A stochastic optimization approach to coarse-graining using a relative-entropy framework

Ilias Bilionis1,2 and Nicholas Zabaras1,2, a)

1Materials Process Design and Control Laboratory, Sibley School of Mechanical and Aerospace Engineering, 101 Frank H.T. Rhodes Hall, Cornell University, Ithaca, New York 14853-3801, USA
2Center for Applied Mathematics, 657 Frank H.T. Rhodes Hall, Cornell University, Ithaca, New York 14853, USA

(Received 15 November 2012; accepted 9 January 2013; published online 29 January 2013)

Relative entropy has been shown to provide a principled framework for the selection of coarse-grained potentials. Despite the intellectual appeal of it, its application has been limited by the fact that it requires the solution of an optimization problem with noisy gradients. When using deterministic optimization schemes, one is forced to either decrease the noise by adequate sampling or to resolve to ad hoc modifications in order to avoid instabilities. The former increases the computational demand of the method while the latter is of questionable validity. In order to address these issues and make relative entropy widely applicable, we propose alternative schemes for the solution of the optimization problem using stochastic algorithms. © 2013 American Institute of Physics. [http://dx.doi.org/10.1063/1.4789308]

I. INTRODUCTION

Coarse-grained models of complex, many-body systems are useful in at least two ways. On one hand, they can enhance our understanding of physical procedures, in the sense that simple underlying physical mechanisms can be revealed. On the other hand, their evaluation requires much less computational resources than the full model allowing us to investigate problems that would be otherwise non-tractable. They are a necessary component for performing realistic multiscale/multi-physics materials simulations.

The construction of coarse-grained models based on mathematically sound grounds has been the goal of many recent efforts. The most popular techniques are Boltzmann inversion1 and its extensions, such as the iterative Boltzmann inversion,2 inverse Monte Carlo,3,4 and the so-called force-matching methods.5,6 One of the most recent developments is the relative entropy framework.7 In this approach, one defines a pool of candidate coarse-grained potentials and selects the one that minimizes the relative entropy between the induced Boltzmann distribution of the coarse-grained system and the distribution that is based on the potential of the mean force. The intuitive appeal of the relative entropy approach rests on the fact that it is trying to find the coarse-grained potential that best replicates the true global statistics of the coarse-grained system as opposed to optimizing for particular statistics or matching the mean forces. Furthermore, its generality makes it applicable to problems that can potentially go far beyond the realm of traditional approaches. Some connections to other coarse-graining techniques are investigated in Ref. 8. So far, it has been used to find coarse-grained water models,5 study protein energy landscapes,10 and coarse-grained peptide models.11

The greatest obstacle in applying the relative entropy framework to realistic problems is that it requires the solution of a very difficult optimization problem. The difficulty arises from the inescapable presence of noise in the gradients of the optimized quantity (i.e., the relative entropy). These gradients include expectations with respect to the probability density of the coarse-grained system and can only be evaluated using a Monte Carlo procedure. When utilizing deterministic optimization schemes (such as gradient descent or Newton-Raphson), one has to decrease the gradient noise to acceptable levels in order to ensure numerical stability. The only way that this can be achieved is by increasing the sample size of the aforementioned Monte Carlo simulations, making it computationally inefficient for even moderate size systems. The goal of this paper is to restate the optimization problem of the relative entropy method as a stochastic optimization problem. Then, we develop stochastic optimization schemes that can deal with noisy gradients, eliminating the need for long Monte Carlo simulations and making the method computationally attractive. One of the major contributions of this paper is the development of a stochastic optimization scheme that makes an effective use of noisy curvature information.

The paper is organized as follows. In Sec. II, we introduce our methodology. In particular, in Sec. II A, we give the definition of the coarse-graining problem in terms of generalized coordinates and in Sec. II B, we provide a concrete mathematical definition of the relative-entropy principle. Section II C discusses the optimization techniques for the minimization of the relative entropy. A discrete version of the problem along with formulas for the Jacobian and the Hessian of the relative entropy is presented in Sec. II C 1, while in Sec. II C 2, we show that the problem is convex when a linear basis is used. In Sec. II C 3, we discuss the requirements of any optimization scheme and emphasize how noise enters into the picture. Finally, in Secs. II C 4–II C 6, we provide the details of the deterministic Newton-Raphson scheme, the
stochastic Robbins-Monro algorithm, and our modified version that makes use of curvature information, respectively. We demonstrate our approach with two numerical examples: obtaining a mean-field approximation to a lattice gas (Sec. III A) and finding a coarse-grained description of the simple point charge/extended (SPC/E) water model (Sec. III B).

II. METHODOLOGY

A. The general coarse-graining problem

We start by stating the mathematical formulation of the coarse-graining problem using generalized coordinates. Let \( M \subset \mathbb{R}^n \) denote the state space of an atomistic system with \( n \gg 1 \) degrees of freedom and \( q \in M \) the vector of generalized coordinates. Suppose that these coordinates are distributed according to a Boltzmann-like distribution:

\[
p(q) = \frac{\exp \left[-\beta U(q)\right]}{Z(\beta)}, \tag{1}
\]

where \( U(q) \) is the atomistic potential, \( \beta = 1/kT \) (\( T \) is the temperature) and \( Z(\beta) \) is the partition function

\[
Z(\beta) = \int_M \exp \left[-\beta U(q)\right] dq. \tag{2}
\]

Let \( M_{CG} \subset \mathbb{R}^n_{CG} \) be the coarse state space with which we wish to replace the full atomistic system. Any map \( \xi : M \rightarrow M_{CG} \) with \( n_{CG} \leq n \) will be called a coarse-graining map. Using \( \xi \), we can define the coarse generalized coordinates \( Q \in M_{CG} \) corresponding to \( q \in M \) by

\[
Q = \xi(q). \tag{3}
\]

Clearly, the coarse-graining map \( \xi \) is not one-to-one and, in general, multiple \( q \)’s might correspond to the same \( Q \). A very important assumption for the coarse-graining map is that it should be onto

\[
\xi^{-1}(M_{CG}) = M. \tag{4}
\]

This ensures all of the fine scale states are compatible with the coarse-graining map. Selecting the right coarse-graining map is problem dependent and beyond the scope of the current work.

