SIAM/ASA J. UNCERTAINTY QUANTIFICATION
Vol. 2, pp. 464–489

Sequential Monte Carlo Methods for High-Dimensional Inverse Problems: A Case Study for the Navier–Stokes Equations*

Nikolas Kantas†, Alexandros Beskos‡, and Ajay Jasra§

Abstract. We consider the inverse problem of estimating the initial condition of a partial differential equation, which is observed only through noisy measurements at discrete time intervals. In particular, we focus on the case where Eulerian measurements are obtained from the time and space evolving vector field, whose evolution obeys the two-dimensional Navier–Stokes equations defined on a torus. This context is particularly relevant to the area of numerical weather forecasting and data assimilation. We will adopt a Bayesian formulation resulting from a particular regularization that ensures the problem is well posed. In the context of Monte Carlo–based inference, it is a challenging task to obtain samples from the resulting high-dimensional posterior on the initial condition. In real data assimilation applications it is common for computational methods to invoke the use of heuristics and Gaussian approximations. As a result, the resulting inferences are biased and not well justified in the presence of nonlinear dynamics and observations. On the other hand, Monte Carlo methods can be used to assimilate data in a *principled* manner, but they are often perceived as inefficient in this context due to the high dimensionality of the problem. In this work we will propose a generic sequential Monte Carlo (SMC) sampling approach for high-dimensional inverse problems that overcomes these difficulties. The method builds upon "state of the art" Markov chain Monte Carlo (MCMC) techniques, which are currently considered as benchmarks for evaluating data assimilation algorithms used in practice. SMC samplers can improve in terms of efficiency, as they possess greater flexibility and one can include steps like sequential tempering, adaptation, and parallelization with a relatively low number of extra computations. We will illustrate this using numerical examples, where our proposed SMC approach can achieve the same accuracy as MCMC but in a much more efficient manner.

Key words. Bayesian inverse problems, sequential Monte Carlo, data assimilation, Navier–Stokes equations

AMS subject classifications. 62, 65, 35R30, 35Q30

DOI. 10.1137/130930364

1. Introduction.

We consider the inverse problem of estimating the initial condition of a dynamical system described by a set of partial differential equations (PDEs) based on noisy observations of its evolution. Such problems are ubiquitous in many application areas, such as meteorology and atmospheric or oceanic sciences, petroleum engineering, and imaging (see, e.g., [1, 14, 34, 33, 8, 20]). In particular, we will look at applications mostly related to numerical weather forecasting and data assimilation, where one is interested in prediction of

*Received by the editors July 24, 2013; accepted for publication (in revised form) June 18, 2014; published electronically September 23, 2014.
http://www.siam.org/journals/juq/2/93036.html

†Department of Mathematics, Imperial College, London, SW7 2AZ, UK (n.kantas@imperial.ac.uk). The work of this author was supported by the EPSRC under grant EP/J01365X/1.

‡Department of Statistical Science, University College, London, WC1E 6BT, UK (a.beskos@ucl.ac.uk). The work of this author was supported by the EPSRC under grant EP/J01365X/1.

§Department of Statistics and Applied Probability, National University of Singapore, Singapore, 117546 (staja@nus.edu.sg). The work of this author was supported by MOE Singapore grant R-155-000-119-133.

the velocity of wind or ocean currents. There, a physical model of the velocity vector field is used together with observed data, in order to estimate its state at some point in the past. This estimated velocity field is then used as an initial condition within the PDE to generate forecasts. In this paper we focus on the case where the model of the evolution of the vector field corresponds to the two-dimensional (2D) Navier–Stokes equations and the data consists of Eulerian observations of the evolving velocity field originating from a regular grid of fixed positions. Although the inverse problem related to the Navier–Stokes dynamics may not be as difficult as some real applications, we believe it can still provide a challenging problem where the potential of our methods can be illustrated. Furthermore, the scope of our work extends beyond this particular model and the computational methods we will present are generic to inverse problems related with dynamical systems.

In a more formal setup, let $(U, \|\cdot\|_U)$ and $(Y, \|\cdot\|_Y)$ be given normed vector spaces. A statistical inverse problem can be formulated as having to find an unknown quantity $u \in U$ that generates data $y \in Y$:

$$y = \mathcal{G}(u) + e\,,$$

where $\mathcal{G} : U \to Y$ is an observation operator and $e \in Y$ denotes a realization of the noise in the observation; see [20] for an overview. In a least squares formulation, one may add a Tikhonov–Phillips regularization term to ensure that the problem is well posed (see, e.g., [8, 23, 33]), in which case one seeks to find the minimizer

$$u^\star = \arg\min_{u\in U} \left(\left\| \Gamma^{-1/2} \left(y - \mathcal{G}(u) \right) \right\|_Y^2 + \left\| \mathcal{C}^{-1/2} (u - \mathfrak{m}) \right\|_U^2 \right) ,$$

where $\Gamma, \mathcal{C}$ are trace class, positive, self-adjoint operators on Y, U, respectively, and $\mathfrak{m} \in U$. In addition, one may also be interested in quantifying the uncertainty related to the estimate $u^\star$. This motivates following a Bayesian inference perspective, which is the one adopted in this work. Under appropriate conditions (to be specified later; see [33]) one can construct a posterior probability measure μ on U such that Bayes' rule holds:

$$\frac{d\mu}{d\mu_0}(u) \propto l(y; u)\,,$$

where μ_0 is the prior and $l(y; u)$ is the likelihood. The prior is chosen to be a Gaussian probability measure $\mu_0 = \mathcal{N}(\mathfrak{m}, \mathcal{C})$ (i.e., a normal distribution on U with mean $\mathfrak{m} \in U$ and covariance operator $\mathcal{C}$), as implied by prior knowledge on the smoothness or regularization considerations. The likelihood, $l(y; u)$, is a density w.r.t some reference measure on Y and is obtained from the statistical model believed to generate the data. For example, one may use

$$l(y; u) = \exp\left(-\frac{1}{2} \left\| \Gamma^{-1/2} \left(y - \mathcal{G}(u) \right) \right\|_Y^2 \right)$$

if a Gaussian additive noise model is adopted.

In this paper we will consider u to be the unknown initial condition of the PDE of interest. We will model the observations as a vector of real random variables, $Y \in \mathbb{R}^{d_y}$, and assume U is an appropriate Hilbert space. Thus, the observation operator is closely related to the semigroup of solution operators of the PDE, $\{\Psi(\cdot, t) : U \to U\}_{t\geq 0}$, which maps a chosen

initial condition $u \in U$ to the present state $\Psi(u,t)$ at time $t \geq 0$. It is straightforward to both extend Bayesian methodology for these spaces [33] and to also ensure that necessary differentiability and smoothness conditions are being enforced w.r.t. to the evolution of the vector field via the appropriate choice of the prior measure. We will also work with periodic boundary domains, which is a convenient choice that allows solving PDEs numerically using a spectral Galerkin method with fast Fourier transforms (FFTs). Notice that here we are confronted with an infinite-dimensional problem, as U is a function space, but in practice a high-dimensional discretization (or mesh) is used. Still, it remains an important requirement that any computational method should be able to cope with an arbitrary fine discretization, i.e., that it is robust to mesh refinement.

A plethora of methods have appeared in the literature to tackle such inverse problems. Usually these adopt various heuristic approximations when new data points are assimilated. The first successful attempt in this direction of algorithms was based on optimization and variational principles [24, 30]. Later, these ideas were combined with Gaussian approximations, linearizations, and Kalman-type computations in [34], leading to the popular 3DVAR and 4DVAR. Another popular method is the ensemble Kalman filter (enKF), which is nowadays employed by an increasing number of weather forecasting centers; see [14] for an overview. Although these methods have been used widely in practice, an important weakness is that their use is not well justified for nonlinear problems and it is hard to quantify under which conditions they are accurate (with the exception of linear Gaussian models; see [25]). A different direction that overcomes this weakness is to use Monte Carlo computations that make full use of Bayes' rule to assimilate data in a principled manner. In this paper we will refer to these methods as "exact," given the resulting estimation error will diminish by using more Monte Carlo samples and also in order to distinguish them from the above methods that use heuristic approximations. Recently, exact Markov chain Monte Carlo (MCMC) methods suitable for high-dimensional inverse problems have been proposed in the literature [8, 22, 33]. This class of MCMC algorithms can be shown to be much more accurate than the popular data assimilation algorithms mentioned earlier (see [23] for a thorough comparison). However, the improvement in performance comes at a much greater computational cost, limiting the effect of the method to providing benchmarks for evaluating data assimilation algorithms used in practice.

In this paper, we aim to improve in terms of the efficiency of obtaining Monte Carlo samples for Bayesian inference. We will use these accurate MCMC methods as building blocks within sequential Monte Carlo (SMC) samplers [6, 12]. Our work builds upon recent advances in MCMC/SMC methodology, and we will propose an SMC sampler suitable for high-dimensional inverse problems. SMC methods have been very successful in a wide range of relatively low-dimensional applications [13], and their validity has been demonstrated by many theoretical results (see [11] for an exhaustive review). SMC can be a useful numerical tool for high-dimensional data assimilation applications, but one needs to implement very efficient versions of these algorithms and not simply use them in their plain generic form. Evidence for this can be provided by the recent success of SMC in high-dimensional applications [19, 31] as well as recent theoretical results with emphasis on high-dimensional problems [2, 3, 32]. The necessity for developing more elaborate implementations of SMC has been also recognized in related high-dimensional filtering problems with stochastic dynamics [36, 7]. Here we will not

consider this type of problem and focus only on the case of deterministic dynamics.

We will propose an efficient algorithm based on the algorithm in [6]. We will generate weighted samples (called particles) from a sequence of target probability measures, $(\mu_n)_{n=0}^T$, that starts from the prior, μ_0, and terminates at the posterior of interest (i.e., $\mu_T = \mu$). This is achieved by a combination of importance sampling, resampling, and MCMC mutation steps. Several important challenges arise when trying to use this approach for the high-dimensional problems of interest in this paper: (i) *Overcoming weight degeneracy:* when the amount of information in an assimilated data point is overwhelming, the importance weights will exhibit a very high variance. For example, at the nth step of the algorithm (when targeting μ_n) the observations about to be assimilated might exhibit a highly peaked likelihood function relative to the previous target and current proposal μ_{n-1}. (ii) *Constructing effective MCMC mutation kernels:* the availability of MCMC kernels with sufficient mixing properties is well known to be critical for algorithmic efficiency of SMC [11]. This is extremely challenging in high dimensions since the target distributions are typically comprised of components with widely varying scales and complex correlation structures. (iii) *Effective design and monitoring of algorithmic performance:* insufficient numbers of particles and MCMC mutation steps or inefficient MCMC kernels might lead to a population of particles without the required diversity to provide good estimates. Even in such an undesirable situation, standard performance indicators such as the effective sample size (ESS) can give satisfactory values and a false sense of security (this has been noted in [6]). Hence the development and use of reliable criteria to monitor performance is required, and these should be easy to compute using the particles.

SMC samplers possess a great amount of flexibility with design elements that can be modified according to the particular problem at hand. Understanding some of the statistical properties of the posterior of interest can be used to design an appropriate (and possibly artificial) target sequence $(\mu_n)_{n=0}^T$ as well as to construct MCMC mutation steps with adequate mixing. To overcome the difficulties mentioned above, in points (i)–(ii) we will revisit some previously successful adaptation ideas from the SMC literature. First, we will employ sequential and adaptive tempering to smooth peaked likelihoods by inserting an intermediate target sequence between μ_{n-1} and μ_n. At each step of the algorithm, the next temperature will be chosen automatically based on information from the particles, as proposed in [19]. In particular for our problem, adaptive tempering will not increase the total computational cost too much, when more tempering is performed at earlier stages of the algorithm, which require shorter runs of the expensive numerical solutions of the PDE. In addition, to address point (ii) above, we will use the particles at each stage of the algorithm and adapt the MCMC steps to the structure of the target as suggested in [6]. In order to achieve this, we will propose a novel MCMC kernel robust to high dimensions using principles similar to those in [8].

