Bayesian Parameter Inference for Nonlinear Stochastic Differential Equation Models

Carlo Albert

Eawag, Swiss Federal Institute of Aquatic Science and Technology, 8600 Dübendorf, Switzerland

E-mail: carlo.albert@eawag.ch

Simone Ulzega

Eawag, Swiss Federal Institute of Aquatic Science and Technology, 8600 Dübendorf, Switzerland

E-mail: simone.ulzega@eawag.ch

Abstract. Bayesian statistics has become an indispensable tool in many applied sciences for the purpose of uncertainty analysis. Inferring parametric uncertainty for stochastic differential equation (SDE) models, however, is a computationally hard problem due to the high dimensional integrals that have to be calculated. Here we present a Monte Carlo-based method for Bayesian parameter inference with one dimensional SDE models to tackle the generic problems of calibrating the model to given data time series while simultaneously quantifying the ensuing parametric uncertainty. We re-interpret the Bayesian posterior distribution for model parameters as the partition function of a statistical mechanical system and employ a Hamiltonian Monte Carlo algorithm to calculate it. Depending on the number of discretization points and the number of measurement points the dynamics of this system happen on very different time scales. Thus, we employ a multiple time scale integration together with a suitable re-parametrization to derive an efficient inference algorithm. While the algorithm is presented by means of a simple SDE model from hydrology, it is readily applicable to a wide range of inference problems. Furthermore the algorithm is highly parallelizable.

PACS numbers: 00.00, 20.00, 42.10

Keywords: Bayesian parameter inference, Hamiltonian Monte Carlo, time series analysis, stochastic differential equations

Submitted to: *New J. Phys.*

1. Introduction

Phenomenological models are employed in many applied areas of research to predict the behaviour of complex systems of all sorts. Parameters of these models often don't have a direct physical meaning and need to be inferred from measurements. In order to make reliable predictions with such models it is important to describe the dominant errors in the model and to quantify the parametric uncertainty resulting from the inference process.

Bayesian statistics describes knowledge about parameters through probability distributions and model-based learning through a consistent update rule. It is thus well suited to quantify parametric uncertainty resulting from a model based inference process. Bayesian statistics is commonly used in many applied sciences and is growing in importance in the physics community as well [1].

A faithful description of the dominant errors in a model naturally leads to *stochastic differential equations* (SDEs). The kind of problems we consider here are the calibration of ordinary 1D SDE models to noisy time series and the quantification of the resulting parametric uncertainty. Whilst the techniques we use are generic they will be presented by means of a simple yet non trivial SDE model from hydrology.

Bayesian parameter inference for SDE models is computationally very expensive, as the posterior probability density for the parameters is a *path-integral*. Problems of this kind are commonly solved by means of *Monte Carlo* methods that are based on simulating model realizations and comparing them to the data. Algorithms of this kind are *particle filters* [2]. For the case of linear SDEs that are coupled to nonlinear deterministic ordinary differential equations (ODEs), more efficient algorithms of this kind can be derived [3, 4]. The problem with these simulation-based methods is their inefficiency in the presence of many data points. One solution is to map the output space to a smaller dimensional space of *summary statistics*, and acccept/reject proposed model parameters depending on how compatible associated model runs are with the data in terms of these summary statistics. Such *Approximate Bayes Computations* [5] are relatively easy to apply as they only require us to run the simulator. However, it is largely an unsolved problem how to choose the summary statistics so as to achieve a sufficient approximation of the posterior parameter distribution.

Exact inference algorithms of high efficiency can be derived from a reinterpretation of the posterior distribution as the partition function of a statistical mechanical system and by simulating the dynamics of the latter. Such *Hamiltonian Monte Carlo* (HMC) algorithms [6] incorporate the data points for the suggestion of new parameters, whereby much higher acceptance rates are achieved. The drawback of these methods is that the model equations need to be known and derivatives have to be calculated. The latter problem, however, is largely remedied by the use of automated differentiation algorithms.

Tuning of HMC algorithms is a non-trivial matter. Two sorts of tuning parameters need to be adjusted: (i) those of the *kinetic energy* of the statistical mechanical system and (ii) those of the numerical integration scheme of Hamilton's equation in

the molecular dynamics part of the HMC algorithm. It has been shown that efficiency is gained when the kinetic term is made dependent on the configuration of the statistical mechanical system [7]. Here we explore a different, computationally simper route. Depending on the number of discretization points needed to approximate the original SDE system and the number of measurement points, the dynamics of the statistical mechanical system happen on very different time scales. Thus, we employ a multiple time scale integration technique for the simulation of the statistical mechanical system [8]. At least for 1D SDEs we always find a parametrization, which allows to partly uncouple the dominant harmonic part of the Hamiltonian in the form of harmonic oscillators and integrate them analytically.

The resulting algorithm is very efficient and, through the use of automated differentiation, easily applicable to a wide range of inference problems with SDEs. Furthermore, it is easily parallelizable.

2. An Exemplary SDE Model

The methods we present are generic and applicable to a wide range of SDE models. However, with later applications in mind and to make things concrete, we present them by means of a simple yet non-trivial model from hydrology. Consider a hydrological catchment whose dynamics at the observation time scale is well described by a linear reservoir and whose other processes happen at much shorter time scales so that they can be described by white noise. Furthermore, assume this noise to scale linearly with the system state, $S(t)$, which is the water content in the reservoir. The model equation is thus given by the SDE

$$\dot{S}(t) = r(t) - \frac{1}{K}\left(1 + \frac{\gamma}{2}\right) S(t) + \sqrt{\frac{\gamma}{K}} S(t)\eta(t)\,, \tag{1}$$

where $r(t)$ denotes the time varying rain input and $\eta(t)$ denotes white noise, i.e.,

$$\langle \eta(t)\eta(t')\rangle = \delta(t - t')\,. \tag{2}$$

Eq. (1) is to be understood in the Stratonovich sense [9]. Our parametrization is such that, for constant rain input $r(t) = r_0$ and in the long-time limit, the mean of S converges to the equilibrium solution of the unperturbed ($\gamma = 0$) system, $S_{\mathrm{eq}} = Kr_0$ (see Eq. (16) below). Scale-invariance of the model of Eq. (1) for large S leads to power law tails in the system state probability distribution (see Eq. (15)), which is in line with the observation that errors in hydrological models are often fat-tailed [10].

Conceptual models of this kind are commonly used in hydrology, for the purpose of predicting rainfall-runoff behaviour, for natural and urban catchments (see, e.g., [11]). Furthermore, by means of the transformation $S(t) = 1/n(t)$, eq. (1) turns into a model that has been suggested as a phenomenological description of the dynamics of the neutron density in nuclear reactors and extensively studied [12, 13].