In this work, we assume that a coarse-graining map is given and our goal is to find the probability distribution of the coarse states \( Q \) as induced by Eqs. (1) and (3). In fact, we are looking for a potential energy function \( U_{CG}(Q; \beta) \) such that

\[
p(Q) = \frac{\exp \left[-\beta U_{CG}(Q; \beta)\right]}{Z_{CG}(\beta)}, \tag{5}
\]

where \( Z_{CG}(\beta) \) is the partition function of the coarse system

\[
Z_{CG}(\beta) = \int_{M_{CG}} \exp \left[-\beta U_{CG}(Q; \beta)\right] dQ. \tag{6}
\]

We will refer to \( U_{CG}(\cdot) \) as the effective potential of the coarse-grained system. Let us briefly mention that \( U_{CG}(\cdot) \) is undefined up to an arbitrary additive function of \( \beta \).

There exists an analytic formula for \( U_{CG}(\cdot) \):

\[
U_{CG}(Q; \beta) = -\beta^{-1} \ln \int_M \exp \left[-\beta U(q)\right] \delta(\xi(q) - Q) dq, \tag{7}
\]

where \( \delta(\cdot) \) is the multi-dimensional Dirac \( \delta \) and the integral should be interpreted as being taken over the manifold \( M(q) = \{q \in M : \xi(q) = Q\} \). Equation (7) can be recognized as the free energy with respect to the reaction coordinate \( \xi \).

Let us finish this section by mentioning that in general the coarse-graining potential \( U_{CG}(\cdot) \) is not transferable. That is, it depends on the temperature of the system (or any other parameters that might be present).

B. The relative entropy method

Suppose that the coarse-graining map \( \xi(\cdot) \) is selected and kept fixed. The relative entropy method can be used to find an approximation of \( U_{CG}(Q; \beta) \) at a fixed inverse temperature \( \beta \). For now on, we drop the dependence of \( U_{CG}(\cdot) \) on \( \beta \) for notational convenience. One assumes that the coarse-graining potential may be chosen from a class of candidate functions \( \tilde{U}_{CG} \). Each \( \tilde{U}_{CG}(\cdot) \in \mathcal{U}_{CG} \) is regarded as a candidate coarse-graining potential and the corresponding probability density

\[
p(Q; \tilde{U}_{CG}(\cdot); \beta) \propto \exp \left[-\beta \tilde{U}_{CG}(Q)\right], \tag{8}
\]

as a candidate for the true probability density of the coarse variables, \( p(Q; \beta) \). Intuitively, we wish to select the potential \( \tilde{U}_{CG}(\cdot) \) so that the “distance” between the two densities \( p(Q; \tilde{U}_{CG}(\cdot); \beta) \) and \( p(Q; \beta) \) is minimized. An appropriate “distance measure” for probability density functions is the relative entropy, or the Kullback-Leibler divergence (as is otherwise known)

\[
S_{rel}[\tilde{U}_{CG}(\cdot)] := \left\langle \ln \left( \frac{p(Q; \beta)}{p(Q; \tilde{U}_{CG}(\cdot); \beta)} \right) \right\rangle_{Q; \beta}, \tag{9}
\]

where \( \langle \cdot \rangle_{Q; \beta} \) denotes the expectation with respect to \( p(Q; \beta) \). We can express this quantity in terms of the probability density of \( Q \)

\[
S_{rel}[\tilde{U}_{CG}(\cdot)] = \left\langle \ln \left( \frac{p(Q; \beta)}{p(Q; \tilde{U}_{CG}(\cdot); \beta)} \right) \right\rangle_{Q; \beta} + S_{map}[\xi(\cdot)], \tag{10}
\]

where \( S_{map}[\xi(\cdot)] \) measures the degeneracy of states as induced by the coarse-graining map \( \xi(\cdot) \) and is independent from the choice of the coarse-graining potential \( \tilde{U}_{CG}(\cdot) \).

Remark 1. Since \( S_{map}[\xi(\cdot)] \) is independent of \( \tilde{U}_{CG}(\cdot) \), minimization of the relative entropy is equivalent to the minimization of

\[
\mathcal{F}[\tilde{U}_{CG}(\cdot)] := \left\langle \ln \left( \frac{p(Q; \beta)}{p(Q; \tilde{U}_{CG}(\cdot); \beta)} \right) \right\rangle_{Q; \beta}. \tag{11}
\]

The principle of relative entropy states that the “best” coarse-graining potential from the candidate class \( \mathcal{U}_{CG} \) is given by

\[
U^*_{CG}(\cdot) = \arg \min_{\tilde{U}_{CG}(\cdot) \in \mathcal{U}_{CG}} \mathcal{F}[\tilde{U}_{CG}(\cdot)]. \tag{12}
\]
The minimum always exists, since (by the non-negativity of the Kullback-Leibler divergence)

$$\mathcal{F} \left[ \tilde{U}_{CG} (\cdot) \right] \geq -S_{\text{map}} [ \tilde{\xi} (\cdot) ] , \ \forall \ \tilde{U}_{CG} (\cdot) \in \mathcal{U}_{CG}.$$  

Finally, using the fact that the Kullback-Leibler divergence is zero if and only if the two probability distributions are the same, we obtain that if the true coarse-graining potential $U_{CG} (\cdot) \in \mathcal{U}_{CG}$, then the $U_{CG}^* (\cdot) = U_{CG} (\cdot)$, almost surely.

**Remark 2.** The relative entropy principle can be potentially extended to include the selection of the coarse-graining map $\tilde{\xi} (\cdot)$ within a set of candidate maps. This is beyond the scope of the current paper, but it is an indication that the method is apparently more general than the restricted version presented in the current literature.

### C. Optimization techniques for minimization of the relative entropy

#### 1. General observations

Suppose that the class of candidate potentials $\mathcal{U}_{CG}$ is parametrized by a set of $m$ numbers $\mathbf{y} \in \Gamma \subset \mathbb{R}^m$:

$$\mathcal{U}_{CG} = \{ U_{CG} (\cdot; \mathbf{y}) : \mathbf{y} \in \Gamma \} .$$

The optimization problem of Eq. (10), can now be stated as

$$\mathbf{y}^* = \arg \min_{\mathbf{y} \in \Gamma} \mathcal{F} [ U_{CG} (\cdot; \mathbf{y}) ] . \quad (13)$$

Notice, that $\mathbf{y}^*$ will depend on $\beta$.

