Approximate Bayesian Computation for Granular and Molecular Dynamics Simulations

Lina Kulakova
Computational Science and Engineering Laboratory
ETH Zürich
Clausiusstrasse 33
CH-8092, Switzerland
lina.kulakova@mavt.ethz.ch

Panagiotis Angelikopoulos
Computational Science and Engineering Laboratory
ETH Zürich
Clausiusstrasse 33
CH-8092, Switzerland
pangelik@inf.ethz.ch

Panagiotis E. Hadjidoukas
Computational Science and Engineering Laboratory
ETH Zürich
Clausiusstrasse 33
CH-8092, Switzerland
phadjido@mavt.ethz.ch

Costas Papadimitriou
Department of Mechanical Engineering
University of Thessaly
Pedion Areos
GR-38334 Volos, Greece
costasp@uth.gr

Petros Koumoutsakos*
Computational Science and Engineering Laboratory
ETH Zürich
Clausiusstrasse 33
CH-8092, Switzerland
petros@ethz.ch

ABSTRACT

The effective integration of models with data through Bayesian uncertainty quantification hinges on the formulation of a suitable likelihood function. In many cases such a likelihood may not be readily available or it may be difficult to compute. The Approximate Bayesian Computation (ABC) proposes the formulation of a likelihood function through the comparison between low dimensional summary statistics of the model predictions and corresponding statistics on the data. In this work we report a computationally efficient approach to the Bayesian updating of Molecular Dynamics (MD) models through ABC using a variant of the Subset Simulation method. We demonstrate that ABC can also be used for Bayesian updating of models with an explicitly defined likelihood function, and compare ABC-SubSim implementation and efficiency with the transitional Markov chain Monte Carlo (TMCMC). ABC-SubSim is then used in force-field identification of MD simulations. Furthermore, we examine the concept of relative entropy minimization for the calibration of force fields and exploit it within ABC. Using different approximate posterior formulations, we showcase that assuming Gaussian ensemble fluctuations of molecular systems quantities of interest can potentially lead to erroneous parameter identification.

Keywords

Uncertainty Quantification, Approximate Bayesian Computation, Subset Simulation, Molecular Dynamics, High Performance Computing

1. INTRODUCTION

Bayesian inference is amongst the prevalent methods for predictive engineering simulations. In the context of uncertainty quantification, Bayesian statistics quantify the predictive envelope of computational models in light of experimental measurements. Bayesian methods utilize at their core the likelihood and the prior probability distribution functions (PDF). While the prior distribution function encompasses all physical limitations and expert knowledge, it is the likelihood PDF that quantifies the plausibility of a stochastic model class to predict experimental measurements. In many cases however, the likelihood function can be theoretically or computationally intractable, as in the case of stochastic reaction networks [24]. To remedy this, the Approximate Bayesian Computation (ABC) [24, 29] framework was introduced as a likelihood-free variant to the classical Bayesian inference methodology. Examples of its widespread usage include stochastic population models in ecology [33, 37, 5], biology [39, 30] and chemistry [21].

The inference performed in ABC relaxes the need to formulate a likelihood for a probabilistic prediction error model [6], so that one can use summary statistics of the Quantities of Interest (QoIs) to calibrate the model parameters. This is achieved as the likelihood is replaced with a discrepancy measure that compares predictions of a stochastic model class to observed data [38].

A wide variety of stochastic methods exist for the posterior inference in ABC. A non-exhaustive list can be found in [5, 9, 36]. These methods involve numerous evaluations of the model of interest. In order to deal with complex posterior PDF supports and peaked distributions, an extension

*Corresponding author at: Computational Science and Engineering Laboratory, ETH Zürich, Clausiusstrasse 33, CH-8092, Switzerland, E-mail address: petros@ethz.ch

PASC '16, June 08–10, 2016, Lausanne, Switzerland
© 2016 ACM. ISBN 978-1-4503-4126-4/16/06...$15.00
DOI: http://dx.doi.org/10.1145/2929908.2929918

has been proposed to the established algorithm Subset Simulation (SubSim) [4] used for estimating rare events. The new algorithm, named ABC-SubSim, can be used for posterior sampling within the ABC framework [9] and allows for perfect sampling parallelization on high performance computing architectures.

A stochastic model class where the application of ABC is of specific interest is Molecular Dynamics (MD). MD is a popular method to investigate properties and hypotheses in atomistic scales of materials. MD simulations integrate Newton's equation of motion for interacting atoms, and use the particle trajectories and forces to estimate quantities of interest in the form of moments of statistical distribution averaged over time and number of atoms. Before MD simulations can be used for predicting any QoIs, one must assign a functional form and parameter values describing the interatomic interactions. These interactions are commonly referred to as a Force-Field (FF).

Several recent works examined the usage of Bayesian inference in MD-FF calibration [1, 2, 8, 7, 31, 23, 13]. In these works, the underlying assumption during the Bayesian probabilistic error treatment relies on the Gaussianity of the fluctuations of the target properties (diffusion coefficients, radial distribution functions, etc.) [1, 13]. However, it is often the case in MD that some thermodynamical properties of interest do not exhibit a Gaussian ensemble. Shell [34] states that the most relevant target when calibrating molecular models is the relative entropy between the two ensembles. In the case of Gaussian distributed target QoIs ensembles, the relative entropy relaxes to accurate estimation of means and variances. However, things get considerably more complex for non-Gaussian distributions.

In this paper we discuss the usage of ABC-SubSim for Bayesian model updating. We present a state-of-the-art High Performance Computing (HPC) implementation for ABC-SubsSim in heterogeneous computer architectures and include it into our large scale parallel sampling framework Π4U [18]. We compare its algorithmic and implementation differences with the Transitional Markov Chain Monte Carlo method (TMCMC) [10] already available in Π4U. We extend the implementation of ABC-SubSim to cover the explicit likelihood-driven Bayesian inference case which allows us to compare the results of ABC-SubSim and TMCMC on a deterministic problem with a complex support posterior PDF. Finally, we employ ABC-SubSim to explore the calibration of MD simulations using three different summary statistics: 1) the first two moments, 2) the four chosen quantiles, and 3) the relative entropy. We show that using ABC with the first two moments as summary statistics fails to adequately infer MD model parameters when the data represents a thermodynamical quantity with non-Gaussian fluctuations.

This paper is structured as follows: Section 2 provides the mathematical formulation of ABC and ABC-SubSim, for likelihood-driven and likelihood-free Bayesian inference for stochastic and deterministic models. Section 3 discusses the similarities and parallel implementation issues of TMCMC and ABC-SubSim which allow us to incorporate ABC-SubSim in the general purpose Π4U software to extend its use to likelihood-free Bayesian inference problems. Section 4 compares ABC-SubSim and TMCMC making use of the likelihood function for a complex posterior distribution arising in discrete element method [22] with deterministic model predictions. Section 5 illustrates the importance of applying likelihood-free model inference to calibrate MD models in which the MD outcomes are inherently stochastic and non-Gaussian. The use of ABC-SubSim within the HPC Π4U framework reduces the excessive computations to manageable levels using parallel computing architectures.

2. APPROXIMATE BAYESIAN COMPUTATION (ABC)

Here and further in the paper we assume that all the probability distributions have PDFs. We denote a probability by $P(\cdot)$ and a PDF by $p(\cdot)$.

Bayesian inference involves updating the prior information about the parameters θ of the stochastic forward model $p(y|\theta, \mathcal{M})$ of the model class $\mathcal{M}$, where y is the forward model prediction. It starts by imposing a prior PDF $p(\theta|\mathcal{M})$ over the model parameters θ. Then, the likelihood $p(D|\theta, \mathcal{M})$ is formulated by replacing the prediction y in the forward model with the data D. Finally, Bayes' Theorem is used to get the posterior distribution over the parameters θ given the data D:

$$p(\theta|D, \mathcal{M}) = \frac{p(\theta|\mathcal{M})\; p(D|\theta, \mathcal{M})}{p(D|\mathcal{M})}. \tag{1}$$

In the rest of the paper the conditioning on $\mathcal{M}$ is omitted for simplicity.

2.1 Likelihood-Free Bayesian Inference

For certain model classes the likelihood $p(D|\theta)$ can be intractable. In such cases the ABC framework is used. The main idea of ABC is to estimate the joint PDF $p((\theta, y)|D)$ of the parameter set θ and the forward model prediction y. This can be done by applying Bayes' Theorem and the chain rule:

$$p(\theta, y|D) = \frac{p(\theta)p(y|\theta)p(D|\theta, y)}{p(D)}, \tag{2}$$

where $p(D|\theta, y)$ is approximated as

$$p(D|\theta, y) \approx P(\rho(y, D) \leq \delta|y) \tag{3}$$

for an arbitrary value of $\delta \geq 0$ and some discrepancy function $\rho(\cdot, \cdot)$ assigning smaller values for pairs (y, D) in which the arguments are closer to each other. Note that $P(\rho(y, D) \leq \delta|y)$ is simply an indicator function of the set $\mathcal{A}_\delta(D) = \{y : \rho(y, D) \leq \delta\}$.