Our main contribution will be to combine these two ideas and propose a generic and efficient SMC algorithm that can be used for a variety of inverse problems of interest to the data assimilation community. In addition, we will propose a statistic to measure the amount of diversity of the particles during the MCMC mutation. We will use a particular standardized square distance travelled by the particles during the mutation, which to the best of our knowledge has not been used before. Good values for this criterion might be chosen by requiring a minimum amount of decorrelation. The performance of the proposed scheme will be demonstrated numerically on the inverse problem related to the Navier–Stokes equations,

but we expect similar performance in other problems, such as those described in [8]. For the numerical implementation, we will exploit the fact that many steps in SMC are trivially parallelizable. This leads to high speedups in execution time when implemented on appropriate hardware platforms, such as computing clusters or GPUs [26]. Although this is becoming an increasingly common practice in the SMC community, to the best of our knowledge this is the first attempt to investigate the practical speed-up benefits in high-dimensional inverse problems.

The organization of the paper is as follows. In section 2 we formulate the inverse problem related to the Navier–Stokes equations that will be used in this paper. In section 3 we present the MCMC sampling procedure of [8] and a basic SMC sampling method. In section 4 we will extend the SMC methodology for high-dimensional inverse problems. In section 5 we present two numerical examples with the inverse problem for the Navier–Stokes equations: in the first one, SMC appears to achieve the same accuracy as MCMC at a fraction of the computational cost; in the second one, it is unrealistic to use MCMC from a computational perspective, but SMC can provide satisfactory numerical solutions at a reasonable computational cost. Finally, in section 6 we present some concluding remarks.

2. Problem formulation.

In this section we will give a brief description of the Navier–Stokes equations defined on a torus, specify the observation mechanism, and present the posterior distribution of interest for the initial condition. We will later use the problem formulated in this section as a case study for the proposed SMC algorithm for inverse problems.

2.1. Navier–Stokes equations on a torus.

We will first set up the appropriate state space and then present the dynamics. Readers with less interest in the Navier–Stokes equations should be able to go directly to section 2.2 with little loss in continuity.

2.1.1. Preliminaries.

Consider the state (or phase) space being the 2D torus, $\mathbb{T} = [0, 2\pi) \times [0, 2\pi)$, with $x \in \mathbb{T}$ being a point on the space. The initial condition of interest is a 2D vector field $u : \mathbb{T} \to \mathbb{R}^2$. We set $u = (u_1(x), u_2(x))'$, where $u_1, u_2 \in L^2(\mathbb{T})$ and $\cdot'$ denotes vector/matrix transpose. We will define the vorticity as $\varpi = \varpi(x,t) = -\nabla \times u(t,x)$, with the (slightly unusual) convention that clockwise rotation leads to positive vorticity. Let $|\cdot|$ denote the magnitude of a vector or complex variate. For a scalar field $g : \mathbb{T} \to \mathbb{R}$ we will write $\nabla^\perp g = (-\partial_{x_2} g, \partial_{x_1} g)'$. We will also consider the vector Laplacian operator, $\Delta u = \left(\partial^2_{x_1} u_1 + \partial^2_{x_2} u_1, \partial^2_{x_1} u_2 + \partial^2_{x_2} u_2\right)'$, and, for functions $\widetilde{v}, v : \mathbb{T} \to \mathbb{R}^2$, the operator $(v \cdot \nabla)\,\widetilde{v} = (v_1 \partial_{x_1} \widetilde{v}_1 + v_2 \partial_{x_2} \widetilde{v}_1, v_1 \partial_{x_1} \widetilde{v}_2 + v_2 \partial_{x_2} \widetilde{v}_2)'$. Define the Hilbert space,

$$\mathbb{U} := \left\{ 2\pi\text{-periodic trigonometric polynomials } u : \mathbb{T} \to \mathbb{R}^2 \middle|\ \nabla \cdot u = 0\ ,\ \int_{\mathbb{T}} u(x)dx = 0 \right\},$$

and let U be the closure of $\mathbb{U}$ w.r.t. the norm in $L^2(\mathbb{T})^2$. Let also $P : (L^2(\mathbb{T}))^2 \to U$ denote the Leray–Helmholtz orthogonal projector. An appropriate orthonormal basis for U is comprised of the functions $\psi_k(x) = \frac{k^\perp}{2\pi|k|} \exp(ik \cdot x)$, $k \in \mathbb{Z}^2 \setminus \{0\}$, where $k^\perp = (-k_2, k_1)'$ and $i^2 = -1$. So k corresponds to a (bivariate) frequency and the Fourier series decomposition of an element $u \in U$ is written as

$$u(x) = \sum_{k \in \mathbb{Z}^2/\{0\}} u_k \psi_k(x)\,, \quad u_k = \langle u, \psi_k \rangle = \int_{\mathbb{T}} u \cdot \bar{\psi}_k(x)dx\,,$$

for the Fourier coefficients u_k, with $\bar{\cdot}$ denoting complex conjugate. Notice that since u is real valued, we will have $\overline{u_k} = -u_{-k}$.

Also we define $A = -P\Delta$ to be the Stokes operator; note that A is diagonalized in U in the basis $\{\psi_k\}_{k\in\mathbb{Z}^2\setminus\{0\}}$ with eigenvalues $\{\lambda_k\}_{k\in\mathbb{Z}^2\setminus\{0\}}$, where $\lambda_k = |k|^2$. Fractional powers of the Stokes operator can then be defined by the diagonalization. For any $s \geq 0$, we define A^s as the operator with eigenvalues $\lambda_{k,s} = |k|^{2s}$ and eigenfunctions $\{\psi_k\}_{k\in\mathbb{Z}^2\setminus\{0\}}$ and the Hilbert spaces $U^s \subseteq U$ as the domain of $A^{s/2}$, that is, the set of $u \in U$ such that $\sum_{k\in\mathbb{Z}^2/\{0\}} |k|^{2s}|u_k|^2 < \infty$.

2.1.2. The Navier–Stokes equations. The Navier–Stokes equations describe Newton's laws of motion for an incompressible flow of fluid defined on $\mathbb{T}$. Let the flow be initialized with $u \in U$, and consider the case where the mean flow is zero. We will denote the time and space varying velocity field as $v : \mathbb{T} \times [0,\infty) \to \mathbb{R}^2$, $v(x,t) = (v_1(x,t), v_2(x,t))'$, and this is given as follows:

$$\partial_t v - \nu\Delta v + (v\cdot\nabla)\, v = f - \nabla \mathfrak{p}\,, \quad v(x,0) = u(x)\,,$$
$$\nabla\cdot v = 0\,, \quad \int_{\mathbb{T}} v_{\mathtt{j}}(x,\cdot)dx = 0\,,\ \mathtt{j} = 1,2\,,$$

where $\nu > 0$ is the viscosity parameter, $\mathfrak{p} : \mathbb{T}\times[0,\infty) \to \mathbb{R}$ is the pressure function, and $f : \mathbb{T} \to \mathbb{R}^2$ is an exogenous time-homogeneous forcing. We assume periodic boundary conditions:

$$v_{\mathtt{j}}(\cdot,0,t) = v_{\mathtt{j}}(\cdot,2\pi,t)\,, \quad v_{\mathtt{j}}(0,\cdot,t) = v_{\mathtt{j}}(2\pi,\cdot,t)\,, \quad \mathtt{j} = 1,2\,.$$

Applying the projection P to v, we may write the equations in the form of an ordinary differential equation (ODE) in U:

$$\frac{dv}{dt} + \nu Av + B(v,v) = P(f)\,,\ v(0) = u\,, \tag{2.1}$$

where the symmetric bilinear form is defined as

$$B(v,\tilde{v}) = \tfrac{1}{2}P((v\cdot\nabla)\,\tilde{v}) + \tfrac{1}{2}P((\tilde{v}\cdot\nabla)\,v)\,.$$

Intuitively, P projects an arbitrary forcing f into the space of incompressible functions U. See [29, 15] for more details. Let $\{\Psi(\cdot,t) : U \to U\}_{t\geq 0}$ denote the semigroup of solution operators for (2.1) through t time units. We also define the discrete-time semigroup $G_\delta^{(n)}(\cdot) = \Psi(\cdot, n\delta)$ corresponding to time instances $t = n\delta$, of lag $\delta > 0$ and $n = 0,\ldots,T$, with the conventions $G_\delta^{(0)} = I$, $G_\delta^{(1)} = G_\delta$, and $G_\delta^{(n)} = G_\delta \circ G_\delta^{(n-1)}$.

In practice we will use a finite but high-dimensional approximation for $G_\delta^{(n)}(u)$, which is obtained numerically using a mesh for u, v; we will present the details of the numerical solution of (2.1) in section 5.

2.2. A Bayesian framework for the initial condition. We will model the data as noisy measurements of the evolving velocity field v on a fixed grid of points, $x_1,\ldots,x_\Upsilon$, for $\Upsilon \geq 1$. These are obtained at regular time intervals that are δ time units apart. So the observations will be as follows:

$$y_{n,\varsigma} = v(x_\varsigma, n\delta) + \gamma\zeta_{n,\varsigma}\,, \quad \zeta_{n,\varsigma} \overset{iid}{\sim} \mathcal{N}(0,1)\,, \quad 1 \leq \varsigma \leq \Upsilon,\ 1 \leq n \leq T\,,$$

where $\gamma \geq 0$ is constant and v is initialized by the unknown "true" initial vector field, $u^\dagger$. To simplify the expressions, we will write $y = \left((y_{n,\varsigma})_{\varsigma=1}^{\Upsilon}\right)_{n=1}^{T}$. Performing inference with this type of data is referred to as Eulerian data assimilation. The likelihood of the data, conditionally on the unknown initial condition u, can be written as

$$l(y;u) = \frac{1}{\mathcal{Z}(y)} \prod_{n=1}^{T} \prod_{\varsigma=1}^{\Upsilon} \exp\left(-\tfrac{1}{2\gamma^2} \left(y_{n,\varsigma} - G_\delta^{(n)}(u)(x_\varsigma) \right)^2 \right), \tag{2.2}$$

where $\mathcal{Z}(y)$ is a normalizing constant that does not depend on u.

We will also consider the following family of priors:

$$\mu_0 = \mathcal{N}(0, \beta^2 A^{-\alpha}), \tag{2.3}$$

with hyperparameters $\alpha > 1$, $\beta > 0$ affecting the roughness and magnitude of the initial vector field. This is a convenient but still a flexible enough choice of a prior; see [10, sections 2.3 and 4.1] for an introduction to Gaussian distributions on Hilbert spaces. Indeed, when considering the Fourier domain, we have the real function constraint for the complex conjugate coefficients ($u_k = -\overline{u_{-k}}$), so we split the domain by defining

$$\mathbb{Z}^2_\uparrow = \left\{k = (k_1, k_2) \in \mathbb{Z}^2 \setminus \{0\} : \; k_1 + k_2 > 0\right\} \cup \left\{k = (k_1, k_2) \in \mathbb{Z}^2 \setminus \{0\} : \; k_1 + k_2 = 0, \, k_1 > 0\right\}.$$

We will impose that $u_k = -\overline{u_{-k}}$ for $k \in \{\mathbb{Z}^2 \setminus \{0\}\} \setminus \mathbb{Z}^2_\uparrow$. Since the covariance operator is determined via the Stokes operator A, we have the following equivalence when sampling from the prior:

$$u \sim \mu_0 \;\Leftrightarrow\; \mathrm{Re}(u_k), \, \mathrm{Im}(u_k) \overset{iid}{\sim} \mathcal{N}(0, \tfrac{1}{2}\beta^2 |k|^{-2\alpha}), \; k \in \mathbb{Z}^2_\uparrow.$$

That is, μ_0 admits the the following Karhunen–Loève expansion:

$$\mu_0 = \mathcal{L}aw \left(\sum_{k \in \mathbb{Z}^2 \setminus \{0\}} \tfrac{\beta}{\sqrt{2}} |k|^{-\alpha} \, \xi_k \, \psi_k \right); \tag{2.4}$$

$$\mathrm{Re}(\xi_k), \, \mathrm{Im}(\xi_k) \overset{iid}{\sim} \mathcal{N}(0,1), \; k \in \mathbb{Z}^2_\uparrow; \quad \xi_k = -\overline{\xi_{-k}}, \; k \in \{\mathbb{Z}^2 \setminus \{0\}\} \setminus \mathbb{Z}^2_\uparrow. \tag{2.5}$$

Thus a priori, the Fourier coefficients u_k with $k \in \mathbb{Z}^2_\uparrow$ are assumed independent and normally distributed, with a particular rate of decay for their variances as $|k|$ increases.