For future reference, let us explicitly provide the first and second derivatives of $\mathcal{F} [ U_{CG} (\cdot; \mathbf{y}) ]$ with respect to $\mathbf{y}$. The first derivative is

$$J_i (\mathbf{y}) : = \frac{\partial}{\partial y_i} \mathcal{F} [ U_{CG} (\cdot; \mathbf{y}) ]
= \beta \left( \left[ \frac{\partial}{\partial y_i} U_{CG} (\xi (\mathbf{q}); \mathbf{y}) \right]_{\mathbf{q}, \beta} - \left[ \frac{\partial}{\partial y_i} U_{CG} (\mathbf{Q}; \mathbf{y}) \right]_{\mathbf{q}, \beta} \right) , \quad (14)$$

for $i = 1, \ldots, m$ and where $\langle \cdot \rangle_{\mathbf{q}, \beta}$ denotes the expectation with respect to the candidate probability density for the coarse variables induced by the candidate potential $U_{CG} (\cdot; \mathbf{y})$:

$$p (\mathbf{Q}; \mathbf{y}, \beta) \propto \exp \left(-\beta U_{CG} (\mathbf{Q}; \mathbf{y})\right) .$$

The second derivative is given by

$$H_{ij} (\mathbf{y}) : = \frac{\partial^2}{\partial y_i \partial y_j} \mathcal{F} [ U_{CG} (\cdot; \mathbf{y}) ]
= \beta \left( \left[ \frac{\partial^2}{\partial y_i \partial y_j} U_{CG} (\xi (\mathbf{q}); \mathbf{y}) \right]_{\mathbf{q}, \beta} - \left[ \frac{\partial^2}{\partial y_i \partial y_j} U_{CG} (\mathbf{Q}; \mathbf{y}) \right]_{\mathbf{q}, \beta} \right) + \beta^2 \left( \left[ \frac{\partial}{\partial y_i} U_{CG} (\mathbf{Q}; \mathbf{y}) \right]_{\mathbf{q}, \beta} \left[ \frac{\partial}{\partial y_j} U_{CG} (\mathbf{Q}; \mathbf{y}) \right]_{\mathbf{q}, \beta} \right) + \beta \left( \left[ \frac{\partial}{\partial y_i} U_{CG} (\mathbf{Q}; \mathbf{y}) \right]_{\mathbf{q}, \beta} \left[ \frac{\partial}{\partial y_j} U_{CG} (\mathbf{Q}; \mathbf{y}) \right]_{\mathbf{q}, \beta} \right) , \quad (15)$$

for $i, j = 1, \ldots, m$. Notice that the second part of the right hand side is the $ij$th element of the covariance matrix $\text{C} (\mathbf{y}) = (C_{ij} (\mathbf{y}))$:

$$C_{ij} (\mathbf{y}) : = \text{Cov}_{\mathbf{y}, \beta} \left[ \frac{\partial U_{CG} (\mathbf{Q}; \mathbf{y})}{\partial y_i} , \frac{\partial U_{CG} (\mathbf{Q}; \mathbf{y})}{\partial y_j} \right]$$

$$= \left( \frac{\partial U_{CG} (\mathbf{Q}; \mathbf{y})}{\partial y_i} \right)_{\mathbf{q}, \beta} \left( \frac{\partial U_{CG} (\mathbf{Q}; \mathbf{y})}{\partial y_j} \right)_{\mathbf{q}, \beta} . \quad (16)$$

#### 2. Choice of the representation and convexity of the optimization problem

A very common choice of $\mathcal{U}_{CG}$ is

$$\mathcal{U}_{CG} = \left\{ \sum_{i=1}^{m} \gamma_i \phi_i (\cdot) + \phi_0 (\cdot) : \mathbf{y} \in \Gamma \right\} . \quad (17)$$

where $\phi_i (\cdot), i = 1, \ldots, m$ is a finite set of basis functions. The function $\phi (\cdot)$ is not a basis function. It is defined by the particular problem of interest and its purpose is to allow the modeling of non-canonical distributions, e.g., in a Ising model in a grand canonical ensemble it could be given by the chemical potential times the particle number. In all our numerical examples, this is how we choose to represent $\mathcal{U}_{CG}$. From now on, we assume that $\mathcal{U}_{CG}$ is of the form given in Eq. (17).

Notice that the Hessian matrix becomes (the first terms of Eq. (15) disappear because they involve second derivatives with respect to $\mathbf{y}$)

$$H (\mathbf{y}) = \beta^2 \text{C} (\mathbf{y}) ,$$

which, in general, is positive-semidefinite and symmetric. Hence, if $\Gamma$ is convex, the optimization problem defined in Eq. (13) is convex. Furthermore, if the Hessian is positive-definite, then the optimization problem is strictly convex and possesses a unique solution.

#### 3. On the requirements of any optimization method

Any optimization method would have to estimate the Jacobian and optionally the Hessian of $\mathcal{F} [ U_{CG} (\cdot; \mathbf{y}) ]$. This requires two kinds of expectations:

1. $\langle \cdot \rangle_{\mathbf{q}, \beta}$ with respect to the fine scale probability density. Sampling $\mathbf{q}$ from $p (\mathbf{q}; \beta)$ does not depend on the parameters of the coarse potential $\mathbf{y}$. Therefore, in order to evaluate these expectations we can simply store the results of a very long fine scale simulation and use the same samples every time we need to evaluate an expectation.

2. $\langle \cdot \rangle_{\mathbf{q}, \beta}$ with respect to the candidate coarse scale probability density. Sampling $\mathbf{Q}$ from $p (\mathbf{Q}; \mathbf{y}, \beta)$ depends on the parameters of the coarse potential $\mathbf{y}$. Therefore, in order to approximate the expectations, samples from $p (\mathbf{Q}; \mathbf{y}, \beta)$ must be drawn anew every time $\mathbf{y}$ changes.

