.8)	0.56(1.6)	
		SLICE	-0.06(1.9)	-0.77(2.5)	0.56(1.9)	
β_{42}		Gold standard		0.23	-0.87	0.94
			TA-SMC	RW	0.23(1.0)	-0.82(1.0)
	MALA			0.24(0.1)	-0.86(0.1)	0.94(0.1)
	SLICE	0.19(5.4)		-0.72(6.7)	0.88(29.9)	
	NS-SMC	RW	0.23(0.6)	-0.85(0.3)	0.94(0.5)	
		MALA	0.22(1.9)	-0.78(2.2)	0.92(2.6)	
		SLICE	0.22(2.1)	-0.81(2.2)	0.93(4.2)	
	β_{52}	Gold standard		0.23	-0.79	0.89
			TA-SMC	RW	0.23(1.0)	-0.74(1.0)
MALA				0.24(0.1)	-0.78(0.1)	0.88(0.1)
SLICE		0.20(5.7)		-0.65(5.7)	0.82(15.4)	
NS-SMC		RW	0.23(0.7)	-0.77(0.3)	0.88(0.6)	
		MALA	0.22(1.9)	-0.70(1.9)	0.86(2.6)	
		SLICE	0.22(2.2)	-0.74(2.3)	0.87(2.9)	
β_{62}		Gold standard		0.15	-0.81	0.86
			TA-SMC	RW	0.15(1.0)	-0.77(1.0)
	MALA			0.16(0.1)	-0.80(0.1)	0.86(0.1)
	SLICE	0.12(4.9)		-0.70(6.0)	0.79(16.8)	
	NS-SMC	RW	0.15(0.5)	-0.80(0.4)	0.86(0.5)	
		MALA	0.13(1.8)	-0.74(2.2)	0.82(1.9)	
		SLICE	0.14(1.9)	-0.76(1.9)	0.84(3.0)	

Table 12: Results for FA3 — Part 4 of 4.

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$	
$\log \beta_{33}$	Gold standard		-1.21	-3.63	-0.16	
		TA-SMC	RW	-1.16(1.0)	-3.30(1.0)	-0.19(1.0)
			MALA	-1.23(0.2)	-3.71(0.3)	-0.17(0.1)
	SLICE		-1.23(6.7)	-2.98(2.1)	-0.28(4.2)	
	NS-SMC	RW	-1.22(0.7)	-3.69(1.3)	-0.18(0.5)	
		MALA	-1.17(2.8)	-3.00(2.5)	-0.21(1.6)	
		SLICE	-1.23(3.8)	-3.50(3.4)	-0.20(1.6)	
	β_{43}	Gold standard		0.57	-0.54	0.97
			TA-SMC	RW	0.57(1.0)	-0.40(1.0)
MALA				0.56(0.1)	-0.49(0.1)	0.97(0.1)
SLICE		0.55(5.6)		-0.24(2.3)	0.96(13.1)	
NS-SMC		RW	0.56(0.5)	-0.50(0.7)	0.97(0.5)	
		MALA	0.57(1.7)	-0.35(1.2)	0.97(2.0)	
		SLICE	0.57(1.7)	-0.38(1.6)	0.97(2.7)	
β_{53}		Gold standard		0.47	-0.53	0.89
			TA-SMC	RW	0.47(1.0)	-0.45(1.0)
	MALA			0.46(0.1)	-0.52(0.1)	0.89(0.1)
	SLICE	0.44(5.5)		-0.32(3.6)	0.87(6.7)	
	NS-SMC	RW	0.46(0.5)	-0.51(0.6)	0.89(0.6)	
		MALA	0.47(1.8)	-0.40(1.6)	0.88(2.0)	
		SLICE	0.47(1.5)	-0.43(1.7)	0.88(1.9)	
	β_{63}	Gold standard		0.53	-0.44	0.90
			TA-SMC	RW	0.53(1.0)	-0.33(1.0)
MALA				0.51(0.1)	-0.41(0.1)	0.90(0.1)
SLICE		0.51(6.2)		-0.19(2.6)	0.88(5.5)	
NS-SMC		RW	0.51(0.6)	-0.40(0.8)	0.90(0.6)	
		MALA	0.53(1.9)	-0.27(1.4)	0.89(1.3)	
		SLICE	0.53(2.1)	-0.30(1.9)	0.90(2.1)	