Adopting a Bayesian inference perspective, we need to construct a posterior probability measure μ on U:

$$\frac{d\mu}{d\mu_0}(u) = \frac{1}{Z(y)} \, l(y;u), \tag{2.6}$$

where $Z(y)$ is the normalization constant. Due to the generality of the state space, some care is needed here to make sure that under the chosen prior, the mappings $G_\delta^{(n)}$ possess enough regularity (i.e., they are μ_0-measurable) and hence the change of measure is well defined. We refer the interested reader to a longer preprint of this paper, [21], for more details.

Algorithm 1 MCMC for High-Dimensional Inverse Problems (see [8, 33] for more details).

- Run a μ-invariant Markov chain $(u(m); m \geq 0)$ as follows:
- Initialize $u(0) \sim \mu_0$. For $m \geq 1$:
  1. Propose:
  $$\widetilde{u} = \rho\, u(m-1) + \sqrt{1-\rho^2}\, \mathfrak{Z}, \quad \mathfrak{Z} \sim \mu_0.$$
  2. Accept $u(m) = \widetilde{u}$ with probability:
  $$1 \wedge \frac{l(y;\widetilde{u})}{l(y;u(m-1))}; \tag{3.1}$$
  otherwise $u(m) = u(m-1)$.

3. Monte Carlo methods for the inverse problem.

In this section we present some Monte Carlo algorithms that can be used for inverse problems such as the one involving the Navier–Stokes dynamics formulated in section 2. We will present first a well-established MCMC method applied in this context and then outline a basic general-purpose SMC sampling algorithm. We postpone the presentation of our proposed method until section 4.

3.1. An MCMC method on the Hilbert space.

MCMC is an iterative procedure for sampling from μ, where one simulates a long run of an ergodic time-homogeneous Markov chain $(u(m); m \geq 0)$ that is μ-invariant. After a few iterations (burn-in) the samples of this chain can be treated as approximate samples from μ. There are many possible transition kernels for implementing MCMC chains, but we will focus only on some algorithms that have been carefully designed for the posteriors of interest in this paper and seem to be particularly appropriate for Hilbert-space-valued measures arising as a change of measure from Gaussian laws. In Algorithm 1 we present such an algorithm, which has appeared earlier in [27] as a regression tool for Gaussian processes and in [8, 33] in the context of high-dimensional inference. ρ is the only tuning parameter taking values in $(0, 1)$ and controls the step size of the proposed move. Note that here and throughout we will use the convention $u(m)$ to denote the mth iteration of any MCMC transition kernel and the notation $\min\{a, b\} = a \wedge b$.

There are some particular challenges in this application that make other more standard MCMC algorithms (e.g., Gibbs, random-walk Metropolis) difficult to apply:

(i) The posterior lacks a hierarchical modeling structure, so there are no conditional independencies of the Fourier coefficients present to exploit. Therefore, implementation of conditional updates of updates of a fraction (or block) of Fourier coefficients would require calculations over all coefficients (or dimensions), making Gibbs-type schemes not useful in practice.

(ii) The posterior is defined on an infinite-dimensional state space. In practice a high-dimensional approximation (mesh) will be used, but we still require the method to be valid for an arbitrary mesh size and hence robust to mesh refinement.

(iii) The information in the observations is not spread uniformly over the Fourier coefficients. A posteriori these can have very different scaling, ranging from very low frequencies to very high ones. At low frequencies one expects that the support of the posterior can change

drastically from the prior. All these different scales cannot be easily determined either analytically or approximately, making it difficult for MCMC algorithms to adjust their proposal's step sizes in the many different directions of the state space.

Considerations (i) and (ii) have prompted the development of Algorithm 1 and various extensions, forming a family of global-update MCMC algorithms, which are well defined on the Hilbert space and robust upon mesh refinement. In direct relevance to the purposes of this paper, Algorithm 1 has been applied in the context of data assimilation and is often used as the "gold standard" benchmark to compare data assimilation algorithms, as done in [23]. One interpretation as to why the method works in infinite dimensions is that step 1 of Algorithm 1 provides a proposal transition kernel that preserves the Gaussian prior μ_0, while the posterior itself will be preserved using the accept/reject rule in step 2. In contrast, standard random-walk Metropolis proposals (of the type $\widetilde{u} = u(m-1) + \text{noise}$) would provide proposals of a distribution which is singular w.r.t. the target μ and would thus be assigned zero acceptance probability. In practice, when a finite-dimensional approximation of u is used, both the standard MCMC methods and Algorithm 1 will have nonzero acceptance probability, but in the limit, only Algorithm 1 is valid. The mixing properties of the standard MCMC transition kernels will also diminish quickly to zero upon mesh refinement (in addition to the acceptance probability), whereas this is not true for Algorithm 1; see [17] for more details.

As a limitation, it has been noted often in practice that a value of ρ very close to 1 is needed (e.g., 0.9998 will be used later on) to achieve a reasonable average acceptance probability (say 0.2–0.3). This is because the algorithm is optimally tuned to the prior Gaussian measure μ_0, whereas the posterior resembles closely μ_0 only at the Fourier coefficients of very high frequencies. This leads to small exploration steps in the proposal and relatively slow mixing of the MCMC chain, which means that one needs to run the chain for an excessive number of iterations (of the order 10^6) to get a set of samples with reasonable quality. In addition, each iteration requires running a PDE solver until time T to compute $l(y;u)$ in step 2, so the approach is very computationally expensive. To sum up, although Algorithm 1 has provided satisfying results in many applications [8], there is still a great need for improving the efficiency of the method. One possible approach in this direction is described in [22].

3.2. A generic SMC approach. We proceed by a short presentation of SMC and refer the reader to [6, 12] for a more thorough treatment. Consider a sequence of probability measures $(\mu_n)_{n=0}^T$ defined on U such that $\mu_T = \mu$ and μ_0 is a prior as in (2.3). A popular choice is to define each target μ_n in the sequence as follows:

$$\frac{d\mu_n}{d\mu_0}(u) = \frac{1}{Z_n}\prod_{p=1}^{n} l_p(y_p;u)\,, \quad 0 \le n \le T\,, \tag{3.2}$$

where each l_p is the likelihood of the block of observations at the pth epoch:

$$l_p(y_p;u) := \frac{1}{\mathcal{Z}_p(y_p)}\prod_{\varsigma=1}^{\Upsilon}\exp\left(-\tfrac{1}{2\gamma^2}\left(y_{p,\varsigma} - G_\delta^{(p)}(u)(x_\varsigma)\right)^2\right).$$

Note that as p increases, so does the computational effort required to compute l_p due to using a numerical PDE solution to evaluate $G_\delta^{(p)}(u)$. This forms a bridging sequence of distributions

between the prior and the posterior, which also admits a Karhunen–Loève expansion:

$$\mu_n = \mathcal{L}aw\left(\sum_{k\in\mathbb{Z}^2\setminus\{0\}} \frac{\beta}{\sqrt{2}}\,|k|^{-\alpha}\,\xi_{k,n}\,\psi_k\right),\ k\in\mathbb{Z}^2_{\uparrow};\quad \xi_{k,n} = -\bar{\xi}_{-k,n}\,,\ k\in\{\mathbb{Z}^2\setminus\{0\}\}\setminus\mathbb{Z}^2_{\uparrow}\,, \tag{3.3}$$

where compared to (2.4)–(2.5), $\{\xi_{k,n}\}_{k\in\mathbb{Z}^2_{\uparrow}}$ are now correlated random variables from some unknown distribution. Note that the particular choice of $(\mu_n)_{n=0}^T$ in (3.2) is a natural choice for this problem. In fact, there are other alternatives involving artificial sequences and introduction of auxiliary variables; see [12] for some more examples. The SMC algorithm will target sequentially each intermediate μ_n, which will be approximated by a weighted collection of $N \gg 1$ samples often referred to as particles. This is achieved by a sequence of selection and mutation steps (see [11, Chapter 5]).

Selection step. Assume at the nth iteration we have N equally weighted samples of μ_{n-1}, denoted by$\{u_{n-1}^j\}_{j=1}^N$. These will be used as importance proposals for μ_n and are assigned the incremental (normalized) weights:

$$W_n^j \propto \frac{d\mu_n}{d\mu_{n-1}}(u_{n-1}^j) = l_n(y_n; u_{n-1}^j),\quad \sum_{j=1}^N W_n^j = 1\,,\quad 1\le j\le N\,.$$

The weighting step is succeeded by a resampling step so as to discard samples with low weights. The particles are resampled probabilistically with replacement according to their weights W_n^j.

Mutation step. Carrying out only selection steps will eventually lead to degeneracy in the diversity of the particle population. During each successive resampling step, only a few parent particles will survive and copy themselves. Thus, one needs to reinsert the lost diversity in the particle population while maintaining the statistical properties of μ_n. This is achieved by evolving the particles independently by a small number of transitions of an MCMC kernel $\mathcal{K}_n$ such that $\mu_n\mathcal{K}_n = \mu_n$.

In Algorithm 2 we present the general purpose SMC algorithm that has appeared in [6]. For the resampling step, we have used $\mathcal{R}$ to denote the distribution of the indices of the parent particles. A simple choice for $\mathcal{R}$ is to use multinomial distribution, which we will follow in this paper, but other better performing choices are possible; see [13, 11] for more details. Recall also that u_n^j denotes the jth particle approximating μ_n, and in this paper it will be thought of as a concatenated vector of the real and imaginary parts of the Fourier coefficients in $\mathbb{Z}^2_{\uparrow}$ (or its finite truncation). Upon completion of step 2, one obtains particle approximations for μ_n:

$$\mu_n^N = \sum_{j=1}^N W_n^j \delta_{u_n^j}\,,$$

where δ_u denotes the Dirac point measure at $u\in U$. Many convergence results have been established for μ_n^N; we refer the reader to [11] for a book length review. Note that most steps in the algorithm allow for a trivial parallel implementation and hence very fast execution times; see [26] for more details. In addition, the resampling step is typically performed only when

Algorithm 2 Basic SMC

- At $n = 0$, for $j = 1, \ldots, N$ sample $u_0^j \sim \mu_0$.
- Repeat for $n = 1, \ldots, T$:
  1. Selection:
     (a) Importance sampling: weight particles $W_n^j \propto W_{n-1}^j \frac{d\mu_n}{d\mu_{n-1}}(u_{n-1}^j)$, $\sum_{j=1}^N W_n^j = 1$.
     (b) Resample (if required):
        i. Sample offsprings $(p_n^1, \ldots, p_n^N) \sim \mathcal{R}(W_n^1, \ldots, W_n^N)$.
        ii. Set $\breve{u}_n^j = u_{n-1}^{p_n^j}$ and $W_n^j = \frac{1}{N}$, $1 \le j \le N$.
  2. μ_n-invariant mutation: update $u_n^j \sim \mathcal{K}_n(\breve{u}_n^j, \cdot)$, $1 \le j \le N$, where $\mu_n \mathcal{K}_n = \mu_n$.

an appropriate statistic (commonly the effective sample size (ESS)) will indicate its necessity, e.g., when the ESS will drop below a prescribed threshold:

$$\mathrm{ESS}_n = \left(\sum_{j=1}^N \left(W_n^j \right)^2 \right)^{-1} < N_{thresh}.$$

When not resampling, particles keep their different weights W_n^j (and are not all set to $1/N$), which are then multiplied with the next incremental weights. In [6] the author uses the particle population to design $\mathcal{K}_n$ either as a standard random walk or as an independent Metropolis–Hastings sampler based on particle approximation μ_n^N. We will proceed by extending these ideas for our problem.