This is the approach we will follow in this paper. To make the ideas concrete, let us denote the fine scale simulation by

$$D_{\text{fine}} = \{ \mathbf{q}^{(1)} \}_{s=1}^{S_{\text{fine}}} , \quad (18)$$
where $S_{\text{fine}}$ is the total number of observed fine scale samples. Note that—typically—this is different than the actual number of fine scale samples that have been taken, which is

$$S_{\text{fine,actual}} = S_{\text{fine,eq}} + S_{\text{fine,burn}}S_{\text{fine}},$$

where $S_{\text{fine,eq}}$ is the number of equilibration steps and $S_{\text{fine,burn}}$ is the number of MD steps burnt in order to ensure that the final observations are sufficiently decorrelated. That is the production period consists of a total of $S_{\text{fine,burn}}S_{\text{fine}}$ steps. In order to evaluate the Jacobian and the Hessian at a particular $y \in \Gamma$, we need samples from the coarse-grained model for that parameter

$$D_{\text{CG},y} = \{Q_y^{(s)}\}_{s=1}^{S_{\text{CG}}},$$

where $S_{\text{CG}}$ is the total number of observed coarse-grained samples. Again, the actual number of samples is

$$S_{\text{CG,actual}} = S_{\text{CG,eq}} + S_{\text{CG,burn}}S_{\text{CG}},$$

where $S_{\text{CG,eq}}$ is the number of equilibration steps and $S_{\text{CG,burn}}$ the number of burnt MD steps ($S_{\text{CG,burn}}S_{\text{CG}}$ production steps). The Jacobian at a particular $y$ can be approximated by

$$\hat{J}_i(y) = \beta \left( \frac{1}{S_{\text{fine}}} \sum_{s=1}^{S_{\text{fine}}} \phi_i (\xi (q^{(s)})) - \frac{1}{S_{\text{CG}}} \sum_{s=1}^{S_{\text{CG}}} \phi_i (Q_y^{(s)}) \right),$$

for $i = 1, \ldots, m$ and the Hessian by

$$\hat{H}_{ij}(y) = \beta^2 \left( \frac{1}{S_{\text{CG}}} \sum_{s=1}^{S_{\text{CG}}} \phi_i (Q_y^{(s)}) \phi_j (Q_y^{(s)}) \right) - \left( \frac{1}{S_{\text{CG}}} \sum_{s=1}^{S_{\text{CG}}} \phi_i (Q_y^{(s)}) \right) \left( \frac{1}{S_{\text{CG}}} \sum_{s=1}^{S_{\text{CG}}} \phi_j (Q_y^{(s)}) \right),$$

for $i, j = 1, \ldots, m$. Since, we will be using a single fine scale simulation, the approximate Jacobian will always have a small bias. Assuming that the fine scale simulation is sufficiently long, this bias is ignored in this paper. Accurate evaluation of the Jacobian and the Hessian requires a satisfactorily long coarse-grained simulation for every iteration step. This is exactly why classical schemes (e.g., Newton-Raphson) are extremely difficult to apply to large-scale problems. Our aim is to investigate the performance of stochastic optimization algorithms that can deal with noisy gradients (i.e., smaller $S_{\text{CG,actual}}$).

### 4. Newton-Raphson optimization

Reference 8 suggests solving the optimization problem defined in Eq. (13) using a Newton-Raphson (NR) approach.\(^\text{12}\) We start from some initial guess $y^0$ and update it via

$$y^{k+1} = y^k - \chi \hat{H}^{-1}(y^k)\hat{J}(y^k),$$

where $0 < \chi \leq 1$ is a mixing coefficient that is adjusted to assure stability of the algorithm. In our numerical examples, we observe that when the Jacobian and the Hessian are not accurate (that is when $S_{\text{CG,actual}}$ is low), then the algorithm fails to converge. For very small $S_{\text{CG,actual}}$, the approximation to the Hessian stops being non-singular. We observe, that in the presence of noise, the algorithm might not even be able to complete one iteration step.

### 5. Robbins-Monro stochastic optimization

The stability of the approach described above, requires that the Jacobian $J$ and—most importantly—the Hessian $H$ are calculated accurately. This requires that at each step of the algorithm a sufficiently long CG simulation is carried out, which may be extremely time consuming. A smaller number of CG steps ($S_{\text{CG,actual}}$ small) maybe used, if we employ a stochastic optimization algorithm that can work with noisy gradients. In this section, we present the simplest such algorithm, the well-known Robbin-Monro\(^\text{13}\) stochastic optimization scheme (RM).

Starting from an initial guess $y^0$, the procedure is as follows:

$$y^{k+1} = y^k - \alpha_k \hat{J}(y^k),$$

where $\{\alpha_k\}_{k \in \mathbb{R}}$ is a sequence of real numbers such that

$$\sum_{k=1}^{\infty} \alpha_k = +\infty \text{ and } \sum_{k=1}^{\infty} \alpha_k^2 < \infty.$$

Ignoring the small bias of the Jacobian due to the single fine scale simulation and since the objective function is convex, the algorithm theoretically converges as $k \to \infty$ independently of the amount of noise present.\(^\text{14}\) We choose $\alpha_k$ as

$$\alpha_k = \frac{\alpha}{(A + k)^\rho}$$

with $0 < \rho \leq 1$. There exist theoretical results (see Ref. 14) that prove that the optimum convergence rate is obtained for $\rho = 1$. However, these results are valid for $k \to \infty$. For finite number of iterations, we find that a $\rho < 1$ performs actually better.

### 6. Modified Robbins-Monro

The RM algorithm is based solely on gradient information. In order to avoid instabilities, the learning sequence must consist of relatively moderate steps. Therefore, when the optimization scheme finds itself in flat regions of the relative entropy landscape, it is expected to move very slowly. In order to accelerate the convergence rate, curvature information has to be used. This is exactly what NR does, albeit it requires an accurate Hessian. According to the theory behind the NR method, the optimal step should be proportional to $\hat{H}^{-1}(y^k)\hat{J}(y^k)$. The instabilities occur during the solution of the linear system

$$\hat{H}(y^k)x = \hat{J}(y^k).$$

e.g., using the Cholesky decomposition of $\hat{H}(y^k)$.