Using Eq. (3), the approximate joint posterior PDF (2) reads:

$$p_\delta(\theta, y|D) = \frac{p(\theta)p(y|\theta)\mathbb{I}_{\mathcal{A}_\delta(D)}(y)}{p(D)} \propto p(\theta)p(y|\theta)\mathbb{I}_{\mathcal{A}_\delta(D)}(y). \tag{4}$$

Finally, to get the approximate marginal posterior PDF $p_\delta(\theta|D)$, the ABC sampling algorithm takes only the θ-components of the samples from $p_\delta(\theta, y|D)$.

2.2 Likelihood-Driven Bayesian Inference

In the likelihood-driven Bayesian inference, the likelihood is most often formulated using a stochastic forward model of the form:

$$y = x(\theta) + \epsilon, \tag{5}$$

where x is the outcome of the simulation used to obtain the quantity of interest for a given set of parameters θ and ϵ is an additive error term that quantifies the discrepancy between the data and the simulation outcome. The error term ϵ, which can be due to model error or measurement error or both, is assumed to follow a parameterized probability distribution with the PDF $p(\epsilon)$. The likelihood is then derived using Eq. (5) in which y is replaced with D.

As the exact inference, ABC can be used on the likelihood-based stochastic forward model (5), introducing however an additional uniform model error term defined by the discrepancy function and the tolerance δ [40].

3. ABC-SUBSIM

Recently, Chiachio et al. proposed to use the Subset Simulation (SubSim) technique [4] as an ABC algorithm (called ABC-SubSim) [9].

SubSim is an efficient rare event sampler. The algorithm represents a rare event F as an intersection of a nested sequence of events: $F = \cap_{k=1}^{m} F_k$ with $F = F_m \subset \ldots \subset F_1$, and then sequentially samples from conditional distributions $P(F_{k+1}|F_k)$, $k = 1, \ldots, m-1$. The probability of the event F can then be obtained as $P(F) = P(F_1)\prod_{k=1}^{m-1} P(F_{k+1}|F_k)$. One can estimate the first probability $P(F_1)$ using Monte Carlo Sampling [32]. The conditional probabilities can be estimated via the Markov Chain Monte Carlo (MCMC) method, in particular the Metropolis-Hastings (MH) transition kernel [15, 20, 26] as proposed in Ref. [4]. The advantage of the SubSum approach is that although the probability of the event F is small, the conditional probabilities $P(F_{k+1}|F_k)$ can be made sufficiently large by an appropriate choice of events F_k.

ABC-SubSim applies the same idea to the case of ABC. The rare event in this case is $\Delta = \{(\theta, y) : \rho(y, D) \leq \delta\}$ and the intermediate events are $Delta_k = \{(\theta, y) : \rho(y, D) \leq \delta_k\}$, where $\delta = \delta_m < \ldots < \delta_{k+1} < \delta_k < \ldots < \delta_1$, with δ_k chosen adaptively. Specifically, ABC-SubSim sequentially evaluates the approximate posteriors $p_{\delta_k}(\theta, y|D)$ (as stated in Eq. (4)) of increasing quality using the annealing scheme defined by the set of tolerances $\{\delta_k\}_{k=1}^{m}$. The workflow of ABC-SubSim is exactly the same as for the standard version of SubSim. For $k = 0$, ABC-SubSim produces samples from the prior and defines the first tolerance δ_1. For $k \geq 1$ ABC-SubSim runs multiple independent MCMC chains using the samples inside Δ_k as initial states of the chains (seeds) to produce more samples from $p_{\delta_k}(\theta, y|D)$. Then the next tolerance δ_{k+1} is defined. It is important to notice, that due to the choice of the seeds, ABC-SubSim (as well as SubSim) requires no burn-in period thus avoiding wasting samples [9].

This paragraph introduces the notations used in the rest of the paper. Different values of $k = 0, \ldots, m$ are referred to as stages. The n-th sample at the k-th stage is a pair $(\theta_k^{(n)}, y_k^{(n)}), n = 1, \ldots, N,\ k = 1, \ldots, m$, where N is the population size (the number of samples per stage). The discrepancy for the sample $(\theta_k^{(n)}, y_k^{(n)})$ is denoted as $\rho_k^{(n)} := \rho(y_k^{(n)}, D)$. We denote the set of all samples at the stage k as $S_k := \{(\theta_k^{(n)}, y_k^{(n)})\}_{n=1}^{N}$, and the corresponding set of discrepancies as $\mathrm{P}_k = \{\rho_k^{(n)}\}_{n=1}^{N}$.

3.1 Choice of the Intermediate Tolerances

In ABC-SubSim the intermediate tolerances $\delta_k, k = 1, \ldots, m$ are chosen adaptively in the same manner as in Subset Simulation. Namely, δ_{k+1} is chosen to be the $100P_0$ percentile of the set of discrepancies P_k sorted in the ascending order: $\delta_{k+1} = \frac{1}{2}(\rho_k^{(N/\ell)} + \rho_k^{(N/\ell+1)})$. Samples $(\theta_k^{(n)}, y_k^{(n)})$ with $\rho_k^{(n)} \leq \delta_{k+1}$ are used as seeds of the Markov Chains in MCMC to produce the $(k+1)$-th stage of samples.

3.2 Tuning MCMC within ABC-SubSim

In this section we discuss improvements regarding the tuning of the Metropolis-Hastings (MH) transition kernels when used inside ABC-SubSim in an effort to improve the overall algorithm's efficiency.

Markov chain length within MCMC: ABC-SubSim has a fixed Markov chain length $\ell = 1/P_0$. Each of the NP_0 samples with a discrepancy less than δ_{k+1} produces $(\ell - 1)$ new samples. The parameters N and P_0 have to be chosen in such a way that $1/P_0$ and NP_0 are integers. The fact that the length of the Markov chains is known a priori makes ABC-SubSim perfectly parallelizable. Since every Markov chain has the same length ℓ, it is possible to equally distribute NP_0 seeds among all available workers to get the perfect load balance within our large-scale parallel framework $\Pi 4U$ [18]. It has been demonstrated that choosing any P_0 in SubSim between 0.1 and 0.3 leads to similar efficiency [41]. The ABC-SubSim uses $P_0 = 0.2$ which gives the chain length $\ell = 5$.

The proposal distribution within MCMC: As in all MCMC-based methods, the proposal distribution should be chosen carefully in order to lead to uncorrelated samples. Following [9] we use a Gaussian proposal PDF. The spread of the proposal PDF is closely related to the acceptance rate of the MH chain, and hence to the sampling efficiency. A small spread means high acceptance rate which shows that the Markov chain is moving rather slowly, producing highly correlated samples and not exploring the parameter space fully. A large spread means low acceptance rate which shows that the chain is stagnating. One way to find the optimum spread of the proposal comes from the assumption that the interest lies not in the samples themselves, but in some QoI $h(\theta)$ which can be approximated using the samples at the stage k as:

$$\bar{h}_k = \frac{1}{N}\sum_{n=1}^{N} h(\theta_k^{(n)}) \tag{6}$$

with the variance

$$\mathrm{Var}(\bar{h}_k) = \frac{1}{N}\left(R_k^{(0)} + \frac{2}{\ell}\sum_{t=1}^{\ell-1}(\ell - t)R_k^{(t)}\right), \tag{7}$$

where $R_k^{(t)} = \mathbb{E}\left[h(\theta_k^{(1)})h(\theta_k^{(t+1)})\right] - \bar{h}_k^2$ is the autocovariance of $h(\theta)$ at lag t. In this approach, one can minimize the variance $\mathrm{Var}(\bar{h}_k)$ with respect to the spread of the proposal thus getting a better quality of the estimation. As shown in Ref. [41], the nearly optimal strategy is to choose the spread corresponding to an acceptance rate between 30% and 50%. This leads to a recommendation to rescale the proposal covariance matrix on a test chain of a small length until the acceptance rate is within this range. We use the chain length ℓ. The resulting procedure is sketched in the Alg. 1. The algorithm monitors the acceptance rate α and increases the spread of the proposal PDF if $\alpha > 0.5$ or decreases it if $\alpha < 0.3$. The output of the algorithm is the proposal covariance for a given stage. The algorithm is run once per stage starting from the sample with the smallest discrepancy. The updating of the covariance is done in parallel since the runs are independent.