4. Extending SMC for high-dimensional inverse problems. One advantage of SMC is its inherent flexibility due to all different design elements, such as the sequence $(\mu_n)_{n=1}^T$ or the kernels $\mathcal{K}_n$. In high-dimensional applications such as data assimilation a user needs to design these carefully to obtain good performance. In addition, monitoring the performance also includes some challenges itself. We will deal with these issues in this section.

First, recall the equivalence between representing the initial vector field u by its Fourier coefficients and vice versa:

$$u \leftrightarrow \{u_k\}_{k \in \mathbb{Z}^2_\uparrow}; \quad u_k = \langle u, \psi_k \rangle, \ \bar{u}_k = -u_k .$$

In this section u will be treated again as the concatenated vector of the real and imaginary parts of its Fourier coefficients. In theory, this vector is infinite-dimensional, but in practice it will be finite- (but high-) dimensional due to the truncation in the Fourier space used in the numerical PDE solver. We will sometimes refer to the size of the implied mesh, d_u, informally as the dimensionality of u.

SMC as in Algorithm 2 needs to be adjusted so that it is effective especially when dealing with high-dimensional problems. One needs to ensure that both the importance sampling procedure is well behaved and mixing properties of the MCMC kernels are good enough to reintroduce every time the lost diversity due to resampling.

First, we need to ensure that the importance sampling weights (in step 1 of Algorithm 2) are "stable" in the sense that they exhibit low variance (and hence are not favoring one or relatively few particles). For high-dimensional inverse problems it is expected that this is not the case when the sequence $(\mu_n)_{n=1}^T$ is defined as in (3.2). We will modify the sequence of target distributions $(\mu_n)_{n=1}^T$ so that it evolves from the prior μ_0 to the posterior μ more smoothly. One way to achieve this is by bridging the two successive targets μ_{n-1} and μ_n via intermediate tempering steps as in [28]. So one can introduce a (possibly random) number, say q_n, of artificial intermediate targets between μ_{n-1} and μ_n:

$$\mu_{n,r} = \mu_{n-1}\left(\tfrac{d\mu_n}{d\mu_{n-1}}\right)^{\phi_{n,r}}, \tag{4.1}$$

where

$$0 = \phi_{n,0} < \phi_{n,1} < \cdots < \phi_{n,q_n} = 1 \tag{4.2}$$

are a sequence of user-specified temperatures. The accuracy of SMC when using such tempering schemes has been the topic of study in [3, 2, 16, 32].

For the SMC sequence implied jointly by (3.2) and (4.1), in section 4.1 we will present an adaptive implementation for choosing the next temperature on the fly [19] and in section 4.2 propose improved MCMC mutation kernels that use particle approximations for each $\mu_{n,r}$.

4.1. Stabilizing the weights with adaptive tempering. A particularly useful feature of using tempering within SMC is that one does not need to choose for every bridging sequence q_n and $\phi_{n,0}, \ldots, \phi_{n,q_n}$ before running the algorithm. In fact these can be decided on the fly using the particle population as it was originally proposed in [19]. Suppose at the moment the SMC algorithm is about to proceed to iteration n, r (the rth tempering step between μ_{n-1} and μ_n). The MCMC mutation step for temperature $\phi_{r-1,n}$ has just completed, and let $\{u_{n,r-1}^j\}_{j=1}^N$ be equally weighted particles[1] approximating $\mu_{n,r-1}$ as defined in (4.1). The next step is to use $\{u_{n,r-1}^j\}_{j=1}^N$ as importance proposals for $\mu_{n,r}$. The incremental weights are equal to $W_{n,r}^j \propto \frac{d\mu_{n,r}}{d\mu_{n,r-1}}(u_{n,r-1}^j)$, so if $\phi_{n,r}$ has been specified, they can be also written as

$$W_{n,r}^j = \frac{l_n(y_n; u_{n,r-1}^j)^{\phi_{n,r}-\phi_{n,r-1}}}{\sum_{s=1}^N l_n(y_n; u_{n,r-1}^s)^{\phi_{n,r}-\phi_{n,r-1}}}. \tag{4.3}$$

Now from the expression in (4.3) it follows that one can choose $\phi_{n,r}$ by imposing a minimum quality for the particle population after weighting, e.g., a minimum value for the ESS. Therefore we can use the particles $\{u_{n,r-1}^j\}_{j=1}^N$ to specify $\phi_{n,r}$ as the solution of the equation:

$$\mathrm{ESS}_{n,r}(\phi_{n,r}) = \left(\sum_{j=1}^N \left(W_{n,r}^j\right)^2\right)^{-1} \approx N_{thresh}. \tag{4.4}$$

[1]Here $u_{n,r-1}^j$ denotes the concatenated real vector of real and imaginary Fourier coefficients in $\mathbb{Z}_\uparrow^2$ for the jth particle targeting $\mu_{n,r-1}$. Often in the discussion we interpret $\mu_{n,r}$ as a probability measure on a similar real vector $u_{n,r}$.

If $\mathrm{ESS}_{n,r}(1) > N_{thresh}$, one should set $\phi_{n,r} = 1$ and proceed to the next tempering sequence leading to μ_{n+1}. Solving the above equation for $\phi_{n,r}$ can be easily implemented using an iterative bisection on $(\phi_{n,r-1}, 1]$ for the scalar function $\mathrm{ESS}_{n,r}(\phi)$. There is now only one user-specified parameter to be tuned, namely N_{thresh}. This adaptive tempering approach of [19] has been also used successfully in [31, 37, 5]. In [37] one may also find an alternative choice for the quality criterion instead of the ESS.

4.2. Improving the mixing of MCMC steps with adaptive scaling. We proceed by considering the design of the MCMC mutation steps to be used between tempering steps. We will design a random-walk-type method, tuned to the structure of the target distributions, by combining two ingredients: (a) we will use current information from the particles to adapt the proposal on the fly to the target distribution, and (b) we will distinguish between high and low frequencies for the MCMC formulation. The latter is a consideration specific to inverse problems related to dissipative PDEs, where the data often contains more information about the lower frequencies. We will look for a moment at the MCMC mutation kernel of Algorithm 2. Recall the correspondence between an element in U and its Fourier coefficients. To remove the effect of different scaling for each Fourier coefficient (due to the different variances in prior) we will consider the bijection

$$u \leftrightarrow \{\xi_k\}_{k \in \mathbb{Z}^2_\uparrow}$$

as implied by (2.4)–(2.5) for μ_0 or (3.3) for μ_n. As a result, a priori, $\mathrm{Re}(\xi_k)$, $\mathrm{Im}(\xi_k)$ for all $k \in \mathbb{Z}^2_\uparrow$ are independent and identically distributed samples from $\mathcal{N}(0,1)$. The proposal in step 1 of Algorithm 1 written in terms of ξ_k is

$$\widetilde{\xi}_k = \rho\, \xi_k + \sqrt{1-\rho^2}\, \mathfrak{Z}_k\,, \quad \mathrm{Re}(\mathfrak{Z}_k),\, \mathrm{Im}(\mathfrak{Z}_k) \overset{iid}{\sim} \mathcal{N}(0,1)\,, \quad k \in \mathbb{Z}^2_\uparrow\,. \tag{4.5}$$

This would be an excellent proposal when the target is very similar to the prior μ_0. When such a proposal is used within MCMC transition kernels for step 2 of Algorithm 2, then the mixing of the resulting mutation kernels will rapidly deteriorate as we move along the sequence $(\mu_n)_{n=1}^T$. The assimilated information from the observations will change the posterior densities for each ξ_k relative to the prior. In particular, often the data will contain a lot of information for the Fourier coefficients located at low frequencies, thereby shrinking their posterior variance. Thus, at these low frequencies the update in (4.5) will require a choice of ρ very close to 1 for the proposal $\widetilde{\xi}_k$ to have a nonnegligible chance to remain within the domain of the posterior and hence deliver nonvanishing acceptance probabilities. At the same time such small steps will penalize the mixing of the rest of the Fourier coefficients with relatively large posterior variances. This is a well-known issue often seen in practice [22], and somehow the scaling of the random-walk exploration for each frequency needs to be adjusted to the shape of the posterior it is targeting.

We will adapt the proposal to the different posterior scalings in the coefficients using the particles. Assume that the algorithm is currently at iteration n, r, where the importance sampling step with proposals from $\mu_{n-1,r}$ in (4.1) has been completed and we have the weighted particle set $\{u^j_{n,r-1}, W^j_{n,r}\}_{j=1}^N$ approximating $\mu_{n,r}$. We will construct the MCMC mutation kernel $\mathcal{K}_{n,r}$ (so that $\mu_{n,r}\mathcal{K}_{n,r} = \mu_{n,r}$) as follows. With a slight abuse of notation, we denote

by $u^j_{k,n,r-1}$ the bivariate real vector comprised of the real and imaginary parts of the kth Fourier coefficient of $u^j_{n,r-1}$, where $1 \le j \le N$ and $k \in \mathbb{Z}^2_{\uparrow}$. We estimate the marginal mean and covariance of the kth Fourier coefficient under the current target in the sequence $\mu_{n,r}$ as follows:

$$\mathfrak{m}^N_{k,n,r} = \sum_{j=1}^{N} W^j_{k,n,r} u^j_{k,n,r-1}\,, \quad \Sigma^N_{k,n,r} = \sum_{j=1}^{N} W^j_{k,n,r} (u^j_{k,n,r-1} - \mathfrak{m}^N_{k,n,r})(u^j_{k,n,r-1} - \mathfrak{m}^N_{k,n,r})'\,. \tag{4.6}$$

The estimated moments $\mathfrak{m}^N_{k,n,r}$, $\Sigma^N_{k,n,r}$ (and the corresponding Gaussian approximation of the posterior) will be used to provide the scaling of the step size of the random walk for each k. Let $\{\breve{u}^j_{n,r}\}_{j=1}^N$ be the collection of particles obtained after resampling $\{u^j_{n,r-1}\}_{j=1}^N$ with replacement according to the weights $W^j_{n,r}$. In the MCMC mutation step we can use the following proposal instead of (4.5):

$$\widetilde{u}_{k,n,r} = \mathfrak{m}^N_{k,n,r} + \rho\,(\breve{u}^j_{k,n,r} - \mathfrak{m}^N_{k,n,r}) + \sqrt{1-\rho^2}\,\mathcal{N}(0, \Sigma^N_{k,n,r})\,. \tag{4.7}$$

Notice that in (4.7) we propose moving the real and imaginary parts of the Fourier coefficients separately for each frequency $k \in \mathbb{Z}^2_{\uparrow}$. This requires computing only the sample mean and covariances of the kth marginal of $\mu_{n,r}$. Alternatively, one could try to jointly estimate the high-dimensional covariance matrices $\Sigma^N_{n,r}$ (for the joint vector $u_{n,r}$ involving every k). Although this can perform better, some regularization might be required, and we advise some caution when the number of particles is moderate because of the Monte Carlo variance in estimating high-dimensional covariances. A pragmatic option in this case is to use only the diagonal elements of the joint covariance estimator or possibly include some off-diagonal terms where (partial) correlations are high.