Here, we consider the input $r(t)$ to be a smooth and nowhere vanishing function. Realistic multi-fractal rain inputs [14] will be studied elsewhere. Indeed, the focus of this

work is not on hydrological modeling, but on generic methods for parameter inference with SDE models that are calibrated to observed time-series. Here we assume the observed time-series, y_s, to be the outflow of the reservoir, $S(t)/K$, observed at times $0 = t_1 < t_2 < \ldots < t_{n+1} = T$, with multiplicative independent log-normal errors, i.e.,

$$\ln(y_s) = \ln\left(\frac{S(t_s)}{K}\right) + \sigma\epsilon_s\,, \quad s = 1, \ldots, n+1\,. \tag{3}$$

For simplicity, we assume σ as well as the input $r(t)$ to be known so that we are left with the task of inferring parameter combinations (K, γ) that are compatible with the data given by Eq. (3). Here, "compatible" is meant in the Bayesian sense, in which knowledge about parameters is expressed in terms of a probability distribution. In general, prior knowledge about parameters, $\boldsymbol{\theta}$, is expressed in terms of a prior probability distribution, $f_{\text{prior}}(\boldsymbol{\theta})$, while the posterior distribution, which combines prior knowledge with knowledge gained from data, is calculated by means of Bayes' formula

$$f_{\text{post}}(\boldsymbol{\theta}|\mathbf{y}) = \frac{f_{\text{prior}}(\boldsymbol{\theta})L(\mathbf{y}|\boldsymbol{\theta})}{\int f_{\text{prior}}(\boldsymbol{\theta}')L(\mathbf{y}|\boldsymbol{\theta}')d\boldsymbol{\theta}'}\,, \tag{4}$$

where $L(\mathbf{y}|\boldsymbol{\theta})$ is the so-called *likelihood function*, that is, the probability distribution for model outputs given model parameters, evaluated at the measured data. Here, we assume prior knowledge to be that parameters must be non-negative, but otherwise we assume the prior to be flat on an area large enough to encompass the area where the likelihood function is significantly different from zero. That is, the posterior is data-driven.

Before we set out to derive the likelihood function from the model equations (1) through (3) let us express the parameters and state variable with dimensionless quantities. The noise parameter γ is already dimensionless due to scale-invariance of the noise term. We replace the state variable $S(t)$ and parameter K by dimensionless quantities $q(t)$ and β, respectively, by means of the transformations

$$\beta = \sqrt{\frac{T\gamma}{K}}\,, \quad S(t) = \frac{T\gamma r(t)}{\beta^2}e^{\beta q(t)}\,. \tag{5}$$

W.r.t. these new variables and parameters, the model equation (1) becomes

$$\dot{q}(t) = \frac{\beta}{T\gamma}e^{-\beta q(t)} - \frac{1}{T}\rho(t) + \frac{1}{\sqrt{T}}\eta(t)\,, \tag{6}$$

with

$$\rho(t) = \frac{T}{\beta}\frac{\mathrm{d}}{\mathrm{dt}}\left[\ln(r(t))\right] + \frac{(2+\gamma)\beta}{2\gamma}\,. \tag{7}$$

For our algorithm it is important to have the model equation in a form where the noise term does not depend on neither the state variables nor the parameters that have to be inferred. In a one dimensional model this can always be achieved through re-parametrization.

The probability $P(q_1, T|q_0, 0)$ of finding the system in a state q_1 at time $t = T$ given that it was in an initial state q_0 at time $t = 0$, is expressed in the form of a *path-integral* as

$$P(q_1, T|q_0, 0) = \frac{1}{Z} \int e^{-\mathcal{S}[q,\dot{q}]} \delta(q(T) - q_1)\delta(q(0) - q_0)\mathcal{D}q \,, \tag{8}$$

where the integral extends over all paths $q : [0, T] \to \mathbb{R}$. The path-measure $\mathcal{D}q$ is formally written as the infinite product

$$\mathcal{D}q = \prod_t dq(t) \,. \tag{9}$$

The *action* is a functional on the space of paths and reads as [15]

$$\mathcal{S}[q, \dot{q}] = \frac{1}{T} \int_0^T dt \left\{ \frac{1}{2} \left(T\dot{q}(t) + \rho(t) - \frac{\beta}{\gamma} e^{-\beta q(t)} \right)^2 - \frac{\beta^2}{2\gamma} e^{-\beta q(t)} \right\} . \tag{10}$$

Note that the action includes the Jacobian that arises when changing coordinates from $\eta(t)$ to $q(t)$.

We introduce the time-dependent *Hamiltonian*

$$\mathcal{H}(q, t) = \frac{1}{\gamma} e^{-\beta q} + q\rho(t) \,, \tag{11}$$

and rewrite action (10) as,

$$\begin{aligned}
&\mathcal{S}[q, \dot{q}] \\
&= \frac{1}{T} \int_0^T dt \left\{ \frac{1}{2} T^2 \dot{q}^2(t) + \frac{1}{2} \left(\rho(t) - \frac{\beta}{\gamma} e^{-\beta q(t)} \right)^2 - T \frac{\partial \mathcal{H}(q,t)}{\partial t} - \frac{\beta^2}{2\gamma} e^{-\beta q(t)} \right\} \\
&+ \mathcal{H}(q(T), T) - \mathcal{H}(q(0), 0) \\
&= \frac{1}{T} \int_0^T dt \left\{ \frac{1}{2} T^2 \dot{q}^2(t) + \frac{1}{2} \left(\rho(t) - \frac{\beta}{\gamma} e^{-\beta q(t)} \right)^2 - Tq(t)\dot{\rho}(t) - \frac{\beta^2}{2\gamma} e^{-\beta q(t)} \right\} \\
&+ \frac{1}{\gamma} e^{-\beta q(T)} + q(T)\rho(T) - \frac{1}{\gamma} e^{-\beta q(0)} - q(0)\rho(0) \,.
\end{aligned} \tag{12}$$