Table 13: Results for ODE model

			$\widehat{\text{mean}}$	$\widehat{\text{lower}}$	$\widehat{\text{upper}}$	
$\log k_1$	Gold standard	TA-SMC	RW	-3.40	-4.46	-2.71
			MALA	-3.33(1.0)	-4.06(1.0)	-2.71(1.0)
			SLICE	-3.18(1.3)	-3.67(0.4)	-2.66(4.2)
	NS-SMC		RW	-3.36(8.3)	-4.09(6.8)	-2.71(7.3)
			MALA	-3.34(4.2)	-4.10(3.4)	-2.71(2.5)
			SLICE	-3.36(2.7)	-4.06(1.7)	-2.72(2.0)
		SLICE	-3.33(9.8)	-3.99(4.8)	-2.70(7.6)	
$\log V_1$	Gold standard	TA-SMC	RW	-0.98	-2.02	0.19
			MALA	-1.03(1.0)	-2.02(1.0)	0.04(1.0)
			SLICE	-1.09(3.2)	-2.03(4.2)	-0.09(2.2)
	NS-SMC		RW	-1.00(6.7)	-1.98(12.0)	0.00(6.8)
			MALA	-1.03(1.7)	-2.03(2.9)	0.06(2.4)
			SLICE	-0.96(1.3)	-1.95(3.9)	0.15(1.4)
		SLICE	-1.02(5.1)	-1.99(11.0)	-0.00(6.9)	
$\log K_{m1}$	Gold standard	TA-SMC	RW	-1.01	-4.12	1.04
			MALA	-0.98(1.0)	-3.69(1.0)	0.95(1.0)
			SLICE	-1.00(3.2)	-3.57(2.0)	0.79(2.2)
	NS-SMC		RW	-0.98(6.8)	-3.56(4.7)	0.93(4.3)
			MALA	-0.96(2.2)	-3.62(2.9)	0.95(2.6)
			SLICE	-0.90(2.8)	-3.43(2.0)	0.94(2.2)
		SLICE	-1.07(4.9)	-3.69(3.3)	0.88(4.6)	
$\log K_{m2}$	Gold standard	TA-SMC	RW	-2.86	-4.38	-1.79
			MALA	-2.77(1.0)	-3.92(1.0)	-1.81(1.0)
			SLICE	-2.58(1.9)	-3.46(1.0)	-1.76(3.2)
	NS-SMC		RW	-2.80(7.0)	-3.93(4.8)	-1.82(5.6)
			MALA	-2.79(3.9)	-3.97(3.3)	-1.81(2.3)
			SLICE	-2.81(2.1)	-3.93(1.4)	-1.81(1.6)
		SLICE	-2.76(8.3)	-3.84(4.4)	-1.80(6.6)	
$\log V_2$	Gold standard	TA-SMC	RW	-2.05	-2.64	-1.07
			MALA	-2.11(1.0)	-2.65(1.0)	-1.47(1.0)
			SLICE	-2.24(1.1)	-2.70(4.7)	-1.82(0.3)
	NS-SMC		RW	-2.09(9.0)	-2.64(8.8)	-1.43(6.6)
			MALA	-2.10(4.6)	-2.65(2.7)	-1.42(3.3)
			SLICE	-2.09(3.0)	-2.64(2.2)	-1.46(1.6)
		SLICE	-2.10(10.4)	-2.65(8.0)	-1.52(4.4)	
$\log S(0)$	Gold standard	TA-SMC	RW	-0.23	-1.24	0.61
			MALA	-0.22(1.0)	-1.18(1.0)	0.59(1.0)
			SLICE	-0.24(3.0)	-1.15(2.9)	0.56(2.6)
	NS-SMC		RW	-0.23(7.8)	-1.14(10.0)	0.56(9.5)
			MALA	-0.21(1.6)	-1.17(2.5)	0.61(1.6)
			SLICE	-0.26(1.6)	-1.22(2.3)	0.55(2.8)
		SLICE	-0.24(4.5)	-1.16(7.5)	0.57(7.4)	
$\log D(0)$	Gold standard	TA-SMC	RW	-2.88	-6.05	-1.02
			MALA	-2.85(1.0)	-5.76(1.0)	-1.01(1.0)
			SLICE	-2.85(2.5)	-5.49(1.2)	-1.06(4.4)
	NS-SMC		RW	-2.84(6.0)	-5.53(5.3)	-1.04(14.5)
			MALA	-2.89(2.7)	-5.84(3.3)	-1.02(5.0)
			SLICE	-2.90(1.1)	-5.75(1.0)	-1.02(2.1)
		SLICE	-2.95(6.5)	-5.66(4.3)	-1.06(12.5)	
$\log R(0)$	Gold standard	TA-SMC	RW	0.31	-0.28	0.83
			MALA	0.30(1.0)	-0.24(1.0)	0.80(1.0)
			SLICE	0.30(2.1)	-0.22(1.7)	0.77(2.2)
	NS-SMC		RW	0.31(7.2)	-0.21(5.3)	0.79(8.2)
			MALA	0.31(1.9)	-0.24(2.6)	0.81(2.6)
			SLICE	0.32(2.1)	-0.20(1.9)	0.82(2.4)
		SLICE	0.30(5.0)	-0.23(4.6)	0.80(5.8)	
$\log R_{pp}(0)$	Gold standard	TA-SMC	RW	-4.17	-6.89	-2.90
			MALA	-4.14(1.0)	-6.60(1.0)	-2.91(1.0)
			SLICE	-4.09(1.1)	-6.25(0.9)	-2.93(3.7)
	NS-SMC		RW	-4.14(4.3)	-6.32(2.6)	-2.93(6.6)
			MALA	-4.19(6.9)	-6.50(2.1)	-2.93(2.8)
			SLICE	-4.15(1.6)	-6.40(1.0)	-2.93(2.3)
		SLICE	-4.10(2.8)	-6.27(2.3)	-2.92(3.9)	