Algorithm 1 Proposal covariance rescaling

```
Input:
  $\Sigma_0$, {sample covariance of the previous stage}
Output:
  $\Sigma_{prop}$, {proposal covariance}
Steps:
  $c_0 = 2$ // initial value of the scaling parameter
  for $t = 1, \ldots, 10$ do
      Set $\Sigma_{prop} = 2^{-2(t-1)} c_0^2 \Sigma_0$
      Run MCMC and evaluate its acceptance rate:
      $\alpha$ = acceptance rate of MCMC$((\theta_{k-1}^{(0)}, y_{k-1}^{(0)}), \ell, \Sigma_{prop})$
      if $0.3 \leq \alpha \leq 0.5$ then
          return $\Sigma_{prop}$
      end if
  end for
  // return the initial value if a good one was not found
  return $c_0^2 \Sigma_0$
```

Acceptance criterion: the key component of the MH algorithm is a sample acceptance rule it uses. In ABC-SubSim the sample (θ', y') produced from the sample $(\theta_k^{(n)}, y_k^{(n)})$ within a Markov chain is accepted at the stage k with probability

$$r = \min\left\{1, \frac{p(\theta')\; q(\theta_k^{(n)}|\theta')}{p(\theta_k^{(n)})\; q(\theta'|\theta_k^{(n)})}\right\} \mathbb{I}_{\mathcal{A}_{\delta_k}(D)}(y'), \tag{8}$$

where $q(\cdot|\cdot)$ is the proposal PDF. The acceptance probability here is defined by the prior PDF and the indicator function which equals 1 only if $y' \in \mathcal{A}_{\delta_k}(D)$.

Stopping criterion: there is no rigid convergence criterion for the termination of ABC-SubSim, thus leaving us with empirical and computational insights. In our implementation we check for two stopping criteria and terminate when at least one is fulfilled: 1) reaching a specified change in the tolerance, 2) reaching a specified acceptance rate in the MH. The QoI in our case is the stochastic forward model prediction y meaning that the criterion 1) provides an estimate on how much the approximation of the QoI improved, which is the criterion proposed in Ref. [9]. The stopping criterion 2) comes into play when samples get rejected not because of getting in the area of big discrepancy values in the θ-space, but because of the natural variability of y. In that case tuning the covariance matrix does not help since even for a fixed θ from a good region some samples produced by $y(\theta)$ can have a too high discrepancy to be accepted at the current stage. When reporting the results, we indicate which stopping criterion was reached first and provide the corresponding tolerance δ.

Alg. 2 gives a pseudocode implementation of ABC-SubSim. The proposal distribution $q(\cdot|\cdot)$ for the MH algorithm is assumed to be Gaussian with mean 0 and covariance Σ_{prop} which is to be defined adaptively.

Algorithm 2 ABC-SubSim pseudocode

```
Input:
  $N$, {population size}
  $\ell$, {chain length for MH, s.t. $N/\ell$ is an integer value}
Steps:
  Initialize $S_0 = \{(\theta_0^{(n)}, y_0^{(n)})\}_{n=1}^N$ with $\theta_0^{(n)} \sim p(\theta), y_0^{(n)} \sim p(y|\theta)$
  Initialize $k = 0$
  while stopping conditions are not met do
      Evaluate array of discrepancies $\mathrm{P}_{k-1}$ for samples $S_{k-1}$
      Renumber samples $S_{k-1}$, s.t. their discrepancies are sorted in the ascending order: $\rho_{k-1}^{(1)} \leq \ldots \leq \rho_{k-1}^{(N)}$
      Set annealing parameter $\delta_k = \frac{1}{2}\left(\rho_{k-1}^{(N/\ell)} + \rho_{k-1}^{(N/\ell+1)}\right)$
      Set $S_k = \{(\theta_{k-1}^{(n)}, y_{k-1}^{(n)})\}_{n=1}^{N/\ell}$
      Set proposal covariance $\Sigma_{prop}$ for MH using Alg. 1
      for $s = 1, \ldots, N/\ell$ do
          Run MH with seed $(\theta_{k-1}^{(s)}, y_{k-1}^{(s)})$, chain length $\ell$, proposal covariance $\Sigma_{prop}$ and acceptance probability defined by Eq. (8) to produce samples $\left[(\theta_k^{(\ell(s-1)+1)}, y_k^{(\ell(s-1)+1)}), \ldots, (\theta_k^{(ls)}, y_k^{(ls)})\right] \in F_k$
          Append produced samples to $S_k$
      end for
      set $k = k + 1$
  end while
  Report the final tolerance $\delta_k$
```

4. ABC-SUBSIM VERSUS TMCMC: ALGORITHMIC COMPARISON

In this section we compare the ABC-SubSim [9] with the TMCMC [10] algorithm, both fitting into the general framework of Sequential Monte Carlo samplers [11]. ABC-SubSim is a more general algorithm that can be naturally applied for both likelihood-free and likelihood-driven Bayesian inference, while TMCMC only applies to the latter. Alg. 3 gives a pseudocode implementation of TMCMC. Details about the algorithm and the notations can be found in [1, 10, 3].

Both algorithms are population-based sampling methods. To overcome the difficulties of rare event sampling each of them uses an annealing scheme of moving from the prior to the posterior distribution. To do this they introduce intermediate target distributions and sample sequentially from them. Both methods use MH transition kernels to produce the next stage of samples from the current one, although the details of the annealing strategy and the selection of seeds for MH differ. Both algorithms provide the evidence as a by-product, without requiring extra system simulations. Table 1 summarizes the main features of the two algorithms, revealing similarities and differences.

An advantage of TMCMC to ABC-SubSim is its well defined stopping criterion. TMCMC terminates when the annealing parameter $\tilde{p}_k$ (see Table 1) reaches the value of 1. ABC-SubSim terminates when the acceptance rate is below a pre-specified arbitrary threshold value. For likelihood-driven Bayesian inference, TMCMC provides samples from the posterior distribution based on the prediction error equation defined in (5). ABC-SubSim provides samples of the posterior distribution based on the prediction error equation (5) that includes an additional error term uniformly distributed in the domain $\rho(y, D) \leq \delta$. Using the theoretical result in [40] for ABC, the two methods coincide when $\delta \to 0$ which requires a extremely high computational effort for ABC-SubSim [9].

An important feature to compare is the potential efficiency of parallelization. An advantage of ABC-SubSim

Table 1: Comparison between ABC-SubSim and TMCMC.

	TMCMC	ABC-SubSim
Annealing method	Choose $\tilde{p}_{k+1}$ at the stage k such that the coefficient of variation of the weights $w(\theta_k^{(n)}) = \left[p(D\|\theta_k^{(n)})\right]^{\tilde{p}_{k+1}-\tilde{p}_k}, n = 1, \ldots, N$ equals to a prescribed threshold $tolCOV$	Gradually decrease the tolerance such that the tolerance δ_{k+1} equals to the $100P_0$ percentile of the array P_k of discrepancies
Selection strategy	Select $\theta_k^{(n)}$ as a seed for the $(k+1)$-th stage with probability $\tilde{w}(\theta_k^{(n)}) = w(\theta_k^{(n)}) / \sum_{n=1}^{N} w(\theta_k^{(n)})$	Select $\left(\theta_k^{(n)}, y_k^{(n)}\right)$ as a seed for the $(k+1)$-th stage if $\rho_k^{(n)} \leq \delta_{k+1}$
Acceptance in MH	Defined by Eq. (9) (likelihood-based)	Defined by Eq. (8) (discrepancy-based)
Stopping criterion	Stop when the annealing exponent $\tilde{p}_k \geq 1$	Stop when δ_k does not change significantly or the acceptance rate of the MH is below 5%

comes from the fact that it is perfectly parallelizable due to the fact that all Markov chains have the same length specified a priori. It means that the work can be equally split among available work units. TMCMC has a variable Markov chain length which has to be defined at run time, an imbalance which needs to be addressed. Similarly, we observe a load imbalance in models with a variable execution time conditional on their parameters. To perform an unbiased parallelization efficiency comparison amongst the two schemes, we based the implementation of both ABC-SubSim and TMCMC on the TORC tasking library [17].

4.1 Integration with Surrogates

In case of exceedingly expensive function evaluations, surrogates can be used instead of the real function evaluations. Although they introduce an additional layer of approximation, surrogates can provide a reasonable estimate of the posterior PDF, if used with care. In this paragraph we discuss the usage of adaptive kriging as a surrogate inside ABC-SubSim and TMCMC. Kriging is an interpolation method, and in order to achieve a good quality of approximation it needs to have the prediction point within the convex hull of the data points. The method can be applied to non-regular meshes.