Properties of (a transformed version of) (1) have been derived in [12, 13] and we do not repeat them here, as the focus of this paper is not on the properties of this particular model. However, in order to prove two claims made earlier, we calculate the *equilibrium distribution* $P_{\text{eq}}(q) = \lim_{T\to\infty} P(q, T|q_0, 0)$ in the simple case of a constant input $r(t) \equiv r_0$. After plugging (8) and (12) into the detailed balance condition,

$$P(q_1 t_1|q_0 t_0)P_{\text{eq}}(q_0) = P(q_0 t_1|q_1 t_0)P_{\text{eq}}(q_1) \,, \tag{13}$$

and using transformation $q(t) \to q(-t)$, which maps paths from q_0 to q_1 onto paths from q_1 to q_0, we get, since $\dot{\rho}(t) = 0$,

$$P_{\text{eq}}(q) \propto e^{-2\mathcal{H}(q)} \,. \tag{14}$$

Transforming back to the original variables, it turns out that $P_{\text{eq}}(S)$ is an inverse gamma distribution with scale parameter $2Kr_0/\gamma$ and shape parameter $(2 + \gamma)/\gamma$, i.e.,

$$P_{\text{eq}}(S) \propto S^{-2(1+\gamma)/\gamma} e^{-2Kr_0/(\gamma S)} \,, \tag{15}$$

whose mean equals the equilibrium solution of the unperturbed system ($\gamma = 0$),

$$\langle S \rangle_{\rm eq} = K r_0, \tag{16}$$

and whose variance, for $\gamma < 2$, is given by

$$\langle (S - \langle S \rangle_{\rm eq})^2 \rangle_{\rm eq} = K^2 r_0^2 \frac{\gamma}{2-\gamma} \tag{17}$$

and diverges, for $\gamma \geq 2$. The power-law decay of the inverse gamma distribution is reminiscent of the scale-invariance of the error model.

If we denote the parameter vector $\boldsymbol{\theta} = (\beta, \gamma)^T$ and assume a flat prior as described above, the posterior (4), as a function of $\boldsymbol{\theta}$, is proportional to the likelihood function, which is expressed as a path-integral as follows

$$f_{\rm post}(\boldsymbol{\theta}|\mathbf{y}) \propto \int \exp\Big[-\frac{1}{2} \sum_{s=1}^{n+1} \frac{(\ln(y_s/r(t_s)) - \beta q(t_s))^2}{\sigma^2} - \mathcal{S}[q, \dot{q}] \Big] \mathcal{D}q \,. \tag{18}$$

On the right-hand side of the above equation, the first term describes the probability distribution of model outputs, given model parameters, rain inputs and a system realization $q(t_s)$. The second term is the probability of the given system realization q and the path-integral extends over all possible system realizations. Instead of undertaking an often prohibitive numerical computation of such integral, we apply a Hamiltonian Monte Carlo method, outlined in the next section, to sample parameter vectors from a joint distribution of system realizations and model parameters given by an appropriate discretization of the action of the path-integral.

3. Inference Algorithm

In order to derive an efficient algorithm to draw parameter samples from (18) we interpret it as the partition function of a 1D statistical mechanical system and simulate the dynamics of the latter, employing the so-called *Hamiltonian Monte Carlo* (HMC) algorithm [6]. The model parameters $\boldsymbol{\theta}$ are interpreted as additional dynamical degrees of freedom coupling to the system variables $q(t)$. Each degree of freedom, $q(t)$ and $\boldsymbol{\theta}$ in our case, is paired with a conjugate variable, $p(t)$ and $\boldsymbol{\pi}$, respectively, and the system is defined by the Hamiltonian

$$\mathcal{H}_{\rm HMC}(q, \boldsymbol{\theta}; p, \boldsymbol{\pi}) = K(p, \boldsymbol{\pi}) + V(q, \boldsymbol{\theta}) \,, \tag{19}$$

where

$$K(p, \boldsymbol{\pi}) = \int_0^T \frac{p^2(t)}{2m(t)} dt + \sum_{\alpha=1}^{2} \frac{\pi_\alpha^2}{2m_\alpha} \,, \tag{20}$$

and $V(q, \boldsymbol{\theta})$ is the negative logarithm of the kernel of (18). The posterior (18) can then be expressed by the phase space path integral

$$f_{\rm post}(\boldsymbol{\theta}|\mathbf{y}) \propto \int e^{-\mathcal{H}_{\rm HMC}(q, \boldsymbol{\theta}; p, \boldsymbol{\pi})} \mathcal{D}p \mathcal{D}q d\boldsymbol{\pi} \,. \tag{21}$$

The HMC method, which is a combination of the *Metropolis algorithm* [16] and *molecular dynamics* methods [17, 18], iterates the following steps:

(i) Momenta p and π are sampled from the Gaussian distributions defined by Eq. (20).

(ii) The system is then allowed to evolve in $(q, \boldsymbol{\theta}; p, \boldsymbol{\pi})$-phase space for an arbitrary time interval τ according to a volume-preserving and time-reversible solution of a discretization of Hamilton's equations.

(iii) The discretization error on the energy preservation due to the previous step is corrected by a Metropolis acceptance/rejection step.

The last step is the standard Metropolis algorithm, while the first two steps allow us to make arbitrarily large jumps in phase space while maintaining an arbitrarily large acceptance rate. Each new phase space configuration corresponds to a well-defined combination of model parameters $\boldsymbol{\theta}$. Thus, the simulated succession of configurations represent a Markov chain of sampled parameter vectors that naturally leads to the sought probability distribution for the system parameters.

In order to simulate the dynamics of the Hamiltonian (19), we first need to discretize the path-integral (21). Therefore, let us assume that the measurement time points $\{y_s\}_{s=1,\ldots,n+1}$ of the time series (3) are equidistantly distributed on the time interval $[0, T]$, with $t_1 = 0$ and $t_{n+1} = T$. Each interval between two consecutive data points is further partitioned into j bins, such that we have a total of $nj + 1 = N >> 1$ discretization points. The path-integral (21) is then approximated by an ordinary integral, with the approximate path-measure

$$\mathcal{D}p\mathcal{D}q \approx \prod_i dp_i dq_i \tag{22}$$

and the discretized versions of $K(p, \boldsymbol{\pi})$ and $V(q, \boldsymbol{\theta})$

$$K(p, \boldsymbol{\pi}) \approx \sum_{i=1}^{N} \frac{p_i^2}{2m_q}\Delta t + \sum_{\alpha=1}^{2} \frac{\pi_\alpha^2}{2m_\alpha}\,, \tag{23}$$

$$\begin{aligned} V(q, \boldsymbol{\theta}) \approx{}& \frac{\Delta t}{T}\sum_{i=2}^{N}\left\{\frac{1}{2}T^2\dot{q}_i^2 + \frac{1}{2}\left(\rho_i - \frac{\beta}{\gamma}e^{-\beta q_i}\right)^2 - \frac{\beta^2}{2\gamma}e^{-\beta q_i} - Tq_i\dot{\rho}_i\right\} \\ &+\frac{1}{\gamma}e^{-\beta q_N} + q_N\rho_N - \frac{1}{\gamma}e^{-\beta q_1} - q_1\rho_1 \\ &+\sum_{s=1}^{n+1}\frac{(\ln(y_s/r_{(s-1)j+1}) - \beta q_{(s-1)j+1})^2}{2\sigma^2}\,, \end{aligned} \tag{24}$$

with

$$\dot{q}_i = \frac{q_i - q_{i-1}}{\Delta t}\,, \tag{25}$$

and

$$\rho_i = \frac{T}{\beta}\frac{\ln(r(t_i)/r(t_{i-1}))}{\Delta t} + \frac{(2+\gamma)\beta}{2\gamma}\,, \quad \dot{\rho}_i = \frac{\rho_i - \rho_{i-1}}{\Delta t}\,. \tag{26}$$