Reference 15 proposed a stochastic version of NR in which information about the optimal step is extracted by performing a few Conjugate-Gradient iterations for the solution of the system defined in Eq. (26). To be precise, one performs $t \geq 1$ Conjugate-Gradient iterations for the solution of

$$\hat{H}(y^k)x = \hat{J}(y^k).$$
Eq. (26) starting from a zero initial guess. The result, \( \mathbf{p}^k \) is used as an approximation to the NR optimal step. It is shown in this reference, that a few Conjugate-Gradient iterations are enough in order to extract valuable information about the curvature. On the other hand, performing too many Conjugate-Gradient iterations would lead to the same instability issues as NR. Therefore, the number of Conjugate-Gradient iterations \( t \) is a parameter that depends on the noise of the system.

Inspired by the above-mentioned work, we propose a modification to RM, which we term Modified Robbins-Monro (MRM). The resulting method is essentially a hybrid of NR and RM. Starting from an initial guess \( \mathbf{p}^0 \), the procedure is as follows:

\[
\mathbf{p}^{k+1} = \mathbf{p}^k - \alpha_k \mathbf{p}^k,
\]

where \( \alpha_k \) is a learning sequence as in Eq. (25) and \( \mathbf{p}^k \) is the result of \( t \geq 1 \) Conjugate-Gradient iterations for the solution of the linear system of Eq. (26) starting with a zero initial vector. The choice of, the optimal \( t \) remains an open problem. However, we numerically observe that the method is stable for a wide range of \( t \) choices while enjoying accelerated convergence rates compared to plain RM.

### III. NUMERICAL EXAMPLES

#### A. Lattice gas

We address the problem of generating a mean-field approximation to the 2D lattice gas. We choose this simplistic example because (1) the relative entropy optimization problem has an analytic solution for this case and (2) it is computationally cheap. This enables us to systematically test the convergence of the proposed stochastic schemes and compare their performance.

In this problem, what is coarse-grained is interactions and not degrees of freedom. Let \( n = 16^2 \) be the total number of lattice sites each with \( q_i \in \{0, 1\} \) molecules for \( i = 1, \ldots, n \). The degrees of freedom are collectively denoted by \( \mathbf{q} = (q_1, \ldots, q_n) \). To make the connection with our theoretical framework apparent, the fine-scale state space of this problem is

\[
\mathcal{M} = \{0, 1\}^n.
\]

The fine-scale model interacts via

\[
V(\mathbf{q}) = -\sum_{\langle i, j \rangle} q_i q_j,
\]

where \( \langle i, j \rangle \) is interpreted as summation over nearest neighbours. We work in a grand canonical ensemble with chemical potential \( \mu_c \) and inverse temperature \( \beta \). Therefore, the probability density of \( \mathbf{q} \) is given by

\[
p(\mathbf{q} | \beta, \mu_c) \propto \exp \left\{ -\beta \left( V(\mathbf{q}) - \mu_c \sum_{i=1}^n q_i \right) \right\}.
\]

Since, we are not coarse-graining degrees of freedom, the coarse-grained state space is the same as the fine-scale one

\[
\mathcal{M}_{CG} = \mathcal{M},
\]

and the coarse-graining map \( \xi : \mathcal{M} \to \mathcal{M}_{CG} \) is the identity map

\[
\xi(\mathbf{q}) = \mathbf{q}.
\]

The coarse-grained variables are therefore exactly the same as the fine-scale ones, i.e., \( \mathbf{Q} = \xi(\mathbf{q}) = \mathbf{q} \). The coarse-grained probability distribution is assumed to be

\[
p(\mathbf{Q} | \gamma, \beta, \mu_c) \propto \exp \left\{ -\beta \left( V_{CG}(\mathbf{Q}) - \mu_c \sum_{i=1}^n q_i \right) \right\},
\]

where

\[
V_{CG}(\mathbf{Q}) = -\gamma \sum_{i=1}^n Q_i,
\]

where \( \gamma \) is a missing parameter. In order to see that the Boltzmann-like distributions used in Sec. II are not really restrictive, simply define the effective fine-scale potential energy functions. To make apparent the equivalence to Eq. (17), we are actually using a single basis function \( m = 1 \), defined by

\[
\phi_1(\mathbf{Q}) = -\sum_{i=1}^n Q_i,
\]

while

\[
\phi_0(\mathbf{Q}) = -\mu_c \sum_{i=1}^n Q_i.
\]

It can be shown that the parameter minimizing the relative entropy between \( p(\mathbf{q} | \beta, \mu_c) \) and \( p(\mathbf{Q} | \gamma, \beta, \mu_c) \) is

\[
\gamma^* = -\beta^{-1} \log \left[ \sum_{i=1}^n \exp \left( \frac{\beta q_i}{\mu_c} \right) \right] - \mu_c.
\]

Therefore, we can use \( \gamma^* \) to test the convergence of various optimization schemes. Samples from \( p(\mathbf{q} | \beta, \mu_c) \) and \( p(\mathbf{Q} | \gamma, \beta, \mu_c) \) can be easily taken using a Markov Chain Monte Carlo (MCMC) procedure with a single move proposal.

In all simulations (either fine-scale or coarse-grained for a given \( \gamma \)), we start from a random state (i.e., at each lattice site we pick 0 or 1 with probability 0.5), we perform 2000 equilibration steps (i.e., \( S_{\text{fine,eq}} = 2000 \) and \( S_{\text{CG,eq}} = 2000 \)) and then observe the statistics of the system every 20 MCMC moves (i.e., \( S_{\text{fine,burn}} = 20 \) and \( S_{\text{CG,burn}} = 20 \)). We perform a single fine scale simulation with \( 10^6 \) MCMC steps (i.e., \( S_{\text{fine,actual}} = 10^6 \) and \( S_{\text{fine}} = 49,900 \)), which is used to calculate the part of the Jacobian that depends on the fine scale probability density (see Eq. (20)). We compare the various
algorithm fails to perform a single iteration if $S_{CG,actual} < 2878$. The reason this happens, is that the very noisy Hessian makes the system overshoot the value of $\gamma$. To understand this behavior, imagine that as the Jacobian gets closer to zero, noise effects become increasingly important to the point that even the sign of the Jacobian is wrong. To make matters even worse, the value of the Hessian is also random. The result is of course a totally random update of $\gamma$. The same phenomenon occurs for $\chi = 0.5$ for $S_{CG,actual} < 4021$. This is natural since, the bigger step makes the effect of the noise in the Hessian appear earlier. For all cases, we observe that the average RMSE decreases until it reaches a limiting value and then fluctuates about it. This limiting value characterizes the accuracy of the NR scheme and it is—quite naturally—a decreasing function of $S_{CG,actual}$ (i.e., the accuracy of NR increases with increasing $S_{CG,actual}$). Finally, observe that the convergence rate of NR—measured with respect to the total number CG MCMC steps—is actually increasing with decreasing $S_{CG,actual}$. Of course, up to the point where the algorithm fails. This is also expected, since by decreasing $S_{CG,actual}$, one can perform more iterations with the same number of CG MCMC steps.