It is worth pointing out that ABC-SubSim and TMCMC share algorithmic similarities that can be exploited to apply adaptive kriging as a surrogate model to reduce by an order of magnitude the number of full model simulations as was proposed in [3]. To illustrate this, the Δ_k and Δ_{k+1} failure domains in ABC-SubSim (denoted by their corresponding tolerances δ_k, δ_{k+1}) with samples drawn up to stage k (blue dots) are depicted in Fig. 1a. All failure domains are nested, with the samples drawn up to stage k populating the region outside the boundary surface of the domain Δ_{k+1}. A chain generated at stage k from a seed sample located in the domain Δ_k produces samples aiming at populating the failure domain Δ_k, as shown with the red chain in Fig. 1a. Note that there exist samples (blue dots) generated from ABC-SubSim up to stage k that contain in their convex hull the red chain.

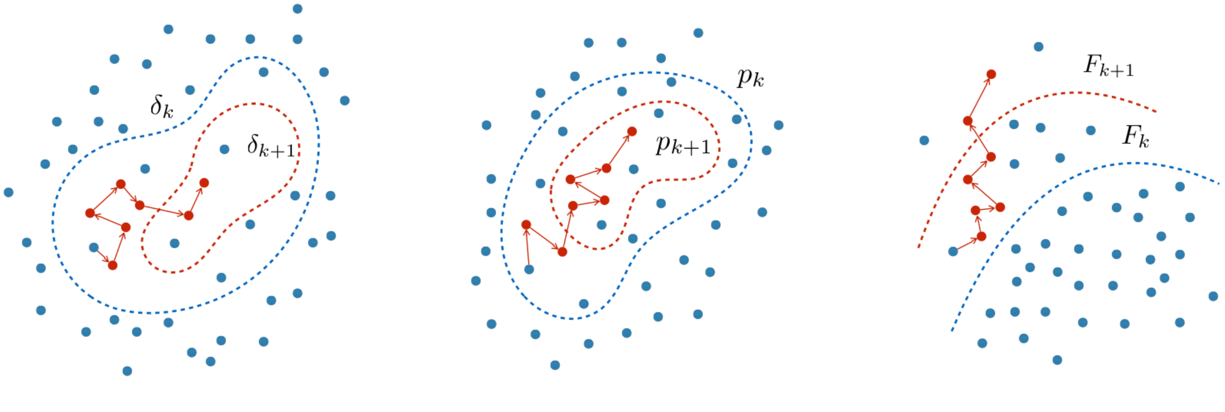
This feature of ABC-SubSim is similar to the one encountered in TMCMC where the main support of intermediate posterior PDFs are also nested domains, shown in Fig. 1b, so that a chain generated by a seed sample at stage k is contained within the convex hull of sample points generated from TMCMC up to stage k. This feature was exploited in [3] in order to introduce an adaptive kriging method within TMCMC, termed K-TMCMC, using the full model simulations at the neighborhood of the seed sample from previous TMCMC stages as support points for performing a kriging estimate, achieving up to one order of magnitude reductions of the number of expensive full model simulations. The same concept can thus be readily applied to ABC-SubSim. From this perspective, ABC-SubSim and TMCMC present significant similarities that allow adaptive kriging to be used within the Π4U software to reduce the number of full model simulations.

The aforementioned feature is worth contrasting with the original SubSim used in BUS [36] for Bayesian parameter estimation as an alternative to MCMC methods such as TMCMC and ABC-SubSim. The nested failure domains in SubSim and the samples drawn up to stage k (blue dots) are shown in Fig. 1c. Each chain generated using SubSim, shown by red color in Fig. 1c, can be located outside of the convex hull of support points generated from SubSim up to stage k. This can happen when the aim is to estimate the tail of the distribution, and the chains move towards the outer unexplored regions in the uncertain parameter space, aiming at populating the next failure domain F_{k+1}. Thus, adaptive kriging estimates are less likely to be effective, in terms of accuracy, due to lack of support points from previous stages able to contain within their convex hull the chain generated from k-stage seed samples.

5. NUMERICAL EXPERIMENTS

5.1 Discrete Element Model (DEM) Using ABC-SubSim and TMCMC

To compare the effectiveness of ABC-SubSim and TMCMC for likelihood-driven Bayesian inference we calibrate a force-displacement model for steel particles. The computational model for this case is a DEM simulation of the collision of a 2D disk particle and a wall (treated as a 2D disk with a very small curvature). The simulation setup and



(a) ABC-SubSim. (b) TMCMC. (c) Subset Simulation.

Figure 1: Schematic representation of intermediate domains at stages k and $(k+1)$ (dashed lines), samples up to stage k (blue dots), and chains generated from the seed samples at the stage k (red dots) for various algorithms.

calibration with TMCMC is described in [19].

The simulation has the following input properties: Young's modulus $E = 2.10 \times 10^{10}$, Poisson ratio $\nu = 0.3$, disk radius $R_i = 0.02225$ m, mass $m = 0.3538$ kg. As a normal force-displacement model between the particles, the extended non-linear model $F^n = -k^n \xi^{3/2} - \gamma^n \dot{\xi} \xi^{\alpha^n}$ is taken. The tangential component of the force can be obtained as $F^t = -|\kappa^t \xi^t| \text{sign}(\xi^t)$, where $\xi^t = \int_{t_0}^{t} v_t H(\mu |F^n| / \kappa^t - |\xi^t|) dt'$. Here $F = F^n + F^t$ is the force exerted on a particle due to a contact with another particle or a wall, ξ is a mutual compression between particles defined as the distance between their centers minus the sum of radii, $H(\cdot)$ is the Heaviside step function and v_t is the relative tangential velocity between the two particles at their contact point.

This model has 4 parameters to be calibrated: $\mu, \alpha^n, \gamma^n, \kappa^t$. The data for the calibration includes the normal and the tangential coefficients of restitution ζ^n, ζ^t along with their uncertainties η^n, η^t at different impact angles ϕ, see Table 2.

The Molecular Mechanics calibration problem illustrated in this work is a fraction of a complete force-field (FF) calibration in the sense that we look at the distribution of only one quantity of interest. In typical FF calibrations, one tries to fit multiple QoIs, e.g multiple scans of atomic quantum calculations.

Table 2: Experimentally measured normal and tangential coefficients of restitution $\zeta_j^n, \zeta_j^t, j = 1, \ldots, 6$ and their uncertainties $\eta_j^n, \eta_j^t, j = 1, \ldots, 6$ depending on the impact angle ϕ. Data from [12].

ϕ	ζ_j^t	η_j^t	ζ_j^n	η_j^n
10.0	0.523	0.010	0.890	0.020
20.0	0.524	0.014	0.892	0.015
30.0	0.637	0.044	0.918	0.014
40.0	0.779	0.013	0.896	0.005
50.0	0.846	0.024	0.918	0.013
60.0	0.886	0.024	0.874	0.033

Similarly to Ref. [19] the model and measurement errors for each coefficient of restitution are assumed to be zero-mean Gaussian and independent so that the combined error term in (5) is given by $\epsilon \sim N(0, \Sigma)$ with $\Sigma = \text{diag}(\eta_j^2 + \sigma^2 \zeta_j^2)$, where $\sigma^2 \zeta_j^2$ is the variance of the model error for each response quantity. The prediction error parameter σ^2 is unknown and has to be inferred, thus we extend the parameter set to $\theta = (\mu, \alpha^n, \gamma^n, \kappa^t, \sigma^2)$. To apply the ABC-SubSim algorithm, we selected as a discrepancy the Euclidean distance: $\rho(y, D) = ||y - D||_2$. The prior PDF of the model parameters was set to be uniform with large enough bounds so that they do not affect the parameter inference.

The detailed information about the prior and the posterior values of the parameters is given in the Table 3. The number of samples, stages and the achieved tolerance for ABC-SubSim are reported in Table 3. It should be observed that the number of stages are very close for both algorithms, while the final δ value cannot be smaller than 0.16.

The results of the calibration with ABC-SubSim and TMCMC are given in Fig. 2. Inspection of the posterior plots in Fig. 2 suggests that there is an unidentifiable manifold, evident in the projection of the samples in the (α^n, γ^n) parameter space. Specifically, the predictions of the measured QoI considered are insensitive to parameter values along certain directions in the parameter space.

Both ABC-SubSim and TMCMC algorithms give qualitatively similar results. They both capture the bi-modality evident in the parameter κ^t, as well as the unidentifiable manifold evident in the projection in the two-dimensional parameter space (α^n, γ^n). It is also observed that ABC-SubSim gives higher uncertainty in the parameter space than TMCMC, which is due to the non-zero value obtained for the annealing algorithmic parameter δ. According to Wilkinson [40], extra uniformly distributed prediction error arises in the domain $\rho(y, D) \leq \delta = 0.16$ which is responsible for the quantitative discrepancies in the parameter inference results obtained from ABC-SubSim and TMCMC.

Evaluating the quality of the results of the calibration is a much disputed concept. Non-Bayesian tests like the posterior p-values can be employed for this task [14, 25]. In our case, based on the posterior uncertainty of the model parameters, we can conclude that given the data we used, γ^n is unidentifiable, μ has a very peaked posterior distribution (well defined), α^n and κ^t are in between these extreme situations. For more results and discussion see Ref. [19].