Note that we've neglected terms of order $\mathcal{O}(N^{-1/2})$ in the action (24).

The discretized Hamiltonian describes a *classical polymer chain* of N beads with harmonic bonds between neighbouring beads in an external field. The latter consists

of two parts, a field that results from the dynamics of the original equations (1) and is felt by all the beads, and a field that results from the measurements and is felt by the measurement beads only. The masses m_q and m_α are tunable parameters of the algorithm. A good strategy consists in assigning larger weights to the beads associated with measurement points than to the intermediate beads. The dynamics of the system will be therefore mostly dependent on the lighter intermediate particles, while the heavy measurement beads will be constrained in the vicinity of the measured data points. We thus reduce the original Bayesian inference problem to simulating the dynamics of a linear polymer, as sketched in Fig. (1). In this context the physical time (horizontal axis) is interpreted as a spatial dimension while a fictitious simulation time is introduced (vertical axis). Each *state* of this fictitious molecule corresponds to a well-defined configuration in the original phase space, characterized by a set of system variables $\{q_i\}_{i=1,\dots,N}$ and a parameter vector $\boldsymbol{\theta}$.

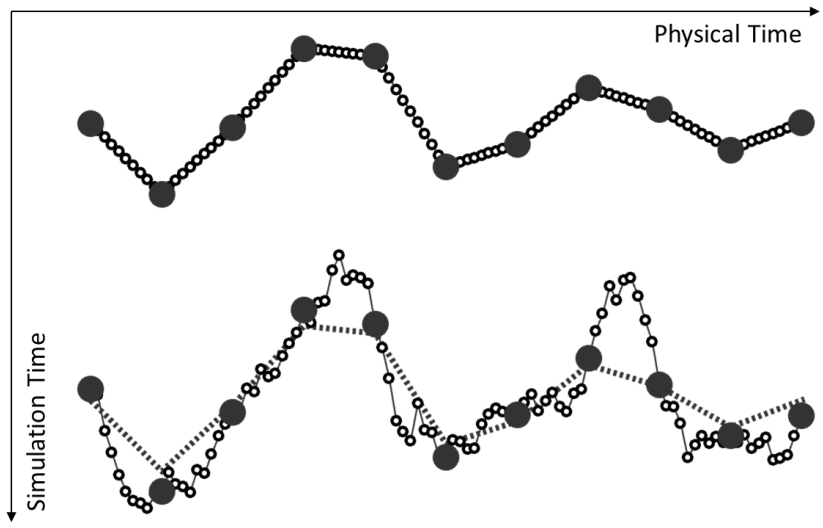

Figure 1: Simulated dynamics of a polymer chain with $n+1 = 11$ data points (large filled circles) and $j - 1 = 9$ intermediate beads (small empty circles). All other parameters are discussed in Section 4. Top: initial state. The intermediate beads represent a linear interpolation to the data points. Bottom: polymer configuration after 1000 iterations of the propagation algorithm. The dotted line represents the initial configuration. It is evident that the new configuration is mostly determined by the dynamics of the low-mass intermediate beads, while the heavy-mass data points barely move from their initial positions.

The potential (24) contains terms with different scaling in the potentially large numbers N and n, which describe dynamics on different time-scales. In particular, for large N, $V(q,\boldsymbol{\theta})$ is dominated by its harmonic part and a brute force numerical integration of Hamilton's equations in step (ii) of the HMC algorithm would require a very small discretization time-step to sufficiently resolve its dynamics. An interesting approximative approach would be to employ a partial averaging of the fast Fourier

modes as described in [19]. We choose an exact approach and employ a multiple time scale integration based on Trotter's formula [8].

For this purpose it proves to be useful to introduce so-called *staging* variables [20], and diagonalize the harmonic part in between the measurement points. Therefore, we rewrite the discretized harmonic part of the action as

$$\begin{aligned}&\sum_{i=2}^{N}\frac{T}{2\Delta t}(q_i-q_{i-1})^2\\&=\frac{T}{2}\sum_{s=1}^{n}\left\{\frac{(q_{(s-1)j+1}-q_{sj+1})^2}{j\Delta t}+\sum_{k=2}^{j}\frac{k}{(k-1)\Delta t}(q_{(s-1)j+k}-q^*_{(s-1)j+k})^2\right\},\end{aligned}\tag{27}$$

with

$$q^*_{(s-1)j+k}=\frac{(k-1)q_{(s-1)j+k+1}+q_{(s-1)j+1}}{k}\,.\tag{28}$$

We apply the coordinate transformations

$$u_{sj+1}=q_{sj+1}\,,\qquad\qquad s=0,\ldots,n\,,\tag{29}$$

for the boundary beads, corresponding to the original measurement points, and

$$u_{sj+k}=q_{sj+k}-q^*_{sj+k}\,,\quad s=0,\ldots,n-1\,,\quad k=2,\ldots,j\,,\tag{30}$$

for the intermediate staging beads. The inverse transformations are given by

$$q_{sj+1}=u_{sj+1}\,,\tag{31}$$

$$q_{sj+k}=\sum_{l=k}^{j+1}\frac{k-1}{l-1}u_{sj+l}+\frac{j-k+1}{j}u_{sj+1}\,.\tag{32}$$