The second ingredient of the proposed MCMC mutation kernel involves distinguishing between low and high frequencies. In particular, we will use the proposal of (4.7) for a window of the Fourier coefficients with relatively low frequencies and the standard proposal of (4.5) that uses the prior for the higher frequencies. This modification ensures that the MCMC kernel will be robust to mesh refinement and valid for the infinite-dimensional problems considered here. In addition, we have found empirically that this hybrid approach gives a better balance between adaptation and variability caused by Monte Carlo error in estimating empirical covariances. We will use the proposal of (4.7) for coefficients with frequencies in the rectangular window defined as

$$\mathbf{K} = \left\{k \in \mathbb{Z}^2 \setminus \{0\} :\ k_1 \vee k_2 \le K\right\}, \tag{4.8}$$

where $k_1 \vee k_2 = \max\{k_1, k_2\}$. It would certainly be possible to use an alternative definition for $\mathbf{K}$; see [21] for more ideas and extensions.

The proposed MCMC kernel is presented in Algorithm 3. For simplicity, in the notation we omit subscripts n, r when writing $u(m), u$ for $u_{n,r}(m), u_{n,r}$. We also use subscripts L, H to refer to the collection of concatenated vectors of real/imaginary parts of Fourier coefficients in $\mathbf{K} \cap \mathbb{Z}^2_{\uparrow}$ and $\mathbf{K}^c \cap \mathbb{Z}^2_{\uparrow}$, respectively, and I is a 2×2 unit matrix. Notice that even with adaptation, a few MCMC iterations (denoted by $M \ge 1$) might be required to introduce enough diversity to the particle population (e.g., 10–30).

Algorithm 3 A $\mu_{n,r}$-Invariant MCMC Mutation Kernel $\mathcal{K}_{n,r}(\mathfrak{u},\cdot)$

1. Initialize $u_{n,r}(0)=\mathfrak{u}$ (when in step 3(f) of Algorithm 4, set $u_{n,r}(0)=\breve{u}^j_{n,r}$). Let $\mathfrak{m}^N_{k,n,r}$, $\Sigma^N_{k,n,r}$ be known approximations for all $\mathbf{K}\cap\mathbb{Z}^2_{\uparrow}$. Choose $\rho_L,\rho_H\in(0,1)$.
2. For $m=0,\ldots,M-1$:
   (a) For $k\in\mathbf{K}\cap\mathbb{Z}^2_{\uparrow}$, propose the update:
   $$\widetilde{u}_k=\mathfrak{m}^N_{k,n,r}+\rho_L(u_k(m)-\mathfrak{m}^N_{k,n,r})+\sqrt{1-\rho_L^2}\,\mathcal{N}(0,\Sigma^N_{k,n,r})\,;$$
   for $k\in\mathbf{K}^c\cap\mathbb{Z}^2_{\uparrow}$, propose the update:
   $$\widetilde{u}_k=\rho_H\,u_k(m)+\sqrt{1-\rho_H^2}\,\mathcal{N}(0,\tfrac{1}{2}\beta^2|k|^{-2\alpha}\,I)\,.$$
   (b) Compute forward dynamics $G^{(n)}_{\delta}(\widetilde{u})$ and likelihood functions $l_1(y_1;\widetilde{u}),\ldots,l_n(y_n;\widetilde{u})$.
   (c) With probability
   $$1\wedge\frac{l_{n,r}(\widetilde{u})}{l_{n,r}(u(m))}\frac{\mu_0(\widetilde{u}_L)}{\mu_0(u_L(m))}\frac{\mathcal{Q}(\widetilde{u}_L,u_L(m))}{\mathcal{Q}(u_L(m),\widetilde{u}_L)},\tag{4.9}$$
   accept the proposal and set $u(m+1)=\widetilde{u}$; otherwise set $u(m+1)=u(m)$. We use
   $$l_{n,r}(u)=l_n(y_n;u)^{\phi_{n,r}}\prod_{s=1}^{n-1}l_s(y_s;u)\,,\tag{4.10}$$
   $$\mu_0(u_L)=\exp\left\{-\sum_{k\in\mathbf{K}\cap\mathbb{Z}^2_{\uparrow}}\beta^{-2}|k|^{2\alpha}|u_k|^2\right\},\tag{4.11}$$
   $$\mathcal{Q}(u_L,\tilde{u}_L)=\exp\left\{-\tfrac{1}{2(1-\rho_L^2)}\sum_{k\in\mathbf{K}\cap\mathbb{Z}^2_{\uparrow}}(\tilde{u}_k-\mathfrak{m}^N_{k,n,r}-\rho_L(u_k-\mathfrak{m}^N_{k,n,r}))'\right.\tag{4.12}$$
   $$\left.\times(\Sigma^N_{k,n,r})^{-1}(\tilde{u}_k-\mathfrak{m}^N_{k,n,r}-\rho_L(u_k-\mathfrak{m}^N_{k,n,r}))\right\}.$$
3. Output $u(M)$ as a sample from $\mathcal{K}_{n,r}(\mathfrak{u},\cdot)$.

4.3. The complete algorithm. The complete SMC algorithm is presented in Algorithm 4. After the completion of step 4 of Algorithm 4, the particles can be used to approximate the intermediate posteriors μ_n; this is emphasized in step 4 by denoting $u^j_{n,r}=u^j_n$ when $\phi_{n,r}=1$. In addition, when $\phi_{n,r}=1$, the resampling steps can be omitted whenever ESS $>N_{thresh}$, as mentioned in section 3.2.

Although standard SMC methods have a well-developed theoretical framework for their

Algorithm 4 An SMC Algorithm for High-Dimensional Inverse Problems

- At $n = 0$, for $j = 1, \dots N$ sample $u_{0,0}^j \sim \mu_0$.
- For $n = 0, \dots, T$
  1. For $j = 1, \dots N$ compute forward dynamics $G_\delta \circ G_\delta^{(n-1)}(u_{n,0}^j)$ and $l_n(y_n, u_{n,0}^j)$.
  2. Set $r = 0$ and $\phi_{n,0} = 0$.
  3. While $\phi_{n,r} < 1$
     (a) Increase r by 1.
     (b) (Compute temperature)
     IF $\min_{\phi \in (\phi_{n,r-1},1]} \mathrm{ESS}_{n,r}(\phi) > N_{thresh}$, set $\phi_{n,r} = 1$,
     ELSE compute $\phi_{n,r}$ such that
     $$\mathrm{ESS}_{n,r}(\phi_{n,r}) \approx N_{thresh}$$
     using a bisection on $(\phi_{n,r-1}, 1]$.
     (c) (Compute weight) for $j = 1, \dots N$:
     $$W_{n,r}^j = \frac{l_n(y_n; u_{n,r-1}^j)^{\phi_{n,r}-\phi_{n,r-1}}}{\sum_{s=1}^N l_n(y_n; u_{n,r-1}^s)^{\phi_{n,r}-\phi_{n,r-1}}}.$$
     (d) (Moment estimates) compute $\mathfrak{m}_{k,n,r}^N$ and $\Sigma_{k,n,r}^N$ for $k \in \mathbf{K} \cap \mathbb{Z}_\uparrow^2$ as in (4.6).
     (e) (Resample) let $(p_n^1, \dots, p_n^N) \sim \mathcal{R}(W_n^1, \dots, W_n^N)$. For $j = 1, \dots, N$ set $\breve{u}_{n,r}^j = u_{n,r-1}^{p_n^j}$ and $W_{n,r}^j = \frac{1}{N}$.
     (f) ($\mu_{n,r}$-invariant mutation) for $j = 1, \dots N$, sample $u_{n,r}^j \sim \mathcal{K}_{n,r}(\breve{u}_{n,r}^j, \cdot)$ (see Algorithm 3).
  4. Set $u_{n+1,0}^j = u_{n,r}^j (:= u_n^j)$.

justification, this literature is much less developed in the presence of the critical adaptive steps considered in Algorithm 4. When both adaptive scaling of MCMC steps and adaptive tempering are considered together, we refer the reader to [4] for asymptotic convergence results (in N). Relevant nonasymptotic results can be found in [16], where the authors consider only adaptive tempering and assume that ideal MCMC kernels with sufficient mixing are available.

As regards the computational cost, for each MCMC mutation at iteration n, r, we need to run the PDE numerical solver M times from $t = 0$ up to the current time $t = n\delta$. Therefore, the total computational cost is proportional to κMNT^2, where κ depends on the random number of tempering steps. In fact, this cost is significantly reduced when more tempering steps are required for small values of n. Finally, the memory requirements are $\mathcal{O}(N)$, as there is no need to store past particles at each step of the algorithm.

4.4. Monitoring the performance: Towards an automatic design. Although SMC is a generic approach suitable for a wide class of problems, this flexibility also means that the user has to select many design elements. We proceed by outlining different measures of performance for some of the remaining tuning parameters.

For ρ_H, ρ_L, we recommend that they should be tuned so that at time T the acceptance ratio in (4.9) averaged over the particles has a reasonable value, e.g., 0.1–0.3. Having said that, the average value of the acceptance ratio should be recorded and monitored for the complete run of the SMC. The same applies for the ESS, which will also reveal how much tempering is used during the SMC run.

It is critical to choose M so as to provide sufficient diversity in the particle population. A question often raised when using SMC samplers is whether a given value of M is adequate. For each frequency k, we propose the following measure for monitoring the movement of the particles in the population during the MCMC steps at iteration n, r:

$$J_{k,n,r} = \frac{\sum_{j=1}^{N} \left| u_{k,n,r}^{j}(M) - u_{k,n,r}^{j}(0) \right|^2}{2 \sum_{j=1}^{N} \left| u_{k,n,r}^{j}(0) - \mu_{k,n,r}^{N} \right|^2}. \tag{4.13}$$

We remark that the statistic $J_{k,n,r}$ has not been used before for SMC to the best of our knowledge. Of course, monitoring every value of k is not necessary, and one could in practice choose a small number of representative frequencies from their complete set. In addition, as N increases, $J_{k,n,r}$ will converge to $1 - \mathrm{corr}(u_{k,n,r}(M), u_{k,n,r}(0))$. Hence, statistical intuition can explain what requirements we should pose for $J_{k,n,r}$, e.g., that it should be at least above 0.01–0.05. In our context the MCMC mutation steps are applied to jitter the population at each of the many steps of the SMC sampler. The role of the MCMC steps here is very different from that in a full MCMC sampler, where the number of MCMC steps has to be large enough for ergodic theory to apply. However, popular monitoring statistics from MCMC such as the autocorrelation for some fixed lag could be related to $J_{k,n,r}$ or potentially used instead of $J_{k,n,r}$, but these need to be averaged over the particle population.

It is also possible to empirically validate whether the chosen value for K, the half-width of $\mathbf{K}$, was appropriate. A revealing plot here is the 2D heat map of the ratio of the variance of the posterior to the variance of the prior against k. $\mathbf{K}$ should include as much as possible the region where this ratio is significantly less than 1, e.g., say less than 0.8.

A particularly interesting point is that summaries like the average acceptance ratio, the statistic $J_{k,n,r}$, or the ratio of the posterior to prior variance can be also be potentially used within decision rules to determine ρ_L, ρ_H, M, and K adaptively on the fly. This can be implemented similarly to how the ESS is used for adaptively choosing the tempering temperatures. An approach for automating the tuning of ρ_L, ρ_H can be found in [19] and a more detailed discussion for the remaining parameters in [21].