Table 3: Prior and posterior information about the parameters $\theta = \left(\mu, \alpha^n, \gamma^n, \kappa^t, \sigma^2\right)$ of the DEM problem for ABC-SubSim and TMCMC. In the table are listed: the uniform prior bounds $[\theta^j_\ell, \theta^j_r]$, the mean values $\bar{\theta}^j$, the coefficients of variation $u_{\theta^j}, j = 1, \ldots, 5$ of the parameters, the number of samples N_{sam}, the number of stages N_{stg}, the achieved tolerance δ for ABC-SubSim.

	$[\mu_\ell, \mu_r]$	$\bar{\mu}$	u_μ	$[\alpha^n_\ell, \alpha^n_r]$	$\bar{\alpha}^n$	u_{α^n}	$[\gamma^n_\ell, \gamma^n_r]$	$\bar{\gamma}^n$	u_{γ^n}
ABC-SubSim	[0.05, 0.2]	0.106	8.8%	[0.15, 1.35]	0.47	17.5%	$[0.1, 15]\times10^4$	7.33×10^4	53.5%
TMCMC	[0.05, 0.2]	1.05	4.5%	[0.15, 1.35]	0.44	20.1%	$[0.1, 15]\times10^4$	6.52×10^4	61.8%

	$[\kappa^t_\ell, \kappa^t_r]$	$\bar{k}^t$	u_{κ^t}	$[\sigma^2_\ell, \sigma^2_r]$	$\bar{\sigma}^2$	u_{σ^2}	N_{sam}	N_{stg}	δ
ABC-SubSim	[0.1, 15]	0.99	31.2%	$[0, 10]\times10^{-5}$	4.40×10^{-5}	62.9%	50,000	7	0.16
TMCMC	[0.1, 15]	1.01	29.4%	$[0, 10]\times10^{-5}$	5.52×10^{-5}	49.2%	50,000	8	–

Algorithm 3 TMCMC pseudocode

```
Input:
  $N$, {population size}
  $tolCOV$, {parameter defining the speed of the annealing}
Steps:
  Initialize $\Theta_0 = \left\{\theta_0^{(n)}\right\}_{n=1}^{N}$, where $\theta_0^{(n)} \sim p(\theta)$
  Initialize the annealing parameter $\tilde{p}_0 = 0$
  Initialize $k = 0$
  while $\tilde{p}_k \leq 1$ do
    Evaluate plausibility weights $w_k = \left\{w(\theta_k^{(n)})\right\}_{n=1}^{N}$, where $w(\theta_k^{(n)}) = \left[p(D|\theta_k^{(n)})\right]^{\tilde{p}_{k+1}-\tilde{p}_k}$ and the annealing parameter $\tilde{p}_{k+1}$ is chosen such that the coefficient of variation (COV) of the weights $w_k$ equals to a prescribed tolerance $tolCOV$
    Select $L$ samples $\theta_k^{(s)}$, $s = n_1, \ldots, n_L$, with probability $\tilde{w}_k^{(s)} = w(\theta_k^{(s)}) / \sum_{n=1}^{N}(w(\theta_k^{(n)}))$ as seeds for $(k+1)$-th stage. The sample $\theta_k^{(s)}$ is selected $\ell_k^{(s)}$ times such that $\sum_{s=n_1}^{n_L} \ell_k^{(s)} = N$
    Append these seed samples to $\Theta_{k+1}$
    Set proposal covariance $\Sigma_{prop}$ for MH to be equal to the sample covariance of samples $\Theta_{k-1}$ (alternatively, use Alg. 1)
    Set the acceptance probability for a sample $\theta'$ inside MH as
    $r = \min\left\{1, \frac{p(\theta')p(D|\theta')^{\tilde{p}_{k+1}-\tilde{p}_k} q(\theta_k^{(prev)}|\theta')}{p(\theta_k^{(prev)})p(D|\theta^{(prev)})^{\tilde{p}_{k+1}-\tilde{p}_k} q(\theta'|\theta_k^{(prev)})}\right\},$   (9)
    where $\theta^{(prev)}$ is the previous sample.
    for $s = 1, \ldots, L$ do
      Run MH with a seed $\theta_k^{(s)}$, a variable chain length $\ell_k^{(s)}$, a proposal covariance $\Sigma_{prop}$ and MH acceptance probability defined by the Eq. (9) to produce samples for the next stage $(k+1)$
      Append produced samples to $\Theta_{k+1}$
    end for
    set $k = k + 1$
  end while
```

5.2 Lennard-Jones Parameters of Helium Using ABC-SubSim

We next consider the calibration of the Lennard-Jones potential parameters for helium using the likelihood-free Bayesian inference. The Lennard-Jones potential is given by

$$V_{LJ}(r; \sigma_{LJ}, \epsilon_{LJ}) = 4\epsilon_{LJ}\left(\left(\frac{\sigma_{LJ}}{r}\right)^{12} - \left(\frac{\sigma_{LJ}}{r}\right)^{6}\right), \quad (10)$$

where ϵ_{LJ} is the depth of the potential well (measured in zeptoJoules), σ_{LJ} is the finite distance at which the inter-particle potential is zero (measured in nanometers), and r is the distance between the particles. The parameters ϵ_{LJ} and σ_{LJ} are uncertain and should be calibrated given the data.

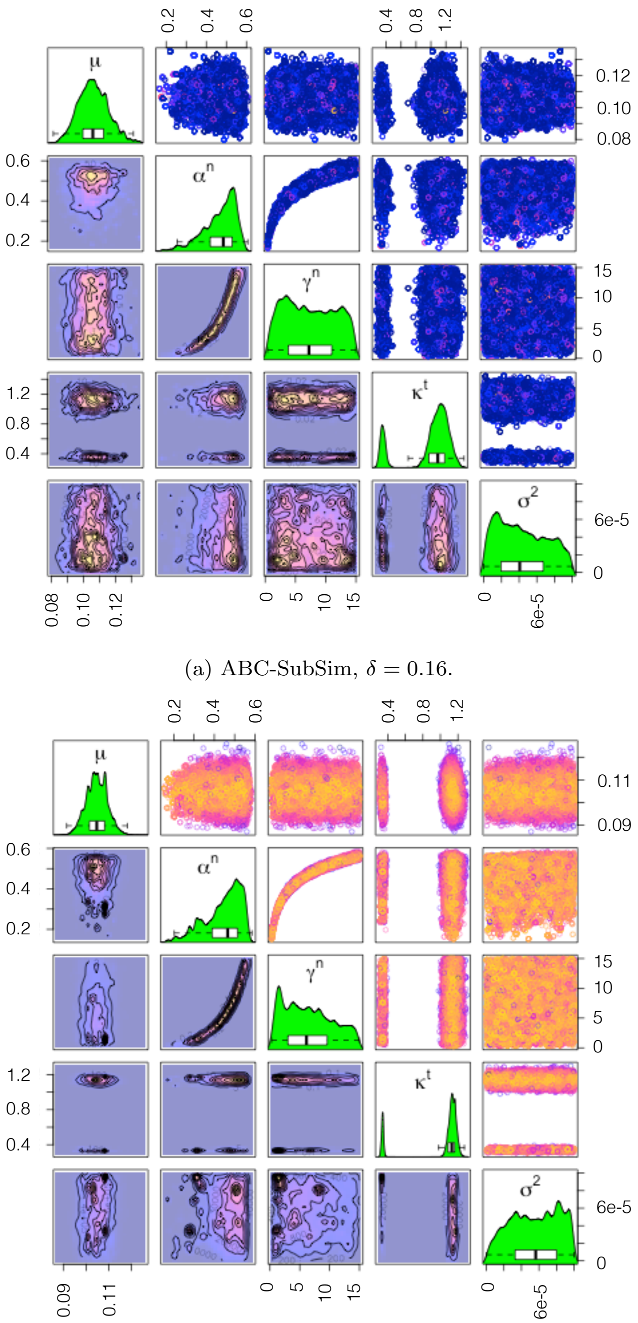
To perform the calibration we used the data on the Boltzmann factor $f_B = \langle \exp\{-H/(k_B T)\}\rangle$, where H is the enthalpy of the system of helium atoms, T is the temperature of the system, k_B is the Boltzmann constant, and $\langle\cdot\rangle$ denotes the ensemble average. The ensemble average in this MD context indicates that the enthalpy is measured at each time instance over all interacting atoms (see Fig. 3a). The data was generated using the software LAMMPS [28] for a system of 1000 atoms for 20 ns in the NPT ensemble [35] with a timestep of 2 fs. The system used for calibration consists of 1000 atoms and is equilibrated for 2 ns, following a production run in the NPT ensemble for another 5 ns with a 2 fs timestep. Note that the model cannot replicate exactly the target data due to the smaller sampling time. This is a typical situation that can arise during calibration of atomistic quantities, where the calibration is done with limited resources.