The latter can be equivalently expressed by the recursive relation

$$q_{sj+k}=u_{sj+k}+\frac{k-1}{k}q_{sj+k+1}+\frac{1}{k}u_{sj+1}\,.\tag{33}$$

We split the Hamiltonian $\mathcal{H}_{\text{HMC}}$ into components with different scaling behaviour in the potentially large numbers n and N and write

$$\mathcal{H}_{\text{HMC}}=\mathcal{H}_N+\mathcal{H}_n+\mathcal{H}_1\,,\tag{34}$$

where

$$\mathcal{H}_N=\frac{1}{2}\sum_{s=1}^{n}\sum_{k=2}^{j}\left\{\frac{\Delta t}{m_q}p^2_{(s-1)j+k}+\frac{Tk}{\Delta t(k-1)}u^2_{(s-1)j+k}\right\},\tag{35}$$

$$\begin{aligned}\mathcal{H}_n=&\frac{1}{2}\sum_{s=1}^{n+1}\left\{\frac{\Delta t}{m_q}p^2_{(s-1)j+1}+\frac{(\ln(y_s/r_{(s-1)j+1})-\beta u_{(s-1)j+1})^2}{\sigma^2}\right\}\\&+\frac{T}{2j\Delta t}\sum_{s=1}^{n}(u_{(s-1)j+1}-u_{sj+1})^2\,,\end{aligned}\tag{36}$$

$$\begin{aligned}\mathcal{H}_1=&\sum_{\alpha=1}^{2}\frac{\pi^2_\alpha}{2m_\alpha}+\frac{\Delta t}{T}\sum_{i=2}^{N}\left\{\frac{1}{2}\left(\rho_i-\frac{\beta}{\gamma}e^{-\beta q_i}\right)^2-\frac{\beta^2}{2\gamma}e^{-\beta q_i}-Tq_i\dot{\rho}_i\right\}\\&+\frac{1}{\gamma}e^{-\beta q_N}+q_N\rho_N-\frac{1}{\gamma}e^{-\beta q_1}-q_1\rho_1\,.\end{aligned}\tag{37}$$

The harmonic part for the staging beads given in Eq. (35) scales linearly with N. The terms of Eq. (36), including both the harmonic part for the boundary beads and the measurement term, scale linearly with n. Finally, Eq. (37) does not scale with neither n nor N. Thanks to the staging variables $\mathcal{H}_N$ decouples completely from $\mathcal{H}_n$.

We use the Trotter formula according to [20] in order to design a reversible molecular dynamics integrator that takes these different time scales into account. In order to design an appropriate partition of the Hamiltonian, we need to distinguish different regimes, such as

i. $\mathcal{H}_N \sim \mathcal{H}_n >> \mathcal{H}_1$,

ii. $\mathcal{H}_N >> \mathcal{H}_n \sim \mathcal{H}_1$,

iii. $\mathcal{H}_N >> \mathcal{H}_n >> \mathcal{H}_1$.

Here, we restrict ourselves to regime (*ii*), i.e., we assume that the number of measurements n is small and/or the measurement error σ is large. The generalization of the method to the other schemes is straightforward. In regime (*ii*) we simply separate the harmonic part of the action, for the staging beads, from the rest and write

$$\mathcal{H}_{\text{HMC}} = \mathcal{H}_N + \mathcal{H}' \,. \tag{38}$$

In order to design a reversible integrator for the associated Hamilton equations, we define the Liouville operators

$$iL_N = \{\cdot\,,\, \mathcal{H}_N\}\,, \quad iL' = \{\cdot\,,\, \mathcal{H}'\}\,, \tag{39}$$

where $\{\cdot\,,\cdot\}$ denotes the Poisson brackets that are defined on functions on the phase space. Trotter's formula [21] allows us to write the Hamiltonian propagator as

$$e^{i(L_N+L')\tau} = (e^{iL_N(\Delta\tau/2)} e^{iL'\Delta\tau} e^{iL_N(\Delta\tau/2)})^P + \mathcal{O}(\tau^3/P^2)\,, \tag{40}$$

for $\tau = P\Delta\tau$. In regime (*ii*) the outer propagator $\exp[iL_N(\Delta\tau/2)]$ describes much faster dynamics than the inner one. However, it is the dynamics of uncoupled harmonic oscillators, which we can readily solve. Masses and frequencies of the oscillators are derived from (35) as

$$m = m_q/\Delta t\,, \quad \omega_k = \sqrt{\frac{Nk}{(k-1)m}}\,. \tag{41}$$

The fast outer propagator is then given by the equations,

$$\begin{aligned} &u_{(s-1)j+k}(\Delta\tau/2) \\ &= u_{(s-1)j+k}(0)\cos(\omega_k\Delta\tau/2) + \frac{p_{(s-1)j+k}(0)}{m\omega_k}\sin(\omega_k\Delta\tau/2)\,, \end{aligned} \tag{42}$$

$$\begin{aligned} &p_{(s-1)j+k}(\Delta\tau/2) \\ &= p_{(s-1)j+k}(0)\cos(\omega_k\Delta\tau/2) - m\omega_k u_{(s-1)j+k}(0)\sin(\omega_k\Delta\tau/2)\,, \end{aligned} \tag{43}$$

for $s = 1, \ldots, n$ and $k = 2, \ldots, j$.

For the inner, slow propagator, we employ the velocity Verlet algorithm [22]. For the boundary beads, it reads

$$\begin{aligned} & u_{(s-1)j+1}(\Delta\tau) \\ & = u_{(s-1)j+1}(0) + \frac{\Delta\tau}{m} p_{(s-1)j+1}(0) + \frac{\Delta\tau^2}{2m} F_{(s-1)j+1}[\mathbf{u}(0), \boldsymbol{\theta}(0)] \,, \end{aligned} \tag{44}$$

$$\begin{aligned} & p_{(s-1)j+1}(\Delta\tau) \\ & = p_{(s-1)j+1}(0) + \frac{\Delta\tau}{2} (F_{(s-1)j+1}[\mathbf{u}(0), \boldsymbol{\theta}(0)] + F_{(s-1)j+1}[\mathbf{u}(\Delta\tau), \boldsymbol{\theta}(\Delta\tau)]) \,, \end{aligned} \tag{45}$$

with $s = 1, \ldots, n+1$ and where $F_i[\mathbf{u}, \boldsymbol{\theta}]$ denotes the partial derivative of $\mathcal{H}'[\mathbf{u}, \boldsymbol{\theta}]$ w.r.t. u_i. For the model parameters $\boldsymbol{\theta}$ and their momenta $\boldsymbol{\pi}$, analogous equations apply. When it comes to the staging beads, however, only the momenta have to be updated, because the associated kinetic term is not part of $\mathcal{H}'$ but of $\mathcal{H}_N$. Thus,

$$\begin{aligned} & p_{(s-1)j+k}(\Delta\tau) \\ & = p_{(s-1)j+k}(0) + \frac{\Delta\tau}{2} (F_{(s-1)j+k}[\mathbf{u}(0), \boldsymbol{\theta}(0)] + F_{(s-1)j+k}[\mathbf{u}(\Delta\tau), \boldsymbol{\theta}(\Delta\tau)]) \,, \end{aligned} \tag{46}$$

with $s = 1, \ldots, n$ and $k = 2, \ldots, j$. The above propagators are applied sequentially P times to calculate the system evolution over the time τ.