C Kernel Parameter Selection

The following plots display the evolution of the automated choice of MCMC kernel parameters h (for RW/MALA) and w (for Slice Sampling) chosen by the automated procedure described in Section ?? . Note here that iteration on the x -axis refers not to MCMC iteration, but instead the time step of the SMC sampler.

Factor Analysis Example

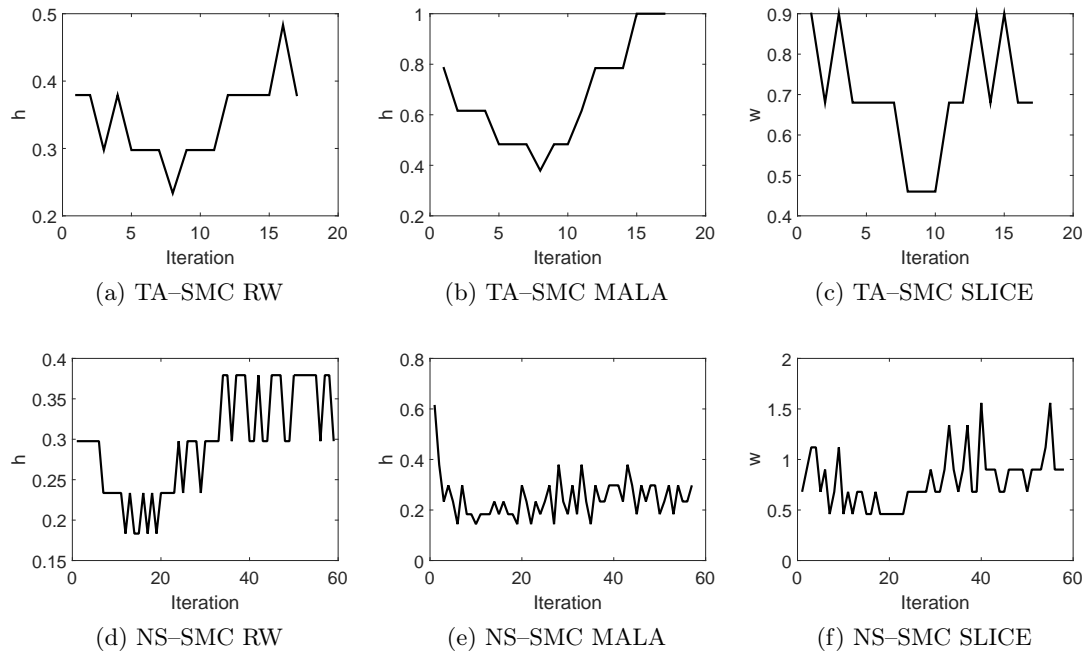


Figure 11: Tuning parameter selection for the one component factor analysis model considered in Section 6.