The distribution of the Boltzmann factor is known to be non-Gaussian due to its statistical thermodynamics definition (see Fig. 3a). Exploiting the flexibility of ABC, we will employ a discrepancy function in an effort to capture higher order moments of the distribution, inspired by the physics of the model.

We performed calibration with 3 different settings. In all the settings below y stands for the Boltzmann factor values obtained in the simulation and D stands for those obtained in the experiment.

In the first setting we selected the discrepancy to be a function of the first two moments of the corresponding quantities:

$$\rho(y, D) = \left(\left(\frac{\bar{y} - \bar{D}}{\bar{y}}\right)^2 + \left(\frac{s_y - s_D}{s_y}\right)^2\right)^{1/2}, \quad (11)$$



(a) ABC-SubSim, $\delta = 0.16$.

(b) TMCMC.

Figure 2: Results of parameter calibration of the DEM problem using 50000 samples. All samples are taken from the last stage. Diagonal: marginal distribution of parameters estimated using kernel histograms accompanied by a Tukey boxplot. Above the diagonal: projection of the samples of the posterior distributions of all pairs of 2-d parameter space. Below the diagonal: projected densities in 2-d parameter space constructed via a bivariate kernel estimate.

where $\bar{X}$ denotes a sample mean and s_X denotes the sample standard deviation of the quantity X. This discrepancy is based on a sufficient statistic of the Gaussian distribution and is intended to mimic the likelihood-driven approach which usually assumes Gaussianity of the error term.

In the second setting we selected the discrepancy to be a function of the quintiles (5-quantiles) QU of the data sets:

$$\rho(y, D) = \left(\sum_{k=1}^{4} \left(\frac{QU_k[y] - QU_k[D]}{QU_k[y]} \right)^2 \right)^{1/2} . \qquad (12)$$

Finally in the third setting we set the discrepancy to be:

$$\rho(y, D) = \int \chi(z) \log \frac{\chi(z)}{\psi(z)} dz, \qquad (13)$$

which is a relative entropy (also called Kullback-Leibler divergence) of the PDF χ of D and the PDF ψ of y. The PDFs χ and ψ are computed by a kernel density estimate.

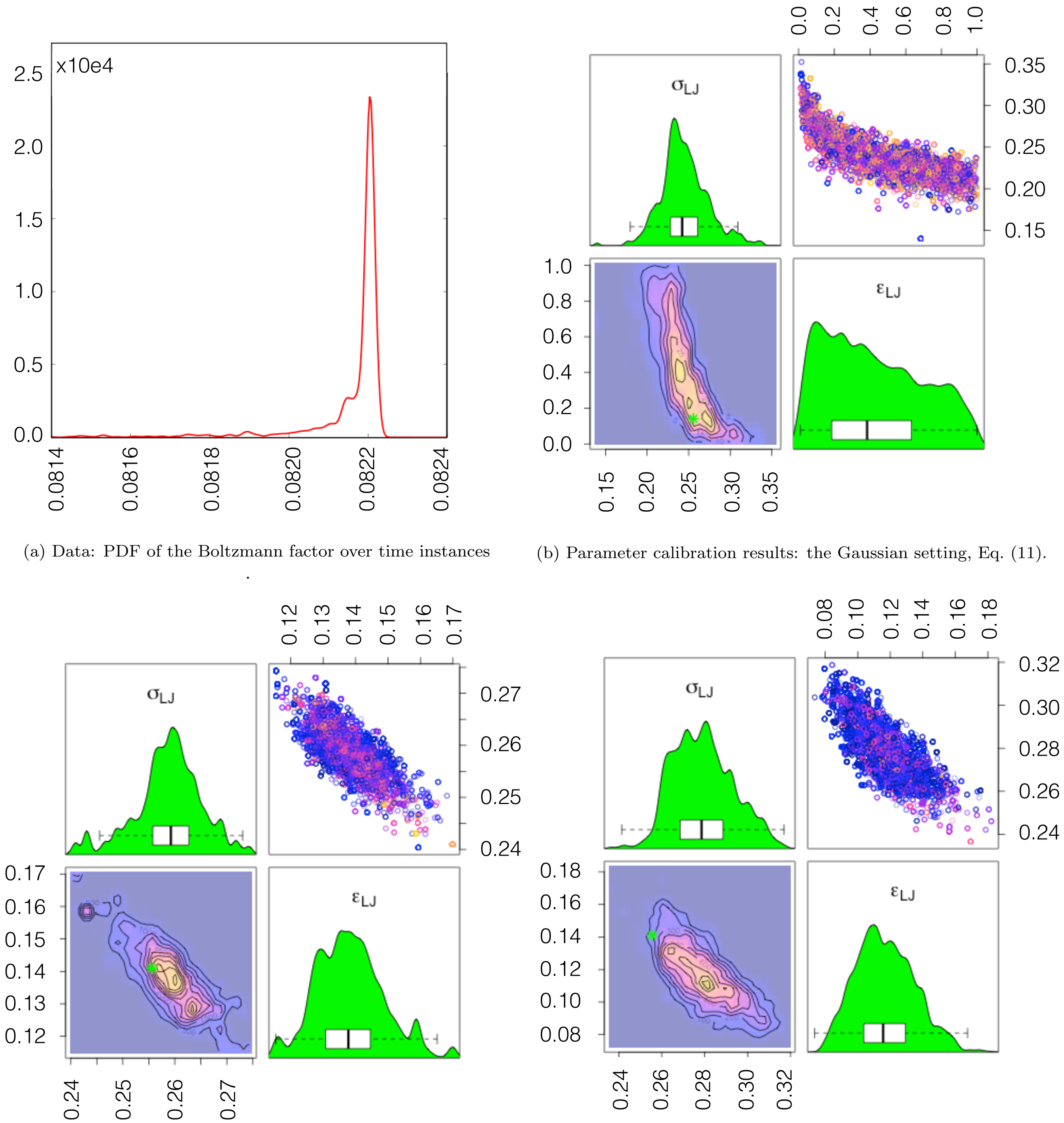
Note that it is not a symmetric distance function, but this does not violate any assumptions as we never commute the target data set with the one produced during the calibration. The motivation behind the relative entropy as a discrepancy, comes from the work of Shell [34], where he treats it as a discrepancy measure between the ensembles of a target and a model molecular configuration.

The information about the prior and the posterior values of the parameters is given in Table 4, and the calibration results are depicted in Fig. 3.

We observe that the results for the three discrepancy functions differ both in terms of the most probable parameters as well as their associated uncertainty. Under the stochastic model class M_G where we assume that the PDF of the Boltzmann factor f_b can be approximated by a Gaussian (Fig. 3b), the parameter ϵ_{LJ} is unidentifiable, as the approximate posterior PDF spans across all the prior range. The reason for this is that the standard deviation σ_y of the Boltzmann factor f_B (see the Eq. (11)) appears to be insensitive to changes along certain directions in the parameter space (ϵ_{LJ}, σ_{LJ}), while this standard deviation contributes significantly more to the discrepancy (11) (a typical contribution of the fraction $(\bar{y} - \bar{D})/\bar{y}$ is around 3%, where for the fraction $(\sigma_y - \sigma_D)/\sigma_D$ is around 97%). Note that the coefficient of variation of σ_{LJ} is quite large, with a value of 11.5%. The stopping criterion based on acceptance is reached after 4 stages, with a discrepancy $\delta = 0.0034$.

Inspection of Fig. 3a reveals the skewed nature of the Boltzmann factor distribution, which is restrictively pruned when modeled with a Gaussian distribution. The difference in the calibration procedure when using the quantile setting (Fig. 3c) and the relative entropy discrepancy (Fig. 3d) compared to the Gaussian setting (Fig. 3b), lies mainly in their ability to reduce the uncertainty of both ϵ_{LJ} and σ_{LJ}. There are no significant differences between the quantile and the relative entropy settings. We note that the quantile setting has identified however a smaller uncertainty in the parameters.

The calibration campaign using discrepancy models M_Q and M_{RE} identified the force-field parameters, employing a full MD run per sample Fig. 3d , whereas M_G resulted in a wide PDF that spans across the entire prior range Fig. 3b. This happened because we have effectively reduced the information regarding the ensemble PDF of the quantity of interest into 2 numbers in the Gaussian case, the mean and vari-



(a) Data: PDF of the Boltzmann factor over time instances

.

(b) Parameter calibration results: the Gaussian setting, Eq. (11).

(c) Parameter calibration results: the quantile setting, Eq. (12).

(d) Parameter calibration results: the relative entropy setting, Eq. (13).