4. Numerical Results

The algorithm described above was implemented using the free and open-source programming language *Julia* (version 0.3.9) [23]. One of the attractive features of the language is its (fast growing) library of over 600 packages for numerical and scientific applications. In our case, we made extensive use of the *ReverseDiffSource* package, which provides a powerful tool for reverse-mode automated differentiation (AD) of arbitrarily complex user-defined expressions. Our algorithm benefits greatly from the use of AD. Indeed, it gives us the possibility to modify the model of Eq. (1) and therefore the action (12) leaving the implementation of the algorithm unaltered. This clearly gives our program significant flexibility making it suitable for a much broader range of applications than the simple exemplary SDE model described here.

We have run preliminary simulations on a 64-bit laptop computer equipped with a Windows 7 operating system, a double-core 1.6 GHz CPU (Intel i5-4200U) and 8 GB of RAM. We have initially used a serial implementation of the algorithm. We assumed $n + 1 = 11$ measurement points and $j = 10, 20, \ldots, 50$ intermediate staging beads, with a total number of discretization points $N = 101, 201, \ldots, 501$. We have set $\Delta\tau = 0.25$ (arbitrary units of time) and $P = 3$ in the Hamiltonian propagator (40), with a constant total observation time $T = 833$ (arbitrary units of time). The "true" values of the parameters to be inferred were set to $K_{\text{true}} = 50$ (arbitrary units of time) and $\gamma_{\text{true}} = 0.2$. Their initial values in the simulations were set to $K = 200$ (arbitrary units of time) and $\gamma = 0.5$.

We have assumed a simple sinusoidal input $r(t) = \sin^2(0.01t) + 0.1$. A system realization was first obtained from Eq. (1) using K_{true} and γ_{true}. Such system

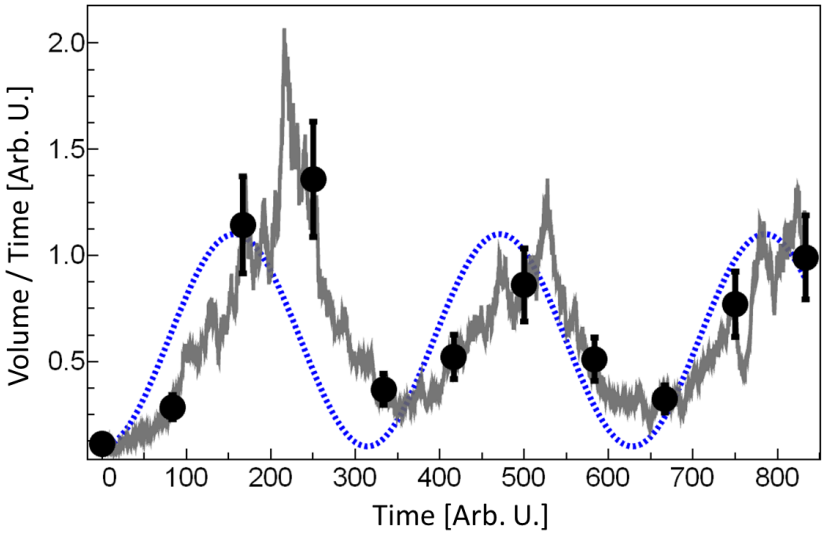

Figure 2: Sinusoidal input $r(t)$ (dashed line), "true" system realization with parameters $K_{\text{true}} = 50$ (in some arbitrary units of time) and $\gamma_{\text{true}} = 0.2$ (solid line) and synthetic observed data. One may notice how the system response follows the oscillations of the input signal.

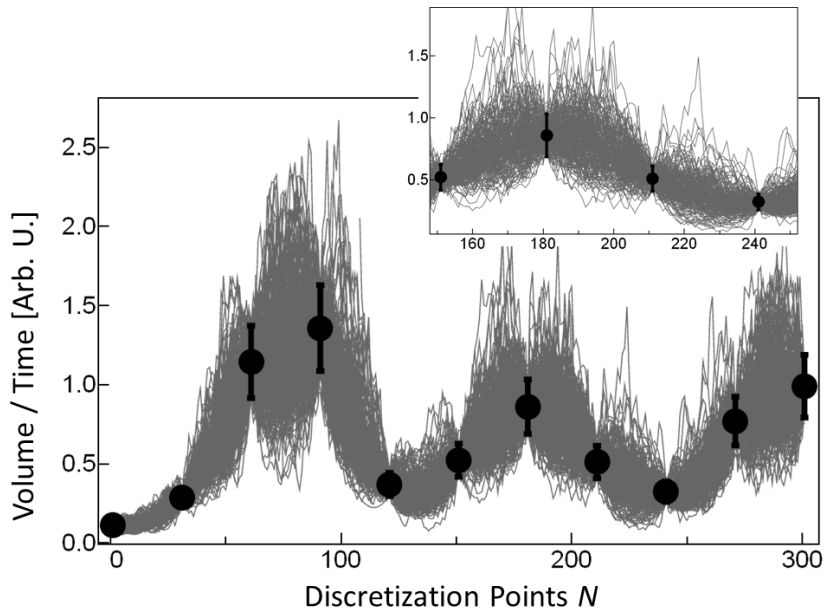

Figure 3: Simulated system realizations and corresponding synthetic data generated as described in the text with $j = 30$ staging beads and therefore $N = 301$ discretization points. In the inset one may appreciate the different dynamics of heavy data points and light staging beads.

realization was then used to generate a synthetic time series of observed data according to Eq. (3). The error σ was set to 0.1. The input signal, the "true" system realization

and the corresponding data time series are shown in Fig. (2)

Within the investigated range of parameters we observed a reasonably linear dependence of the total computing time on the number of discretization points N. For example, with $j = 30$ (i.e., $N = 301$ discretization points) a complete run with 20000 iterations required about 13 minutes. A set of 200 simulated system realizations with $j = 30$ is shown in Fig. (3) together with the synthetic data generated as described above. In the simulation the masses were set (in arbitrary units) to 720 for the boundary beads, 130 for the staging beads and 150 for both the dimensionless parameters β and γ. The different dynamics of the heavy boundary beads and the light staging beads can be appreciated in Fig. (3).