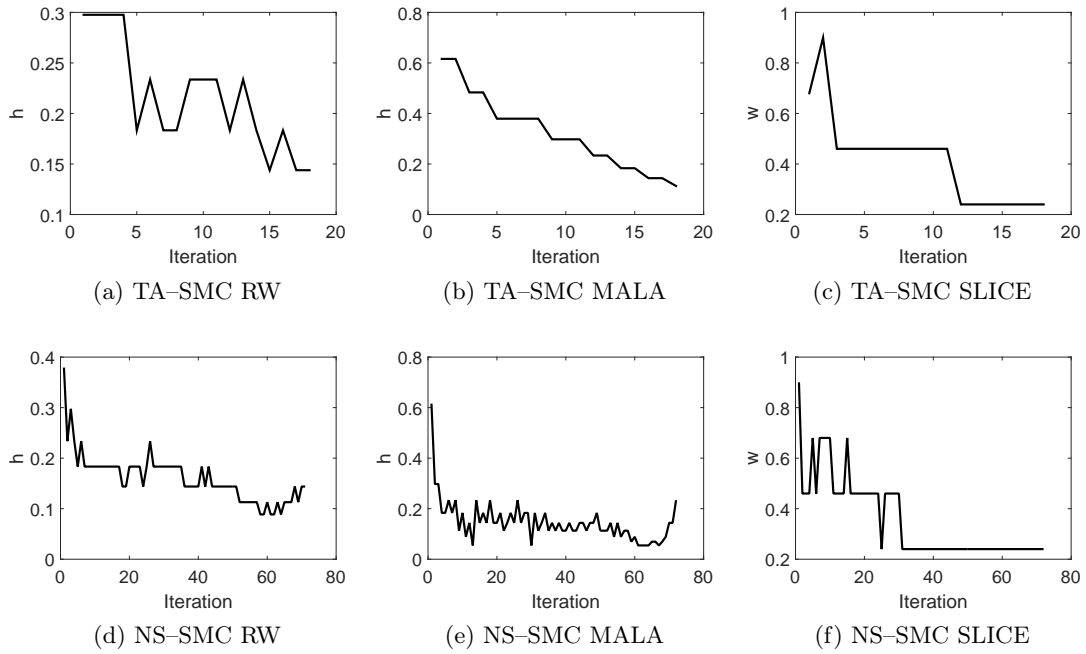


Figure 12: Kernel parameter selection for the two component factor analysis model considered in Section 6.

ODE Example

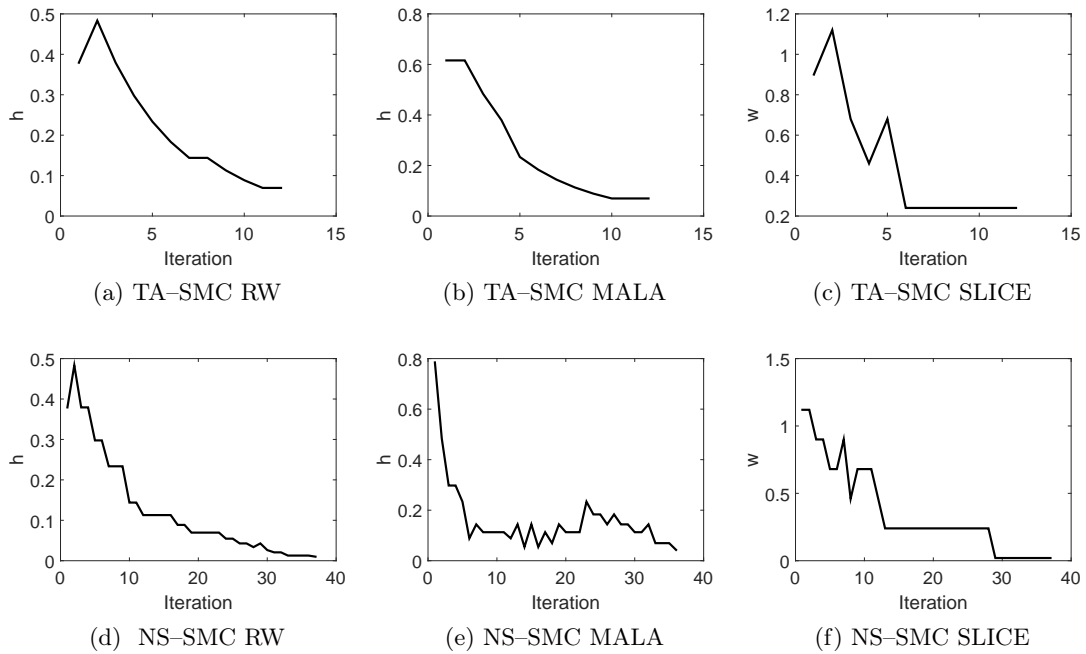


Figure 14: Kernel parameter selection for ODE model considered in Section 6.