5. Numerical examples.

We will use numerical solutions for the Navier–Stokes PDE in (2.1) given an initial condition. We employ a method based on a spectral Galerkin approximation of the velocity field in a divergence-free Fourier basis [18]. The convolutions arising from products in the nonlinear term are computed via FFTs on a 64^2 grid with additional antialiasing using double-sized zero padding [35]. In addition, exponential time differencing is used as in [9], whereby an analytical integration is used for the linear part of the PDE, $\nu A v$, together with an explicit numerical Euler integration scheme for the nonlinear part, $P(f) - B(v, v)$.

In this section we will present two numerical examples using two different synthetic data-sets obtained from some corresponding true initial vector fields, $u^\dagger$. The first example will use Data-set A consisting of few blocks of observations obtained at close time intervals (small δ), but each block is obtained from a dense observation grid (high Υ). For this example we will compare the performance of our method with benchmark results from the MCMC approach described in Algorithm 1. For the second example we will use Data-set B, a longer data-set with blocks of observations spaced apart by longer time periods δ, each originating from a sparser observation grid (low Υ). In both cases, the total number of observations of the vector field will be the same and $\Upsilon T = 80$. We summarize these details in Table 1.

Table 1

The specification of the two data-sets considered in the numerical examples. Data-set A corresponds to a scenario of a short-time data-set with a dense observation grid and Data-set B to the scenario of a long data-set with a sparse observation grid. The true initial condition $u^\dagger$ was sampled from the specified prior.

Data-set A	$\delta = 0.02\ ,\ \Upsilon = 16\ ,\ \ T = 5$	$u^\dagger \sim \mathcal{N}(0,\, \beta^2 A^{-\alpha})\ ,\ \beta^2 = 5,\ \alpha = 2.2$
Data-set B	$\delta = 0.2\ \ ,\ \Upsilon = 4\ ,\ \ T = 20$	$u^\dagger \sim \mathcal{N}(0,\, \beta^2 A^{-\alpha})\ ,\ \beta^2 = 1,\ \alpha = 2$

The two data-sets are synthesized using the numerical PDE solver described above. In both cases we have set $\nu = 0.02$, $f(x) = \nabla^\perp \cos\big((5,5)' \cdot x\big)$, $\gamma^2 = 0.2$, where we remind the reader that ν is the viscosity, f the external forcing, and γ^2 the observation noise variance. Adjoint PDE solvers will be used in the sense that the same numerical solver is used for synthesizing the data and in the Monte Carlo inference algorithms.

In Table 2 we summarize the computational cost of the algorithms used in our experiments. The table presents the number of times a PDE solution is required and the total execution time. We do not provide an MCMC benchmark for Data-set B, as the more expensive PDE solver needed in this case would result in an enormous execution time for Algorithm 1. For SMC, we will present results from an implementation of Algorithm 4 with trivial parallelization, whereby the resampling step is performed at a single computing node that collects and distributes all particles. For Data-set A, in Table 2 we also show the computational cost from a typical run of SMC with $N = 500$ but without using parallelization. Although in the remainder of this section we will not present the actual results from this run, we report that the performance was comparable to MCMC as well as SMC with higher N obtained via the parallel implementation. In a way this demonstrates the efficiency of the SMC method compared to MCMC, but we have to emphasize that for more realistic applications parallelization is critical for effective execution times. We proceed by looking at each example with more detail.

5.1. Data-set A: Short-time data-set with a dense observation grid. Figure 1 plots the posterior mean of the vorticity and velocity fields for the initial condition as estimated by our adaptive SMC algorithm (left) and the MCMC one (right). In the same plot the true field $u^\dagger$ and its vorticity are displayed in the middle. The results from SMC and MCMC are very similar, and both methods manage to capture the main features of the true field $u^\dagger$. The smoothing effect observed for the posterior means by both SMC and MCMC appears because the observations cannot provide substantial information about the high frequency Fourier coefficients.

Note that the objective here is to approximate the full posterior and not just the mean.

Table 2

We present the number of times a numerical PDE solution of total length δ is required by each algorithm. This number is divided by T. The total execution time is also shown for each case. For the parallel implementation of Algorithm 4 we used trivial parallelization (except for the resampling step). The code was written in MATLAB and parallel implementations of SMC run as a parallel MPI job with 60 workers on the computing cluster of CSML-UCL (SunGrid engine). All other simulations were performed in MATLAB on the same computer running Linux with an Intel Xeon CPU E5-1660 at 3.30 GHz (six core) and 16 GB RAM.

	Algorithmic parameters	Number of calls of PDE solver	Execution time
MCMC Data-set A	$\rho = 0.9998$	0.9×10^6	9 days
SMC Data-set A no parallelization	$N = 500$, $N_{thresh} = \frac{N}{3}$, $M = 20$ $\rho_L = 0.99$, $\rho_H = 0.991$, $K = 7$	7.266×10^5	3 days
SMC Data-set A with parallelization	$N = 1,020$, $N_{thresh} = \frac{N}{3}$, $M = 20$ $\rho_L = 0.99$, $\rho_H = 0.991$, $K = 7$	1.403×10^6	7.4 hours
SMC Data-set B with parallelization	$N = 1,020$, $N_{thresh} = \frac{N}{3}$, $M = 20$ $\rho_L = 0.99$, $\rho_H = 0.991$, $K = 7$	1.447×10^6	3.5 days

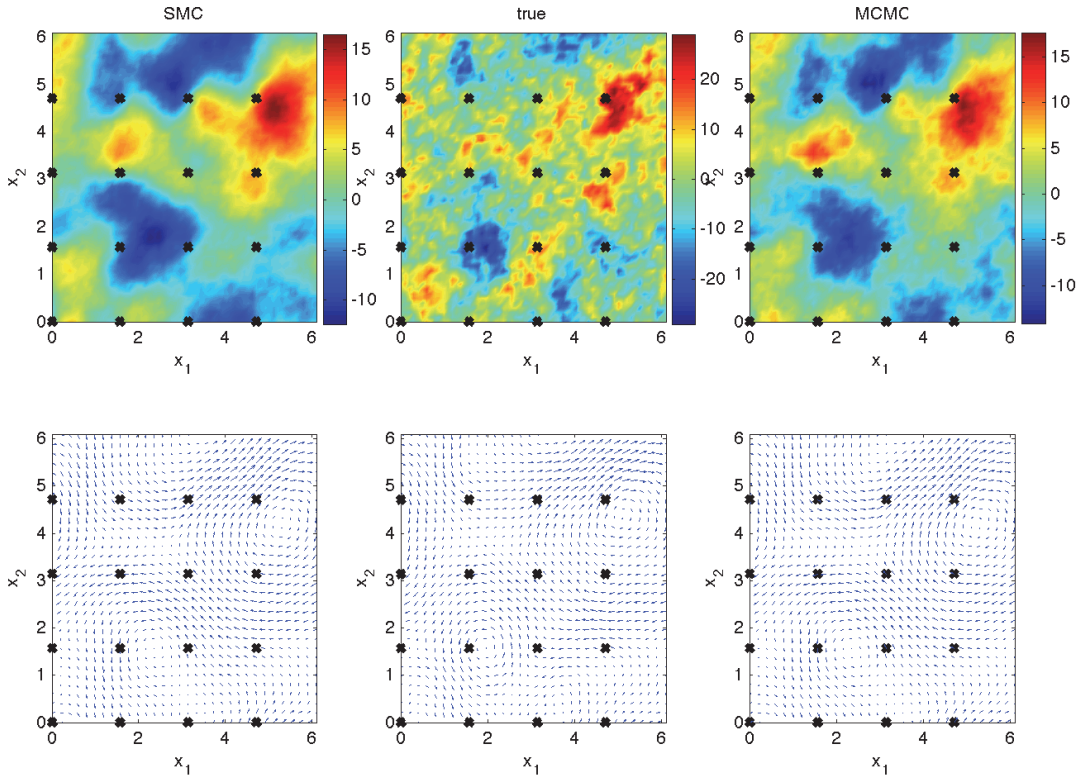

Figure 1. *Data-set A. Top panel: left, posterior mean of initial vorticity from SMC; center, true initial vorticity; right, posterior mean of initial vorticity from MCMC. Bottom panel: corresponding graphs for the velocity fields with the same order from left to right. The crosses indicate the positions $x_1, \ldots, x_\Upsilon$ where the vector field is observed.*

Figure 2 shows the estimated posterior density functions (PDFs) for a few (rescaled) Fourier coefficients, $\xi_{k,T}$ (as defined in (3.3)), of different frequencies k obtained using both SMC and MCMC. Recall the scaling in $\xi_{k,0}$ makes all prior densities equal to standard normals. In Figure 2 we plot the prior density (dotted), the posterior densities from SMC (solid), and MCMC (dashed) together with the true value of $\xi^{\dagger}_{k,T}$ used for generating the data (vertical line).

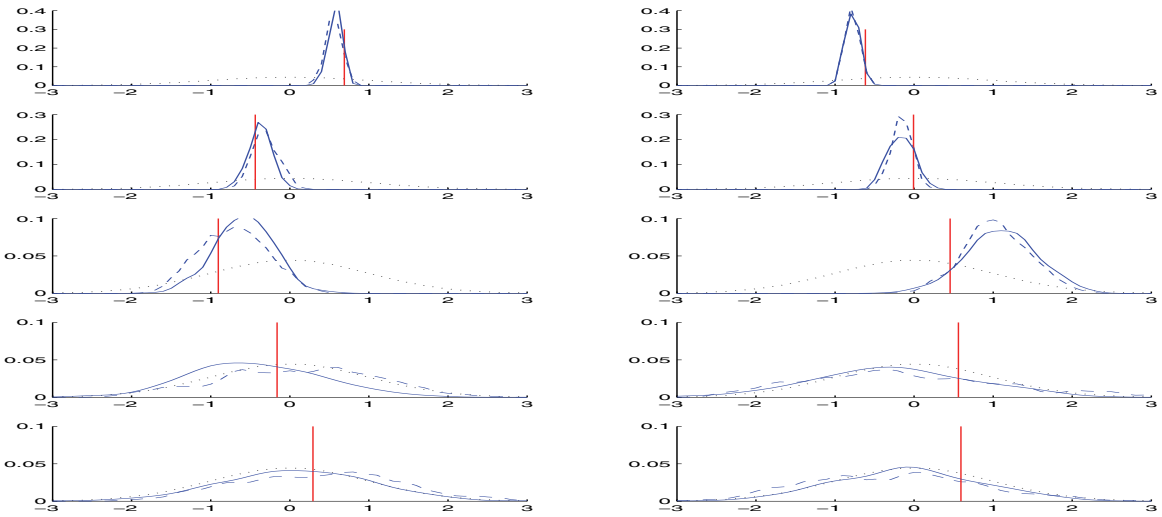

Figure 2. *Data-set A. Estimated PDFs: Blue lines are the estimated posterior densities for $\xi_{k,T}$; the solid lines are for SMC, the dashed ones are for MCMC, and the dotted (black) lines correspond to the prior densities. The left panel is for the real parts and right panel for the imaginary parts. The different rows correspond to each of the frequencies $k = (0,1),(1,1),(2,1),(4,4),(9,9)$ from top to bottom. The red vertical lines designate the true values $\xi^{\dagger}_{k,T}$.*

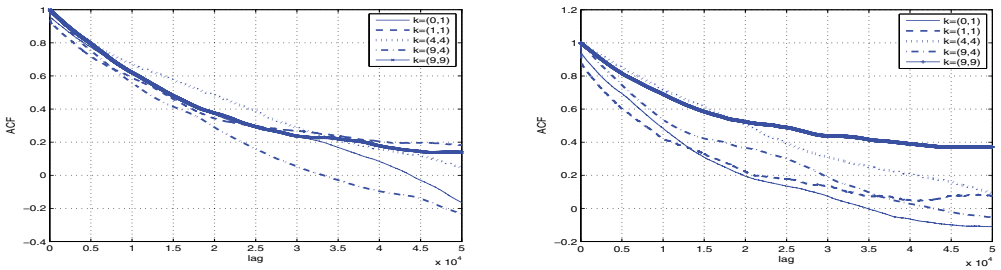

Figure 3. *Data-set A: monitoring MCMC performance. Autocorrelation plots from the MCMC trajectory of $\xi_{k,T}$ for a number of different frequencies; the left graph corresponds to the real parts and the right one to the imaginary ones.*

We proceed by presenting different measures of performance for MCMC and SMC. In Figure 3 we plot the autocorrelations of the MCMC trajectory for different Fourier coefficients.