Next, we present our results for the RM scheme. This scheme has theoretical convergence guarantees. However, its convergence rate depends on the choice of the parameters of the learning sequence ($\alpha_i$) given in Eq. (25). We numerically observed that when $A$ is picked to be $10\%$ of the maximum allowed iterations ($k_{max}$), the convergence rate is insensitive to the choice of $\alpha$ (we tested $\alpha = 0.0001, 0.001, 0.01, \text{and } 0.1$). Therefore, we only report our results for $\alpha = 0.1$. It depends, however, on the choice of $\rho$. Figures 2(a)–2(d) report the results for RM with $\alpha = 0.1$ and $\rho = 0.501, 0.601, 0.701, \text{and } 0.901, \text{respectively. In all cases, we observe an initial part in which the convergence rate is independent of } S_{CG,actual} \text{ and a later part in which it decreases as } S_{CG,actual} \text{ decreases. Furthermore, the overall convergence rate increases as } \rho \text{ is increased. The remarkable observation here is that—contrary to}

\begin{enumerate}
  \item We fix the number of coarse-grained simulations $S_{CG}$, which are used to approximate the averages in Eqs. (20) and (21). The smaller this number, the more noisy the approximation to the gradient is. Keep in mind that the actual number of coarse-grained steps per iteration of the optimization scheme is $S_{CG,actual} = S_{CG,eq} + S_{CG,burn,S_{CG}}$.
  \item Then, we perform as many iterations of the algorithm as required in order to reach a total of approximately $10^6$ actual number of coarse-grained steps. That is, we perform a total of $k_{max}$ iterations given by $k_{max} = 10^6/S_{CG,actual}$.
  \item At each iteration $k$, we record the root mean square error (RMSE)
    \[
    \text{RMSE}_k = \left( \frac{1}{m} \sum_{i=1}^{m} (\gamma_i - \gamma_i^*)^2 \right)^{1/2}.
    \]
    Of course, for the lattice gas problem it is simply $\text{RMSE}_k = |\gamma_i - \gamma_i^*|$.
\end{enumerate}

For each algorithm, we repeat the above procedure 100 times and report the evolution of the average RMSE as a function of the total number coarse-grained MCMC steps. For all schemes, the initial value of $\gamma$ is selected to be the minimum of the Gaussian approximation to the relative-entropy (see Ref. 8 for the details). The whole process is repeated for $S_{CG,actual} = 2243, 2878, 4021, \text{and } 6687$ and for all the NR and RM optimization schemes. To avoid cluttering the legends, in the figures we write $S$ instead of $S_{CG,actual}$. Results are reported for inverse temperature $\beta = 1.6$ and chemical potential $\mu_c = -2.5$. The behavior of the optimization algorithms for different choices of $\beta$ and $\mu_c$ was observed to be qualitatively similar and is therefore not reported (the range of values tested was $0.2 \leq \beta \leq 1.8$ and $-3 \leq \mu_c \leq -1.5$).

The results for NR with $\chi = 0.1$ and 0.5 are shown in Figs. 1(a) and 1(b), respectively. For the case of $\chi = 0.1$, the algorithm fails to perform a single iteration if $S < 2878$ samples per iteration are used. For $\chi = 0.5$, it fails for $S < 4021$.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Lattice gas: Newton-Raphson optimization. (a) and (b) show the evolution of the average RMSE over 100 runs as a function of the total number of CG samples for NR with step size $\chi = 0.1$ and 0.5, respectively. The different lines correspond to different choices of samples per iteration. For $\chi = 0.1$, the algorithm fails to complete a single iteration if $S < 2878$ samples per iteration are used. For $\chi = 0.5$, it fails for $S < 4021$.}
\end{figure}
FIG. 2. Lattice gas: Robbins-Monro optimization. (a)–(d) show the evolution of the average RMSE over 100 runs as a function of the total number of CG samples for RM with step size $\alpha = 0.1$ and $\rho = 0.501, 0.601, 0.701$, and $0.901$, respectively. The different lines correspond to different choices of samples per iteration.

NR—the algorithm never stops improving the average RMSE as the number of samples increases. This is because as the Jacobian moves closer to zero, noise is correctly averaged out by the decreasing learning rate.

Finally, we present our results for the MRM scheme. Since, this is a one-dimensional optimization, the method essentially becomes a stabilized version of NR. To see this, notice that in Eq. (26) all quantities involved are simply scalars. Therefore, just one iteration ($t = 1$) of the conjugate gradients algorithm, is enough to solve it exactly. As a result, the updates $p_k$ involved in Eq. (27) become the same as the NR step, i.e.,

$$p_k = \hat{H}^{-1}(\gamma_k)\hat{J}(\gamma_k).$$

So, in this example we set $t = 1$ for all our runs. The role of the learning sequence $\alpha_k$ is to stabilize these updates. As we are approaching the minimum, noise plays gradually a more important role. Therefore, a decreasing $\alpha_k$ ensures that the effects of noise are kept under control. Again, we choose $A$ to be 10% of the maximum allowed iterations and we observed that the convergence rate is independent of the choice of $\alpha$ (we tested $\alpha = 0.01, 0.1$, and $0.5$). So, we report our results only for $\alpha = 0.1$. Figures 3(a)–3(d) report the results for MRM with $\alpha = 0.1$ and $\rho = 0.501, 0.601, 0.701$, and $0.901$, respectively. A general observation is that the convergence rate is decreased as $\rho$ increases. Remarkably, the convergence is initially faster for lower sample sizes per iteration, while—contrary to NR—the algorithm never stops improving its estimate. However, the convergence rate observed for these parameters is not as good as for RM. We will see in the next example that this situation dramatically changes in high-dimensional settings in which MRM clearly outperforms RM for all the parameters examined. Nevertheless, the results are still better than the ones obtain using the NR scheme.