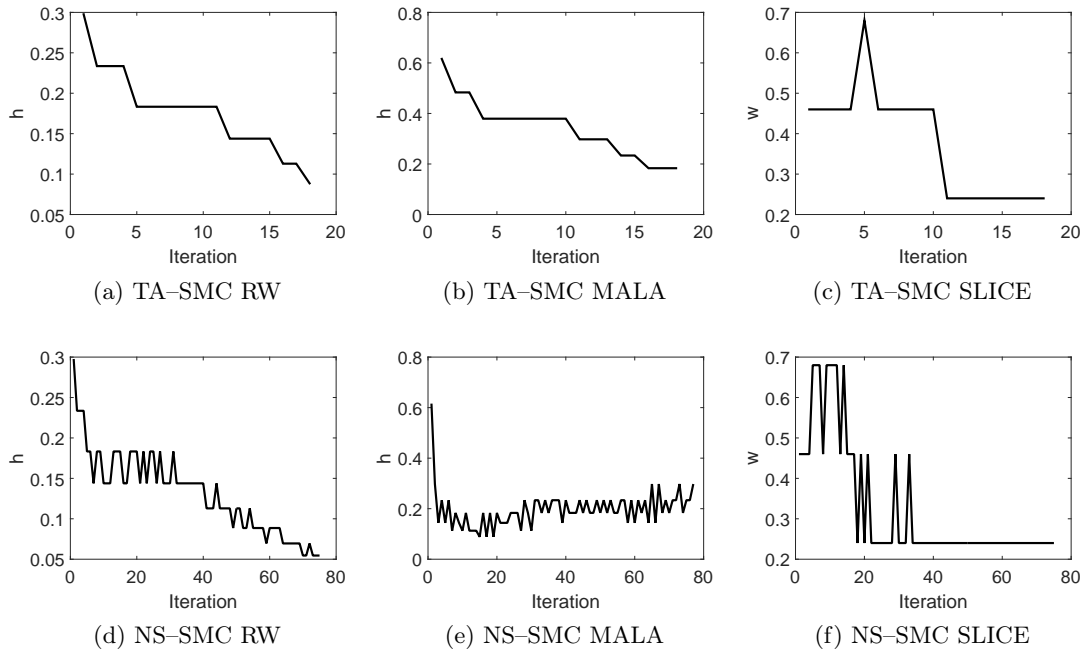


Figure 13: Kernel parameter selection for the three component factor analysis model considered in Section 6.

D MCMC Iterations

The following plots display the evolution of the choice of MCMC iterations (repeats) chosen by the procedure described in Section ???. Note here that iteration on the x -axis refers not to MCMC iteration, but instead the time step of the SMC sampler.