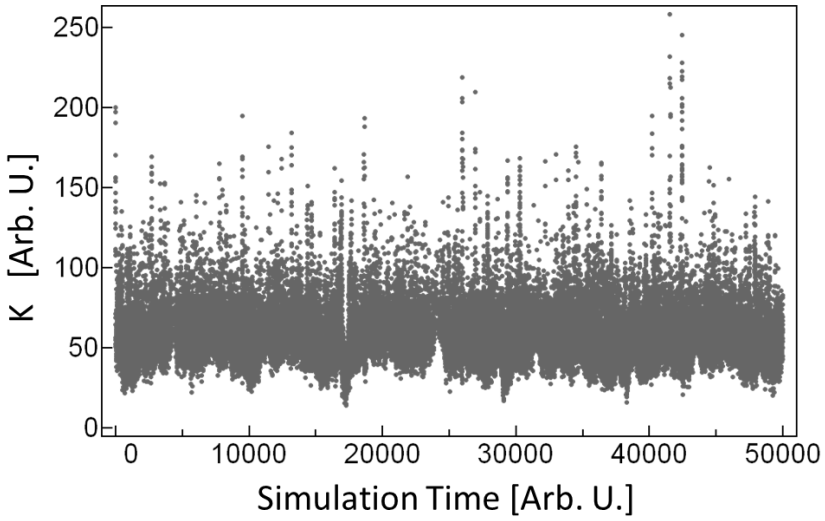

Figure 4: Markov chain for the inferred parameter K.

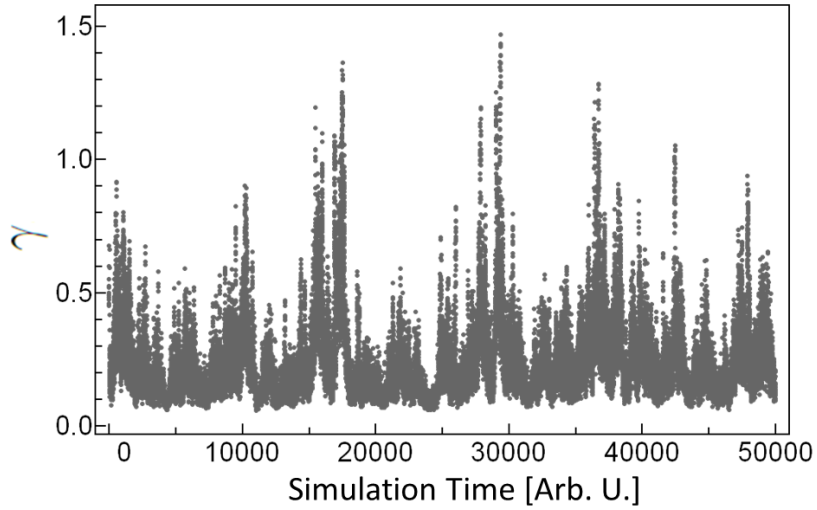

Figure 5: Markov chain for the inferred parameter γ.

The Markov chains for the parameters K and γ generated after 50000 iterations

of the HMC algorithm are shown in Figs. (4) and (5), respectively. Both of them are clearly compatible with the true parameter values $K_{\text{true}} = 50$ (arbitrary units of time) and $\gamma_{\text{true}} = 0.2$. The efficiency of the algorithm can be best appreciated by inspecting the system evolution in the phase space $K - \gamma$, as shown in Fig. (6). The very first step of the algorithm takes already the system in the vicinity of the true parameter values, where most of its dynamics then occur. A few excursions away from such values are immediately compensated by the algorithm.

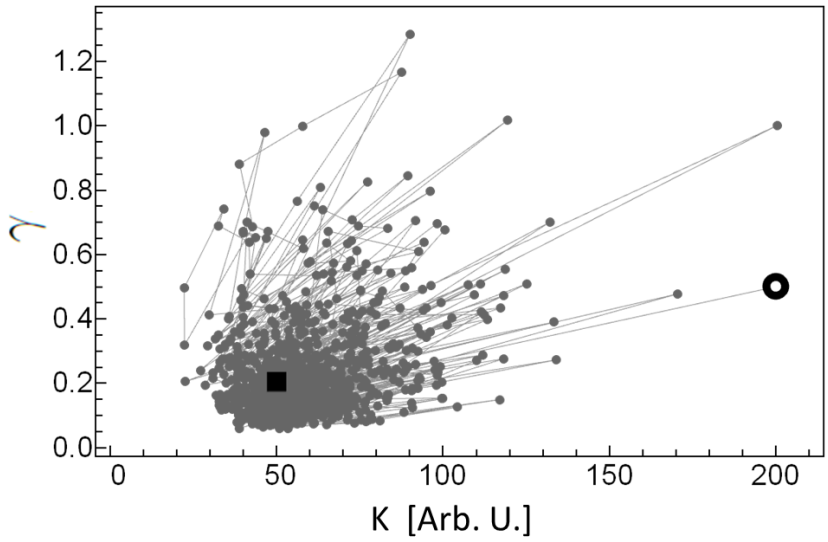

Figure 6: System dynamics in the phase space $K - \gamma$. The empty circle represents the initial state, while the square corresponds to the true parameter values used to generate the data.

The Markov chains of Figs. (4) and (5) naturally lead to the probability density functions (PDF) for the two parameters of interest, K and γ, respectively. These functions were calculated using the built-in kernel density estimator provided by Mathematica (version 10) and are shown in Fig. (7).