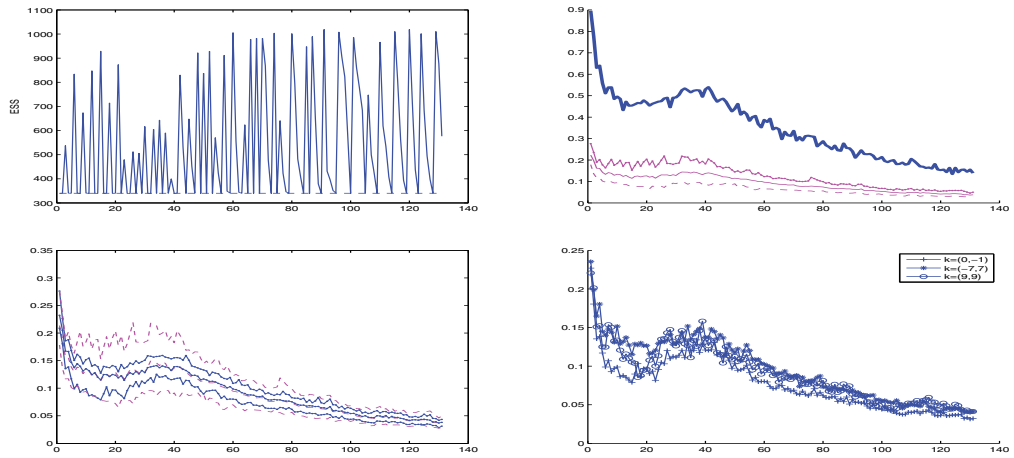

Figure 4. *Data-set A: monitoring SMC performance with $N = 1020$. In all plots the horizontal axis is the index of SMC iteration n, r. Top left: ESS oscillating between N_{thresh} (when $\phi_{n,r} < 1$) and higher values (when $\phi_{n,q_n} = 1$). Top right: thick-solid (blue) is average acceptance ratio (w.r.t. the particles), dot-solid (magenta) is $\max_k J_{k,n,r}$, solid (magenta) is the average of $J_{k,n,r}$ (w.r.t .k), dotted (magenta) is $\min_k J_{k,n,r}$. Bottom left: We plot again the maximum, minimum, and average of $J_{k,n,r}$ w.r.t. k separately for $k \in \mathbf{K} \cap \mathbb{Z}^2_\uparrow$ (dash-dot, blue) and $k \in \mathbf{K}^c \cap \mathbb{Z}^2_\uparrow$ (dashed, magenta). Bottom right: $J_{k,n,r}$ for $k = (0,-1), (-7,7), (9,9)$.*

The mixing of the MCMC chain is quite slow, and hence a large number of iterations was required for the MCMC approach to deliver reliable results. To monitor the performance of the SMC algorithm, Figure 4 includes plots of the ESS, the average acceptance ratio, and the indicator $J_{k,n,r}$ against each SMC iteration[2] index n, r. Compared to the size of the data-set ($\Upsilon T = 80$) the total number of extra tempering steps required here was about 50. The majority of tempering steps was performed early on when assimilating the first few observations. At later stages gradually less tempering was required with SMC being able to assimilate 2–3 data points without resampling during the final stages. This is beneficial in terms of efficiency too, as this extra computational effort in the earlier stages of the adaptive algorithm requires much shorter numerical solutions of the PDE.

In the bottom left plot in Figure 4 we observe how the average, maximum, and minimum (over k) of $J_{k,n,r}$ change with n, r separately when k is within or outside the window of frequencies $\mathbf{K}$. For some indicative values of k we also show $J_{k,n,r}$ in the lower right plot of Figure 4. $J_{k,n,r}$ does not seem to vary a lot with $|k|$. It is certainly reassuring that all the MCMC steps seem to deliver a considerable amount of diversity to all Fourier coefficients. Also, the amount of diversity appears to be fairly evenly spread over all Fourier coefficients, even if different MCMC proposals are used within and outside the window $\mathbf{K}$. Although supporting plots are not shown here, $J_{k,n,r}$ seemed to grow linearly with M for every k and each n, r.

[2]By SMC iteration index n, r we actually mean iteration $r + \sum_{p=0}^{n-1} q_p$, i.e., the number of times steps 1–4 of Algorithm 4 have completed. Note that q_p is as in (4.2) and is a random variable determined by the algorithm.

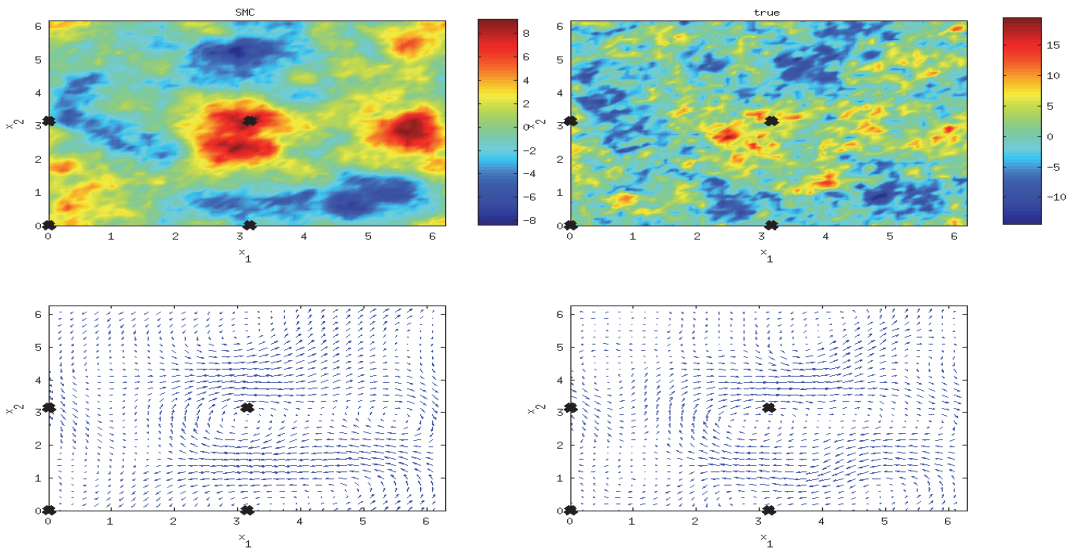

Figure 5. *Data-set B: Vorticity (top) and velocity field (bottom); posterior mean (left) as estimated by SMC with $N = 1020$ and true values (right). The crosses indicate the positions $x_1, \ldots, x_\Upsilon$ where the vector field is observed. The graph is similar to Figure 1 for Data-set A.*

5.2. Data-set B: Long-time data-set with a sparse observation grid. We will now present the results of the SMC method of Algorithm 4 when applied to Data-set B. This scenario is more challenging than the one with Data-set A, as we allow the Navier–Stokes dynamics to evolve for a longer period of time. We will follow a presentation similar to that in the previous example. Figure 5 plots the posterior mean of the initial vorticity and velocity field. This plot can be used to compare the SMC estimates method versus the true values corresponding to $u^\dagger$. Although here we do not have a benchmark available like before, the smoothing effect in the estimates relative to the truth does not seem surprising based on the intuition gained from the previous example.

Figure 6 displays the approximate posterior densities of $\xi_{k,n}$ for a number of frequencies k. The difference compared to Figure 2 for the previous example is that now we can see how μ_n changes for $n = 0$ (dotted), $0.5T$ (dash-dotted), $0.75T$ (dashed), and T (solid). As expected, each new block of observations contributes to shaping a more informative posterior. To monitor the performance of SMC, in Figure 7 we plot the ESS, average acceptance ratio, and $J_{k,n,r}$ all against n, r as we did for the previous example. The algorithm uses almost the same number of tempering steps in total compared to the previous example, and again most are needed in earlier epochs n. In addition, the acceptance ratio and $J_{k,n,r}$ stop decreasing after some iteration. We interpret this as a sign that μ_n stops changing fast with n and that the particles form good approximations of the targeted sequence.

Finally, Figure 8 shows the heat maps against k of the estimated posterior means of $\xi_{k,T}$ and the ratio of their marginal posterior standard deviations over their prior values. Most information gain from the observations appears in the posterior at low frequencies, and the choice of $K = 7$ seems to be justified. Finally, for each of the presented examples one can find

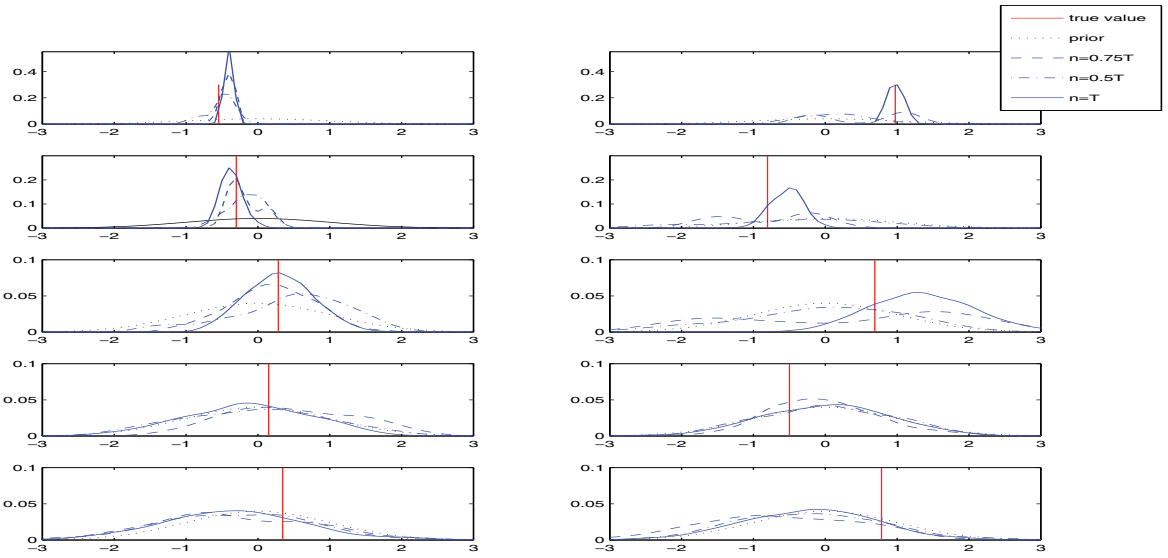

Figure 6. *Data-set B: Estimated posterior PDFs for* $\xi_{k,n}$ *for* $n = 0, 0.5T, 0.75T, T$ *and frequencies* $k = (0,1),(1,1),(2,1),(4,4),(9,9)$. *The details are similar to Figure 2 for Data-set A.*

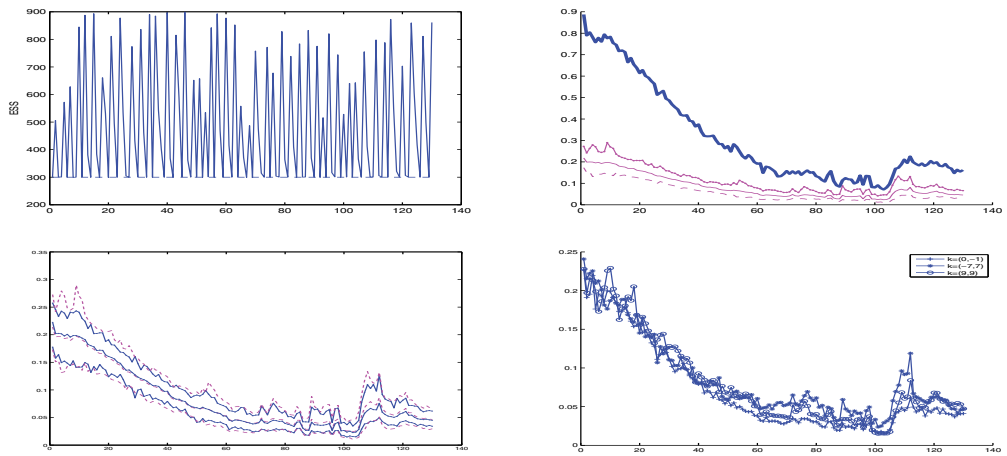

Figure 7. *Data-set B: monitoring SMC performance against iteration* n, r. *The details are similar to Figure 4.*

more figures displayed in [21].