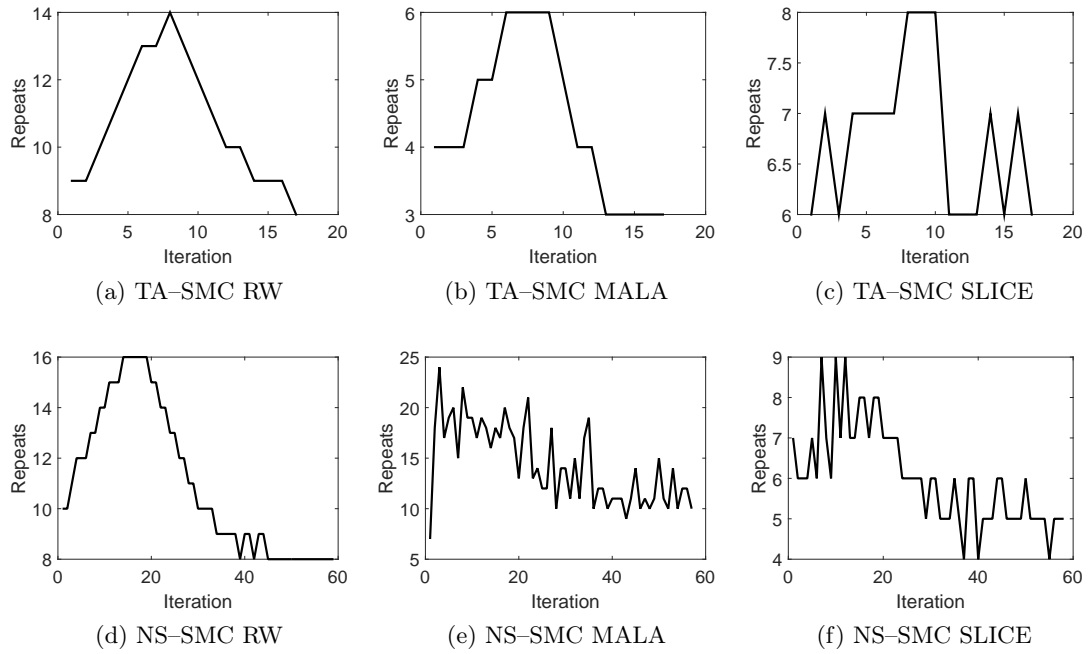


Figure 15: Repeats selection for the one component factor analysis model considered in Section 6.

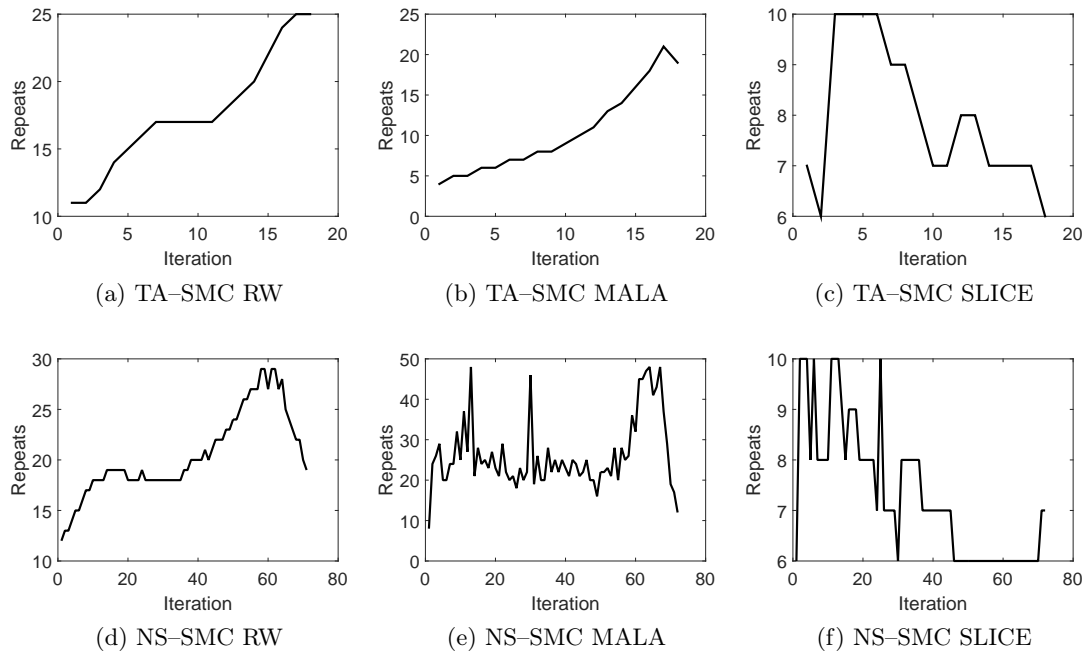


Figure 16: Repeats selection for the two component factor analysis model considered in Section 6.