Figure 3: (a) Data: distribution of the Boltzmann factor $f_B[pg^{-1}]$ of a Helium system with 1000 atoms, $\sigma_{LJ} = 0.2556, \epsilon_{LJ} = 0.141$. (b), (c), (d): Results of parameter calibration of the Lennard-Jones parameters of Helium using ABC-SubSim. All samples are taken from the last stage. Diagonal: marginal distribution of parameters estimated using kernel histograms accompanied by the Tukey boxplot. Above the diagonal: projection of ABC-SubSim samples of the posterior distributions of all pairs of 2-d parameter space colored by discrepancy. Below the diagonal: projected densities in 2-d parameter space constructed via a bivariate kernel estimate. The green star indicates the parameters for which the data were created.

Table 4: Prior and posterior information of parameters $\theta = (\sigma_{LJ}, \epsilon_{LJ})$ of the Helium system. The table shows: the prior bounds $[\theta^j_\ell, \theta^j_r]$, the mean values $\bar{\theta}^j$, the coefficients of variation $u_{\theta^j}, j = 1, 2$ of the Lennard-Jones parameters of Helium, the number of stages N_{stg}, the achieved tolerances δ. The information is given for three models: M_G [Gaussian setting (11)], M_Q [quantile setting, Eq. (12)], M_{RE} [relative entropy setting, Eq. (13)].

Model	$[\sigma_{LJ,\ell}, \sigma_{LJ,r}]$	$\bar{\sigma}_{LJ}$	$u_{\sigma_{LJ}}$	$[\epsilon_{LJ,\ell}, \epsilon_{LJ,r}]$	$\bar{\epsilon}_{LJ}$	$u_{\epsilon_{LJ}}$	N_{stg}	δ
M_G	[0.1,0.8]	0.2452	11.5%	[0.01,1.0]	0.423	64.5 %	4	3.40×10^{-3}
M_Q	[0.1,0.8]	0.2588	2.5 %	[0.01,1.0]	0.138	7.5 %	7	9.55×10^{-6}
M_{RE}	[0.1,0.8]	0.2792	5.0%	[0.01,1.0]	0.117	15.4 %	6	6.70×10^{-2}

ance, which are not enough to uniquely identify the potential values. This is a typical approach however in potential calibration and hierarchical coarse graining strategies [27] in MD, where the values of the QoI (e.g. intermolecular forces) are taken to be the ensemble averages of the coarse and fine molecular models to be matched. Following however the work of Shell [34], the relative entropy can be seen as a likelihood-type approximation to inverse model calibration in the case of using thermodynamic ensembles such as the Boltzmann distribution as QoIs. Thus, ABC equipped with a relative entropy discrepancy provides a natural framework for the development of parallel and efficient sampling algorithms such as ABC-SubSim implemented herein to provide accurate identification of the force field parameters when using thermodynamic ensemble target data in molecular simulations as in hierarchical coarse graining. This is achieved by exploiting the full non-Gaussian description of the MD model thermodynamic ensemble, and by not using Gaussian likelihoods between the target and trial PDFs, as in the case of [1, 8, 13]. The usage of Gaussian likelihoods is a typical step taken in Bayesian model updating, which in the case of hierarchical calibration of MD models unnecessary compresses the available information from the available thermodynamic distributions.

5.3 HPC Aspects on Cray-XC30

Based on the algorithmic similarities with the TMCMC algorithm, the ABC-SubSim algorithm was implemented within our task-parallel Π4U open source framework [18]. This platform-agnostic framework, based on the TORC task-parallel library [17], extracts and schedules the hierarchical task-based parallelism of the algorithm on multicore heterogeneous clusters, providing both ease of programming and transparent load balancing through work stealing. Fig. 4 depicts the hierarchical task graph of the ABC-SubSim algorithm.

In all the cases the samples per stages were set to be 15360. In every case the algorithm stopped because of the acceptance-based criterion (the acceptance rate dropped below 5%). The algorithm runs a full molecular dynamic simulation for every parameter set and hence requires a significant amount of computational work. The algorithm exploits two levels of parallelism, as the different independent Markov chains can be processed in parallel while each single LAMMPS simulation [28] is also run in parallel using the Message Passing Interface (MPI). The time to solution for each function evaluation varies with the given parameters, introducing load imbalance in the algorithm. We deal with this issue by applying the Longest Processing Time algorithm [16], according to which tasks with higher execution time are processed first. As the execution time increases for lower values of the σ_{LJ} parameter, we sort the samples according to this parameter before distributing the corresponding function evaluation or Markov chain tasks to the workers. The proposed task distribution scheme, combined with the task stealing mechanism of TORC, minimizes the maximum completion time of the tasks.

We performed our simulations on 512 compute nodes of the Piz Daint Cray XC30 cluster at the Swiss National SuperComputing Center CSCS. Each node is equipped with an 8-core Intel Xeon E5-2670 processor, resulting in 4096 cores in total. TORC is initialized with two MPI workers per node and each LAMMPS simulation utilizes 4 cores in turn.

Table 5 summarizes the parallel performance of ABC-SubSim. We provide the detailed results only for the case of Gaussian discrepancy, since the other two settings give similar outcomes. Despite the high variance of the time for a single simulation run, we observed that the efficiency of the initialization phase (stage 0) reaches 89% as 15360 function evaluations are distributed among the 1024 workers. The lower efficiency (64%) of stage 1 is attributed to the ex-

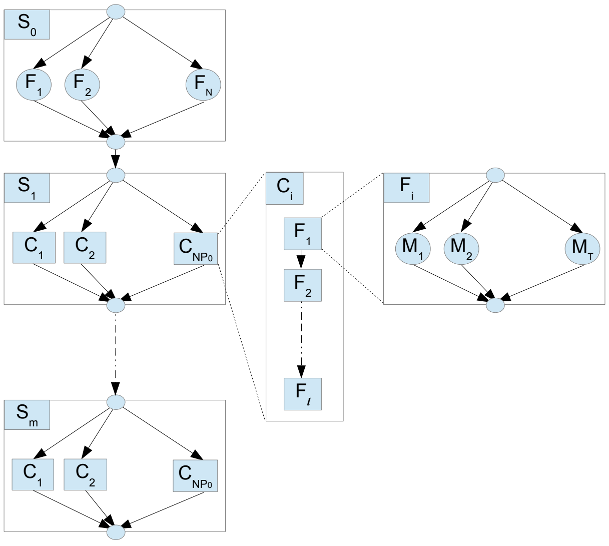


Figure 4: Algorithmic task-graph of ABC-SubSim. The leftmost component represents the highest level of the algorithm: the first stage (S_0) spawns N function evaluation tasks (F_i), which in turn involve up to T model runs (M_i) that are submitted for asynchronous execution. Next stages spawn NP_0 chains (C_i) of length ℓ, depicted in the middle of the figure. A function evaluation is invoked at each step of a chain and, as before, one or multiple model runs can be subsequently executed.

Table 5: Detailed per-stage performance results of ABC-SubSim on 512 compute nodes of the Piz Daint cluster with 2 workers per node. T_f shows the mean and standard deviation of the simulation times, T_{total} is the aggregate execution time of all simulations and T_w is the wall-clock time per stage, respectively. All the times are reported in seconds. The speedup and efficiency are reported for 1024 workers.

Stage	T_f	T_{total}	T_w	Speedup	Efficiency
0	83 ± 83	1270004	1393	912	89%
1	91 ± 61	1397644	2131	656	64%
2	69 ± 15	1061472	1491	712	70%
3	68 ± 13	1045711	1373	762	74%
4	67 ± 6	1026819	1149	894	87%

istence of chains with high accumulated running times and the small number of available chains that correspond to each worker (3072 chains in total, 3 chains per worker). As the algorithm evolves, the efficiency increases and reaches 87% for the last stage, which exhibits a load imbalance of approximately 13% as computed by $\frac{T_{max}-T_{avg}}{T_{max}} = \frac{1149-1002}{1149}$, where T_{max} and T_{avg} are the maximum and average time workers were busy during the processing of the specific stage.

6. CONCLUSIONS

We illustrate the application of Subset Simulation for Bayesian updating and parameter identification in the context of ABC. We demonstrate this method on the calibration of force fields used in the the discrete element method used for granular flows simulations, as well as in MD simulations. We show that the implementation of the ABC-SubSim method, for both likelihood-free and likelihood-driven Bayesian inference, shares common algorithmic structures with TMCMC despite their different theoretical context. ABC-SubSim provides an alternative algorithm to TMCMC when one has formulated an analytical likelihood function. This was evidence in the granular material application where the final results between the two algorithms were similar. ABC-SubSim ideal usage however remains in the cases where the likelihood function is intractable.