B. SPC/E water

In this application, we attempt to coarse grain the simple point charge/extended (SPC/E)\textsuperscript{16,17} water model. The full details of the fine scale model are exactly the same the ones considered by Ref. \textsuperscript{18}. The reader may want to consult this work for a comparison of our results with those of other coarse-graining techniques. The parameters of the model are $\sigma = 3.166$ Å, $\epsilon = 0.650$ kJmol$^{-1}$ (Lennard-Jones part), $l_{OH} = 1.0$ Å (fixed distance of O-H bond), $q_H = +0.4238e$, $q_O = -2.137e$. 

![Diagram](image-url)
FIG. 3. Lattice gas: Modified Robbins-Monro optimization. (a)–(d) show the evolution of the average RMSE over 100 runs as a function of the total number of CG samples for MRM with step size $\alpha = 0.1$, $A$ being 10% of the maximum number of iterations and $\rho = 0.501, 0.601, 0.701,$ and 0.901, respectively. The different lines correspond to different choices of samples per iteration.

$q_O = -0.8476e$ (charges for H and O, respectively) and $\theta_{\text{HH}} = 109.47^\circ$ (angle between the two OH bonds). We consider a total of $N_w = 2180$ water molecules in the NVT ensemble at a temperature $T = 300$ K. The fine scale system is simulated using Gromacs. The simulation took approximately 5 h using 240 computing cores. The details are as follows: (1) First, the system is equilibrated in the NPT ensemble at 300 K and 1 bar for 100 ns using the Berendsen thermostat and barostat; (2) The last 80 ns are used to find the equilibrium box size (found to be 4.031 nm); (3) Finally, we fix the box size and simulate the system in the NVT ensemble for 45 ns using a stochastic algorithm with a time step of 0.002 ps; (4) For the calculation of averages, we use only the last 40 ns observing the trajectory every 200 steps. This results in a total of $10^2$ observations. The total number of fine-scale degrees of freedom is $n = 9N_w = 19620$ (3 degrees of freedom for each atom of each water molecule, i.e., $\mathcal{M} \equiv \mathbb{R}^n$). Let $q_{O,i}$, $q_{H1,i}$, and $q_{H2,i}$ in $\mathbb{R}^3$, be the spatial positions of the oxygen and the two hydrogen atoms, respectively, of the $i$th water molecule. Collectively, we write

$q = (q_{O,1}, q_{H1,1}, q_{H2,1}, \ldots, q_{O,N_w}, q_{H1,N_w}, q_{H2,N_w}).$

The coarse-grained model has $n_{CG} = 3N_w = 6540$ degrees of freedom of the coarse-grained model (i.e., $\mathcal{M}_{CG} = \mathbb{R}^{n_{CG}}$), since we replace the water molecules with their center of mass $Q_i$, given by

$$Q_i \equiv \xi_i(q) := \frac{m_O q_{O,i} + m_H q_{H1,i} + m_H q_{H2,i}}{m_O + 2m_H},$$

for $i = 1, \ldots, N_w$, where $m_O = 15.999$ u and $m_H = 1.008$ u (“u” stands for “atomic mass units”) are the masses of the oxygen and hydrogen atoms, respectively. This completely defines the coarse-graining map $\xi : \mathcal{M} \rightarrow \mathcal{M}_{CG}$, by setting

$$\xi(q) = (\xi_1(q), \ldots, \xi_{N_w}(q)).$$

Our goal is to find the coarse-grained potential that minimizes the relative entropy.

For computational convenience, we restrict the class of candidate potentials $\mathcal{U}_{CG}$ to pairwise interactions. That is, we assume that the coarsened molecules interact via a pairwise potential that depends only on the distance between them.
where \( r_{\text{min}} \) is the minimum allowed distance, \( r_{\text{max}} \) is the cut-off distance, \( \psi_i(r) \) is a set of basis functions, and \( \gamma \) the coefficients we wish to learn. Given this representation of the pairwise interaction, the candidate coarse-grained potential may be written as

\[
U_{\text{CG}}(\mathbf{Q}; \gamma) = \sum_{j < k} V(\| \mathbf{Q}_j - \mathbf{Q}_k \|; \gamma) \quad \text{(31)}
\]

Notice that

\[
U_{\text{CG}}(\mathbf{Q}; \gamma) = \sum_{j < k} V(\| \mathbf{Q}_j - \mathbf{Q}_k \|; \gamma)
\]

\[
= \sum_{j < k} \sum_{i=1}^{m} \gamma_i \psi_i(\| \mathbf{Q}_j - \mathbf{Q}_k \|)
\]

\[
= m \sum_{i=1}^{m} \gamma_i \sum_{j < k} \psi_i(\| \mathbf{Q}_j - \mathbf{Q}_k \|)
\]

\[
= m \sum_{i=1}^{m} \gamma_i \phi_i(\mathbf{Q})
\]

with

\[
\phi_i(\mathbf{Q}) = \sum_{j < k} \psi_i(\| \mathbf{Q}_j - \mathbf{Q}_k \|). \quad \text{(32)}
\]

This is exactly the representation of Eq. (17) with \( \phi_i(\mathbf{Q}) = 0 \).

Remark 3. It is a well-known fact that the region \( r < r_{\text{min}} \) needs to be taken care explicitly. In this work, we extrapolate the potential to this region using an exponential of the form \( f(r) = a_0 e^{a_1 \chi} \). The coefficients \( a_0 \) and \( a_1 \) are found by matching the value of the potential as well as its derivative at \( r = r_{\text{min}} \).