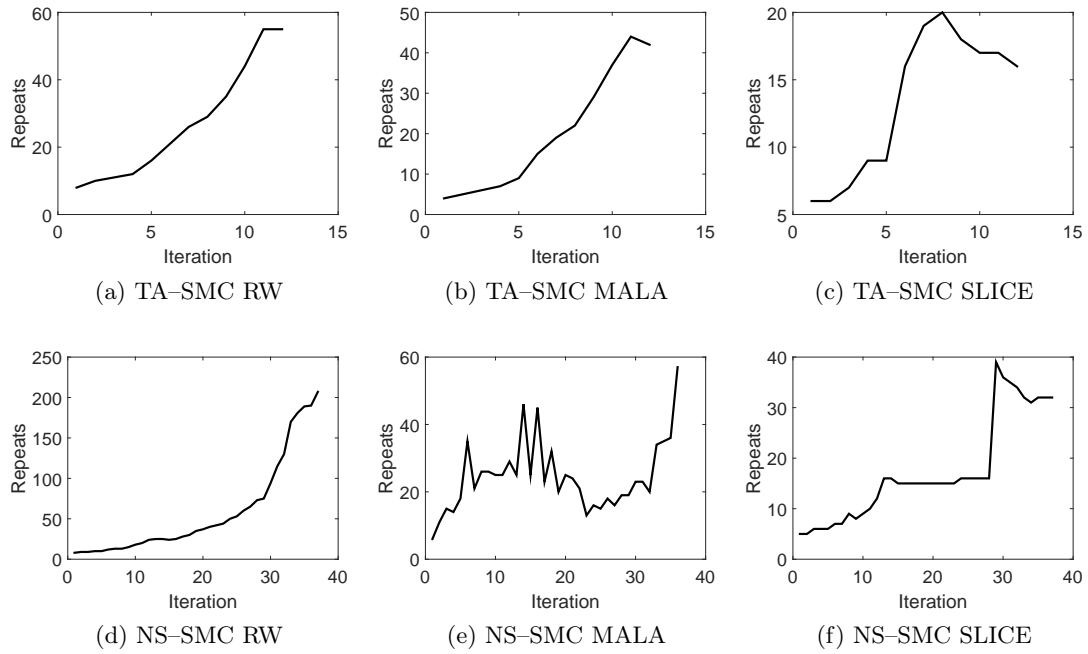


Figure 17: Repeats selection for the three component factor analysis model considered in Section 6.

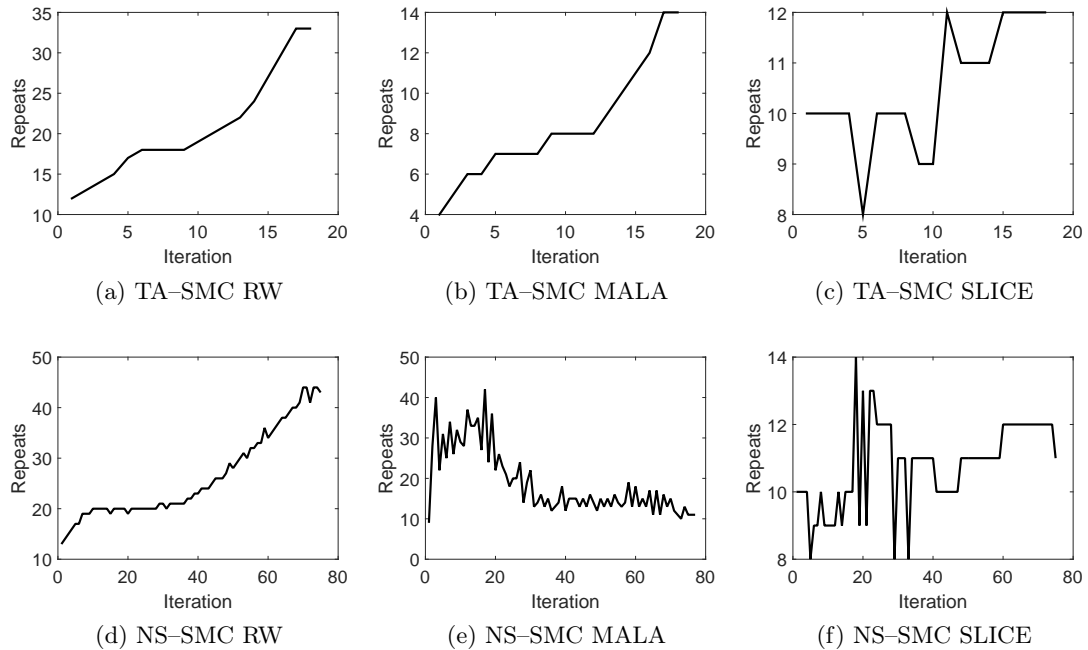


Figure 18: Repeats selection for the ODE model considered in Section 6.