The structure of the HMC algorithm is well suited to be parallelized. We tested the built-in Julia multiprocessing environment to run the program on two separate processes, but we observed a much slower performance compared to the serial implementation. This should not be surprising since with the small number of discretization points used so far, not more than a few hundreds, the overheads associated with the communication between the parallel processes significantly exceeded their advantages. Nevertheless, the method might be applied to systems requiring a much larger number of points. In hydrology, for instance, typical time series corresponding to hourly observations of precipitations and river discharges may contain several tens of thousands of measurements. In such cases a parallel version of the program would definitely improve the performance of the method. However, it is interesting to observe that Julia might not be the best option for parallelization. Indeed, the multiprocessing environment provided by Julia is strongly task oriented, while our

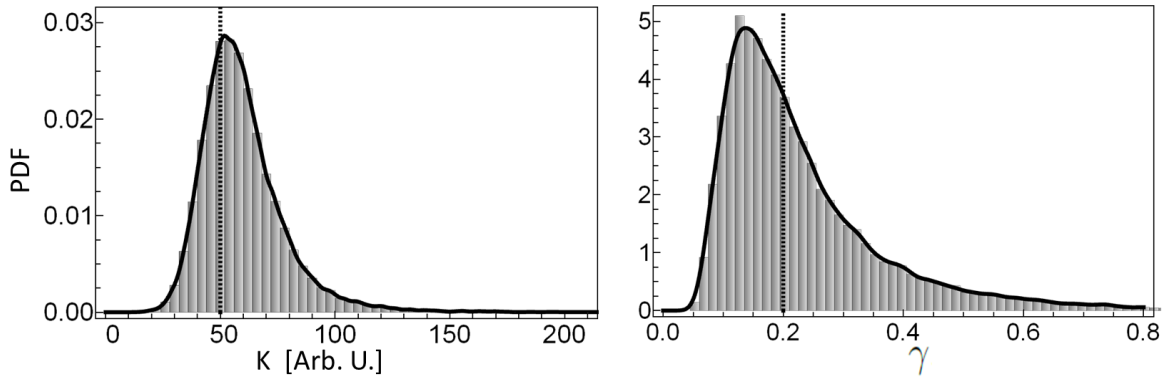

Figure 7: Probability density functions for the inferred parameters K (left) and γ (right). The true values used to generate the data, $K_{\text{true}} = 50$ and $\gamma_{\text{true}} = 0.2$, are represented by the dotted vertical lines.

problem is fundamentally data oriented. In view of a future parallelization of the HMC algorithm it is therefore advisable to investigate also alternative programming languages, such as C++. This issue will be discussed in a forthcoming publication.

5. Conclusions

We've presented a new algorithm, for the generation of posterior parameter samples, for ordinary 1D SDE models that are calibrated to time-series. The algorithm is derived from a re-interpretation of the posterior distribution as a partition function of a 1D statistical mechanical system and employs a Hamiltonian Monte Carlo approach combined with a multiple time-scale integration. Furthermore, a generic re-parametrization is suggested, which decouples harmonic modes in between measurement points from both the measurement potential and the model parameters, and allows for an efficient analytical integration of these modes. The algorithm can be implemented in an efficient and generic fashion using automated differentiation and parallelization. Furthermore, it can easily be adapted to other inference problems. In particular, it can be adapted to higher dimensional SDEs and SDEs coupled to ODEs. We explore these adaptations in our future work.

References

[1] U. von Toussaint. Bayesian inference in physics. *Rev. Mod. Phys.*, 83(3):943, 2011.

[2] N. Chopin, P. E. Jacob, and O. Papaspiliopoulos. Smc2: an efficient algorithm for sequential analysis of state space models. *J. Roy. Stat. Soc. B*, 75(3):397–426, 2013.

[3] L. Tomassini, P. Reichert, H. R. Künsch, C. Buser, R. Knutti, and M. E. Borsuk. A smoothing algorithm for estimating stochastic, continuous time model parameters and its application to a simple climate model. *J. Roy. Stat. Soc. C*, 58(5):679–704, 2009.

[4] P. Reichert and J. Mieleitner. Analyzing input and structural uncertainty of nonlinear dynamic models with stochastic, time-dependent parameters. *Water Resources Res.*, 45, 2009.

[5] C. Albert, H. R. Künsch, and A. Scheidegger. A simulated annealing approach to approximate bayes computations. *Stat. Comput.*, pages 1–16, 2014.

[6] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Phys. Lett. B*, 195(2):216–222, 1987.

[7] M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J. Roy. Stat. Soc. B*, 73(Part 2):123–214, 2011.

[8] M. E. Tuckerman, B. J. Berne, G. J. Martyna, and M. L. Klein. Efficient molecular dynamics and hybrid monte carlo algorithms for path integrals. *J. Chem. Phys.*, 99(4):2796–2808, 1993.

[9] R. L. Stratonovich. *Conditional Markov Processes and Their Application to the Theory of Optimal Control*. Elsevier, New York, 1968.

[10] M. Thyer, B. Renard, D. Kavetski, G. Kuczera, S. W. Franks, and S. Srikanthan. Critical evaluation of parameter consistency and predictive uncertainty in hydrological modeling: A case study using bayesian total error analysis. *Water Resources Res.*, 45(12), 2009.

[11] A. Breinholt, F. O. Thordarson, J K. Moller, M. Grum, P. S. Mikkelsen, and H. Madsen. Grey-box modelling of flow in sewer systems with state-dependent diffusion. *Environmetrics*, 22(8):946–961, 2011.

[12] W. L. Dutré and A. F. Debosscher. Exact Statistical Analysis of Nonlinear Dynamic Nuclear-Power Reactor Models by the Fokker-Planck Method Part I: Reactor with Direct Power Feedback. *Nuclear Science and Engineering*, 62(3):355–363, 1977.

[13] H. Fujisaka, H. Ishii, M. Inoue, and T. Yamada. Intermittency caused by chaotic modulation. iilyapunov exponent, fractal structure and power spectrum. *Progress of theoretical physics*, 76(6):1198–1209, 1986.

[14] Y. Tessier, S. Lovejoy, P. Hubert, D. Schertzer, and S. Pecknold. Multifractal analysis and modeling of rainfall and river flows and scaling, causal transfer functions. *J. Geophys. Res.*, 101(D21):26427–26440, 1996.

[15] A. W. C. Lau and T. C. Lubensky. State-dependent diffusion: thermodynamic consistency and its path integral formulation. *Phys. Rev. E*, 76(1):011123, 2007.

[16] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092, 1953.

[17] B. J. Alder and T. E. Wainwright. Studies in Molecular Dynamics. I. General Method. *J. Chem. Phys.*, 31:459–466, 1959.

[18] A. Rahman. Correlations in the motion of atoms in liquid argon. *Physical Review*, 136:405–411, 1964.

[19] J. D. Doll, R. D. Coalson, and D. L. Freeman. Fourier path-integral monte carlo methods: Partial averaging. *Phys. Rev. Lett.*, 55(1):1, 1985.

[20] M. Tuckerman, B. J. Berne, and G. J. Martyna. Reversible multiple time scale molecular dynamics. *J. Chem. Phys.*, 97(3):1990–2001, 1992.

[21] H. F. Trotter. On the product of semi-groups of operators. *Proc. Amer. Math. Soc.*, 10(4):545–551, 1959.

[22] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *J. Chem. Phys.*, 76(1):637–649, 1982.

[23] http://julialang.org/.