6. Conclusion. This paper aims to make a significant contribution towards demonstrating that SMC can be useful for high-dimensional inverse problems. The added efficiency of our method compared to plain MCMC can be attributed to being able to employ a variety of adaptation steps that take advantage of the evolving particle population, hence tuning the algorithm effectively to the structure of the target distributions in the SMC sequence. SMC algorithms are also appealing to practitioners given the inherent ability to parallelize many

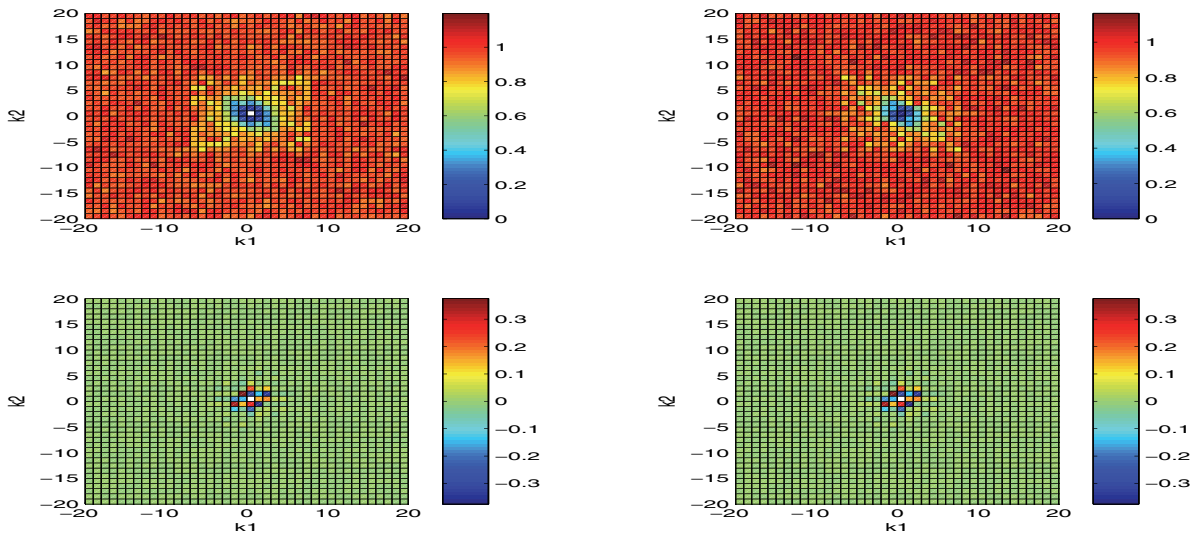

Figure 8. *Data-set B: posterior versus prior statistics for $\xi_{k,T}$ using SMC with $N = 1,020$ particles. Top: heat map of the ratio of estimated posterior (marginal) standard deviations of $\xi_{k,T}$ over the standard deviations of each $\xi_{k,0}$ (prior) against all frequencies k. Bottom: corresponding heat map for the mean of each $\xi_{k,T}$ against k. The left and right plots correspond to the real and imaginary parts, respectively, of the Fourier coefficients.*

steps in the algorithm, thus drastically reducing execution times. As regards to understanding the effect of each block of observations, another useful aspect of the method is that the SMC sequence allows for monitoring the evolution of posterior distributions of interest as more observations arrive. In contrast, MCMC methods would require rerunning the algorithm from scratch. In terms of the accuracy of the estimates we believe that SMC can be on par with expensive MCMC methods, which is illustrated clearly in the example with Data-set A.

Acknowledgments. We would like to thank D. O. Crisan, K. J. H. Law, and A. M. Stuart for many discussions that provided valuable insight on the topic and V. Stathopoulos for his help on using MATLAB MPI jobs on the CSML cluster at UCL.

REFERENCES

[1] A. F. Bennett, *Inverse Modeling of the Ocean and Atmosphere*, Cambridge University Press, Cambridge, UK, 2002.

[2] A. Beskos, D. Crisan, and A. Jasra, *On the stability of sequential Monte Carlo methods in high dimensions*, Ann. Appl. Probab., 24 (2014), pp. 1396–1445.

[3] A. Beskos, D. Crisan, A. Jasra, and N. Whiteley, *Error bounds and normalising constants for sequential Monte Carlo samplers in high dimensions*, Adv. in Appl. Probab., 46 (2014), pp. 279–306.

[4] A. Beskos, A. Jasra, N. Kantas, and A. H. Thiéry, *On the Convergence of Adaptive Sequential Monte Carlo Methods*, preprint, arXiv:1306.6462v3, 2014.

[5] L. Bornn, A. Doucet, and R. Gottardo, *An efficient computational approach for prior sensitivity analysis and cross-validation*, Canad. J. Statist., 38 (2010), pp. 47–64.

[6] N. Chopin, *A sequential particle filter method for static models*, Biometrika, 89 (2002), pp. 539–552.
[7] A. J. Chorin, M. Morzfeld, and X. Tu, *Implicit particle filters for data assimilation*, Commun. Appl. Math. Comput. Sci., 5 (2010), pp. 221–240.
[8] S. L. Cotter, G. O. Roberts, A. M. Stuart, and D. White, *MCMC methods for functions: Modifying old algorithms to make them faster*, Statist. Sci., 28 (2013), pp. 424–446.
[9] S. M. Cox and P. C. Matthews, *Exponential time differencing for stiff systems*, J. Comput. Phys., 176 (2002), pp. 430–455.
[10] G. Da Prato and J. Zabczyk, *Stochastic Equations in Infinite Dimensions*, Cambridge University Press, Cambridge, UK, 2008.
[11] P. Del Moral, *Feynman-Kac Formulae*, Springer, New York, 2004.
[12] P. Del Moral, A. Doucet, and A. Jasra, *Sequential Monte Carlo samplers*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 68 (2006), pp. 411–436.
[13] A. Doucet, N. de Freitas, and N. Gordon, eds., *Sequential Monte Carlo Methods in Practice*, Stat. Eng. Inf. Sci., Springer-Verlag, New York, 2001.
[14] G. Evensen, *Data Assimilation: The Ensemble Kalman Filter*, 2nd ed., Springer, Berlin, 2009.
[15] C. Foias, O. Manley, R. Rosa, and R. Temam, *Navier-Stokes Equations and Turbulence*, Encyclopedia Math. Appl. 83, Cambridge University Press, Cambridge, UK, 2001.
[16] F. Giraud and P. Del Moral, *Non-asymptotic Analysis of Adaptive and Annealed Feynman-Kac Particle Models*, preprint, arXiv:1209.5654, 2012.
[17] M. Hairer, A. Stuart, and S. Vollmer, *Spectral gaps for a Metropolis-Hastings algorithm in infinite dimensions*, Ann. Appl. Probab., to appear.
[18] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb, *Spectral Methods for Time-Dependent Problems*, Cambridge Monogr. Appl. Comput. Math. 21, Cambridge University Press, Cambridge, UK, 2007.
[19] A. Jasra, D. A. Stephens, A. Doucet, and T. Tsagaris, *Inference for Lévy-driven stochastic volatility models via adaptive sequential Monte Carlo*, Scand. J. Stat., 38 (2011), pp. 1–22.
[20] J. P. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Appl. Math. Sci. 160, Springer, New York, 2005.
[21] N. Kantas, A. Beskos, and A. Jasra, *Sequential Monte Carlo Methods for High-Dimensional Inverse Problems: A Case Study for the Navier-Stokes Equations*, preprint, arXiv:1307.6127, 2013.
[22] K. J. H. Law, *Proposals which speed up function-space MCMC*, J. Comput. Appl. Math., 262 (2014), pp. 127–138.
[23] K. J. H. Law and A. M. Stuart, *Evaluating data assimilation algorithms*, Mon. Wea. Rev., 140 (2012), pp. 3757–3782.
[24] F.-X. Le Dimet and O. Talagrand, *Variational algorithms for analysis and assimilation of meteorological observations: Theoretical aspects*, Tellus A, 38 (1986), pp. 97–110.
[25] F. Le Gland, V. Monbet, and V.-D. Tran, *Large sample asymptotics for the ensemble Kalman filter*, in The Oxford Handbook of Nonlinear Filtering, D. Crisan and B. Rozovskii, eds., Oxford University Press, Oxford, UK, 2011, pp. 598–634.
[26] A. Lee, C. Yau, M. B. Giles, A. Doucet, and C. C. Holmes, *On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods*, J. Comput. Graph. Statist., 19 (2010), pp. 769–789.
[27] R. M. Neal, *Regression and classification using Gaussian process priors*, in Bayesian Statistics, 6 (Alcoceber, 1998), J. M. Bernardo, J. O. Berger, A. F. M. Smith, and A. P. Dawid, eds., Oxford University Press, Oxford, UK, 1999, pp. 475–491.
[28] R. M. Neal, *Annealed importance sampling*, Stat. Comput., 11 (2001), pp. 125–139.
[29] J. C. Robinson, *Infinite-Dimensional Dynamical Systems: An Introduction to Dissipative Parabolic PDEs and the Theory of Global Attractors*, Cambridge Texts Appl. Math., Cambridge University Press, Cambridge, UK, 2001.
[30] Y. Sasaki, *An objective analysis based on the variational method*, J. Meteor. Soc. Japan, 36 (1958), pp. 77–88.
[31] C. Schäfer and N. Chopin, *Sequential Monte Carlo on large binary sampling spaces*, Stat. Comput., 23 (2013), pp. 163–184.
[32] N. Schweizer, *Non-asymptotic Error Bounds for Sequential MCMC and Stability of Feynman-Kac Propagators*, preprint, arXiv:1204.2382, 2012.

[33] A. M. Stuart, *Inverse problems: a Bayesian perspective*, Acta Numer., 19 (2010), pp. 451–559.
[34] O. Talagrand and P. Courtier, *Variational assimilation of meteorological observations with the adjoint vorticity equation.* I *and* II: *Theory and numerical results*, Q. J. Roy. Meteorol. Soc., 113 (1987), pp. 1311–1347.
[35] H. Uecker, *A short ad hoc introduction to spectral methods for parabolic PDE and the Navier-Stokes equations*, lecture, Summer School Modern Computational Science, Oldenburg, Germany, 2009, pp. 169–209.
[36] P. J. van Leeuwen, *Nonlinear data assimilation in geosciences: An extremely efficient particle filter*, Q. J. Roy. Meteorol. Soc., 136 (2010), pp. 1991–1999.
[37] Y. Zhou, A. M. Johansen, and J. A. D. Aston, *Towards Automatic Model Comparison: An Adaptive Sequential Monte Carlo Approach*, preprint, arXiv:1303.3123, 2013.