We augmented our open source framework Π4U to include ABC-SubSim. The code is released under the LGPL license and can be downloaded from http://www.cse-lab.ethz.ch/software/Pi4U. We specifically used ABC-SubSim to calibrate MD systems using thermodynamic quantities from a single run. Our large scale parallel runs on a Cray XC-30 machine showed very good efficiency of up to 87%, an outcome of the perfect parallel sampling of the algorithm and its implementation.

We quantified the differences in MD force-field calibration using ABC, by approximating classical Bayesian updating using Gaussian as well as relative entropy discrepancy functions. The ABC formulation is significantly simpler, versatile, and equipped with a relative entropy or other quantile-type discrepancies provides a direct link to the statistical ensemble averaging of thermodynamic QoIs. As such it is suitable and intuitive for force-field calibration and hierarchical coarse graining campaigns.

7. ACKNOWLEDGEMENTS

We acknowledge support by the European Research Council (ERC) Advanced Grant, 34117). This research is implemented under the ARISTEIA Action of the Operational Programme Education and Lifelong Learning and is co-funded by the European Social Fund (ESF) and Greek National Resources. We would like to acknowledge support from ETHZ and computational time from the Swiss supercomputing centre CSCS under project number s448. We would like to acknowledge helpful discussions with Dr. Stephen Wu.

8. REFERENCES

[1] P. Angelikopoulos, C. Papadimitriou, and P. Koumoutsakos. Bayesian uncertainty quantication and propagation in molecular dynamics simulations: a high performance computing framework. *J. Chem. Phys.*, 137(14):144103, 2012.

[2] P. Angelikopoulos, C. Papadimitriou, and P. Koumoutsakos. Data driven, predictive molecular dynamics for nanoscale flow simulations under uncertainty. *J. Phys. Chem. B*, 117(47):14808–14816, 2014/12/01 2013.

[3] P. Angelikopoulos, C. Papadimitriou, and P. Koumoutsakos. X-TMCMC: Adaptive kriging for Bayesian inverse modeling. *Comp. Meth. Appl. Mech. Eng.*, 289:409–428, 6 2015.

[4] S.-K. Au and J. L. Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Prob. Eng. Mech.*, 16(4):263–277, 2001.

[5] M. A. Beaumont. Approximate Bayesian computation in evolution and ecology. *Annu. Rev. Ecol. Evol. Syst.*, 41(1):379–406, 2014/12/01 2010.

[6] J. L. Beck and L. S. Katafygiotis. Updating models and their uncertainties. I: Bayesian statistical framework. *J. Eng. Mech.-ASCE*, 124(4):455–461, April 1998.

[7] F. Cailliez, A. Bourasseau, and P. Pernot. Calibration of forcefields for molecular simulation: Sequential design of computer experiments for building cost-efficient kriging metamodels. *J. Comp. Chem.*, 35(2):130–149, 2014.

[8] F. Cailliez and P. Pernot. Statistical approaches to forcefield calibration and prediction uncertainty in molecular simulation. *J. Chem. Phys.*, 134(5):054124, 2011.

[9] M. Chiachio, J. L. Beck, J. Chiachio, and G. Rus. Approximate Bayesian computation by subset simulation. *SIAM J. Sci. Comput.*, 36(3):A1339–A1358, 2014/10/03 2014.

[10] J. Ching and Y. Chen. Transitional Markov chain Monte Carlo method for Bayesian model updating, model class selection, and model averaging. *J. Eng. Mech.*, 133(7):816–832, 2007.

[11] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *J. R. Stat. Soc. Series B Stat. Methodol.*, 68(3):411–436, 2006.

[12] H. Dong and M. H. Moys. Experimental study of oblique impacts with initial spin. *Powd. Technol.*, 161(1):22–31, 2006.

[13] K. Farrell, J. T. Oden, and D. Faghihi. A bayesian framework for adaptive selection, calibration, and validation of coarse-grained models of atomistic systems. *J. Comput. Phys.*, 295(0):189–208, 8 2015.

[14] A. Gelman, X.-L. Meng, and H. Stern. Posterior predictive assessment of model fitness via realized discrepancies. *Stat. Sinica*, 6:733–807, 1996.

[15] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman & Hall/CRC, London, 1st edition, Dec. 1996.

[16] R. Graham. Bounds on multiprocessing timing anomalies. *SIAM J. App. Math.*, 17(2):416–429, 1969.

[17] P. Hadjidoukas, E. Lappas, and V. Dimakopoulos. A runtime library for platform-independent task parallelism. In *20th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 229–236. IEEE Computer Society, Munich, Germany, 2012.

[18] P. E. Hadjidoukas, P. Angelikopoulos, C. Papadimitriou, and P. Koumoutsakos. Π4U: A high performance computing framework for Bayesian uncertainty quantification of complex models. *J. Comput. Phys.*, 284(1):1–21, 2015.

[19] P. E. Hadjidoukas, P. Angelikopoulos, D. Rossinelli, D. Alexeev, C. Papadimitriou, and P. Koumoutsakos. Bayesian uncertainty quantification and propagation for discrete element simulations of granular materials. *Comp. Meth. Appl. Mech. Eng.*, 282:218–238, 12 2014.

[20] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.

[21] J. S. Lopes, M. Arenas, D. Posada, and M. A. Beaumont. Coestimation of recombination, substitution and molecular adaptation rates by approximate Bayesian computation. *Heredity*, 112(3):255–264, March 2014.

[22] S. Luding. Introduction to discrete element methods : basic of contact force models and how to perform the micro-macro transition to continuum theory. *Eur. J. Env. Civ. Eng.*, 12(7-8):785–826, 2008.

[23] P. Marepalli, J. Y. Murthy, B. Qiu, and X. Ruan. Quantifying uncertainty in multiscale heat conduction calculations. *J. Heat Transfer*, 136(11):111301–111301, 08 2014.

[24] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré. Markov chain Monte Carlo without likelihoods. *Proc. Natl. Ac. Am.*, 100(26):15324–15328, December 2003.

[25] X.-L. Meng. Posterior predictive p-values. *Ann. Stat.*, 22(3):1142–1160, 1994.

[26] 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, June 1953.

[27] C. Peter and K. Kremer. Multiscale simulation of soft matter systems. *Faraday Discuss.*, 144:9–24, 2010.

[28] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.*, 117(1):1–19, March 1995.

[29] J. K. Pritchard, M. T. Seielstad, A. Perez-Lezaun, and M. W. Feldman. Population growth of human y chromosomes: A study of y chromosome microsatellites. *Mol. Biol. Evol.*, 16(12):1791–1798, December 1999.

[30] A. Rau, F. Jaffrezic, J.-L. Foulley, and R. W. Doerge. Reverse engineering gene regulatory networks using approximate Bayesian computation. *Stat. Comput.*, 22(6):1257–1271, November 2012.

[31] F. Rizzi, H. Najm, B. Debusschere, K. Sargsyan, M. Salloum, H. Adalsteinsson, and O. Knio. Uncertainty quantification in MD simulations. part II: Bayesian inference of force-field parameters. *SIAM Multi. Mod. Sim.*, 10(4):1460–1492, 2012.

[32] R. Y. Rubinstein. *Simulation and the Monte Carlo Method*, chapter 4, pages 114–157. New York: Wiley, 1981.

[33] K. Scranton, J. Knape, and P. de Valpine. An approximate Bayesian computation approach to parameter estimation in a stochastic stage-structured population model. *Ecology*, 95(5):1418–1428, May 2014.

[34] M. S. Shell. The relative entropy is fundamental to multiscale and inverse thermodynamic problems. *J. Chem. Phys.*, 129(14):144108, 2008.

[35] W. Shinoda, M. Shiga, and M. Mikami. Rapid estimation of elastic constants by molecular dynamics simulation under constant stress. *Phys. Rev. B*, 69(13), April 2004.

[36] D. Straub and I. Papaioannou. Bayesian updating with structural reliability methods. *J. Eng. Mech.-ASCE*, 141(3):04014134, 2015/06/03 2014.

[37] D. A. Tallmon, G. Luikart, and M. A. Beaumont. Comparative evaluation of a new effective population size estimator based on approximate Bayesian computation. *Genetics*, 167(2):977–988, June 2004.

[38] B. M. Turner and T. Van Zandt. A tutorial on approximate Bayesian computation. *J. Math. Psychol.*, 56(2):69–85, April 2012.

[39] K. Walters. Parameter estimation for an immortal model of colonic stem cell division using approximate Bayesian computation. *J. Theor. Biol.*, 306:104–114, August 2012.

[40] R. D. D. Wilkinson. Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Stat. Appl. Genet. Mol.*, 12(2):129–141, Jan. 2013.

[41] K. M. Zuev, J. L. Beck, S.-K. Au, and L. S. Katafygiotis. Bayesian post-processor and other enhancements of Subset Simulation for estimating failure probabilities in high dimensions. *Computers and Structures*, 92-93:283–296, 2012.