In this work, we use a cubic splines basis\(^{22}\) with 38 equidistant nodes defined between \( r_{\text{min}} = 0.22 \) nm and \( r_{\text{max}} = 1 \) nm, as implemented in GNU Scientific Library (GSL).\(^{23}\) This results in a total of \( m = 39 \) parameters to be estimated. For a given \( \gamma \), the coarse grained system is simulated as follows: (1) The initial state is randomly picked from the 40 ns of the fine scale trajectory; (2) The potential cut-off is set to \( r_{\text{max}} = 1 \) nm (i.e., it is implicitly assumed that the pairwise potential is zero for \( r > r_{\text{max}} \)); (3) The box size is fixed to the same value as for the fine scale simulation (4.031 nm); (4) The same stochastic algorithm as in the fine scale trajectory is used with the same time step of 0.002 ps; (5) The initial 1.6 ps of the simulation are dropped, while the rest are observed every 50 time steps (this choice was made by observing the convergence of the energy and was found to be satisfactory when initial states are selected as in (1)). The initial \( \gamma \) is selected by fitting the spline basis to a Lennard-Jones potential

\[
V_{\text{LJ}}(r) = \frac{A_{\text{LJ}}}{r^6} - \frac{B_{\text{LJ}}}{r^2}
\]

with \( A_{\text{LJ}} = 1.0334 \times 10^6 \) and \( B_{\text{LJ}} = 2.39043 \times 10^3 \), using least squares. The values of \( A_{\text{LJ}} \) and \( B_{\text{LJ}} \) were found by minimizing the Gaussian approximation to the relative entropy (see Ref. 8). This minimization results in a simple two-dimensional linear system. We assess the convergence of all methods in two ways: (1) The normalized norm of the Jacobian defined by

\[
\text{NNJ}(k) = \frac{\hat{J}(\gamma^k)}{J(\gamma^0)} \quad \text{(33)}
\]

(2) The RMSE as defined in Eq. (29). These quantities are plotted as a function of the total coarse-grained simulation time for up to 1 ns. The “S” parameter given in all figures corresponds to the total number of samples per iteration. This corresponds to \( S = 50 \times 0.002 \) ps of simulation time per iteration. Therefore, for \( i \geq 1 \) iterations the “total CG simulation time” is \( i \times S = 50 \times 0.002 \) ps. Each simulation took—on average—one hour on an 8-core node (two quad-core Intel Xeon X5550 (“Nehalem”) 2.67 GHz processors). The reference \( \gamma^* \) is found by performing 30 iterations of NR with \( \chi = 0.5 \) using coarse-grained runs of 1.2 ns. This choice is more than enough to ensure noise-free gradients. Because of the computational complexity of the problem, we were able to run only a single simulation per method per choice of parameters.

Figure 4 shows the evolution of the normalized Jacobian norm and Fig. 5 the RMSE as a function of the total simulation time for the NR scheme. We pick \( \chi = 0.01, 0.1, \) and 0.5 for \( S = 128 \) and 196 samples per iteration. These samples correspond to 12.8 and 19.6 ps of coarse-grained simulation per iteration, respectively. Each line corresponds to a particular choice of \( \chi \) and \( S \). The line is stopped the first time the Hessian matrix is singular and cannot be inverted or when \( \gamma \) update results in a completely random large step. To understand the reason why the Hessian estimator might become singular, one has to consider the nature of the spline basis functions. Each one of them has a finite support over the range of \( r \). Therefore, the first such basis function (i.e., the one that includes the left cut-off \( r_{\text{min}} \) has a high probability of never being observed having a positive value when the
sample size is too low. This is because there is a correspondingly low probability of observing two water molecules being very close together. The effect of this is a zero row and column in the Hessian. Since, the Hessian is actually stochastic, the iteration on which the first singularity appears is also random. The second kind of instability occurs when the Jacobian gets close to zero and it is practically random. This in combination with the noise of the Hessian might lead to a completely random large step.

Next, Fig. 6 shows the evolution of the normalized Jacobian norm and Fig. 7 the RMSE as a function of the total simulation time for the RM scheme. It is necessary that the potential supplied to the MD software is not very noisy. If it is, then the forces vary wildly and lead to numerical instabilities. For a scheme such as RM that relies only on the Jacobian, this means that the learning rate $\alpha$ must be kept small enough so that updates of $\gamma$ do not introduce very large variations in the derivative of the potential. We find numerically that a choice of $\alpha = 0.01$ is the maximum learning rate that does not introduce such instabilities. Since we expect to run just a few hundred iterations, the $A$ parameter of the learning sequence is set to zero and $\rho$ is set to 0.51, i.e., close to the infimum of its allowed values. Greater $\rho$ leads actually to slower convergence rates for the range of iterations considered here. The figures show the results of runs using $S = 64, 128,$ and 196 samples per iteration. We observe that if all other parameters are kept fixed, a lower $S$ leads to improved convergence. This is, of course, because we are able to perform more iterations with the same number of observations. Notice that compared to NR the algorithm has no stability problems. Even though, the Jacobian converges to zero relatively fast, the RMSE converges very slowly. This is more clearly seen in Fig. 8, which shows the predicted potential at various iteration steps for the case of $S = 64$ samples per iteration. The potential takes the right shape in the first few iterations capturing correctly the two distinct wells that are present. From that point on, the left well is indeed moving up, albeit at a very slow rate. We must mention at this point that this phenomenon is not due to the decreasing nature of the learning sequence $\alpha$. We numerically discovered this fact, by restarting the algorithm at the final iteration (i.e., by re-initializing the learning sequence).
The slow-down is observed because the Jacobian gets very close to zero (i.e., by reaching a relatively flat region of the relative entropy landscape). RM is mathematically guaranteed to escape this region, but does so very slowly. One has to use some information about the curvature in order to accelerate the convergence. It was exactly this insight that motivated us to develop MRM.

Figure 9 shows the evolution of the normalized Jacobian and Fig. 10 the RMSE as a function of the total simulated time for the MRM scheme. We are mainly interested in investigating the performance of the algorithm as the number of conjugate-gradient iterations \( t \) changes keeping the other parameters the same. So, we pick \( \rho = 0.51, \alpha = 0.5, \) and \( A = 0 \) while \( t \) takes the values 10, 20, and 30. We run the scheme for \( S = 128 \) and 196 samples per iteration. Again, no stability problems are observed while—compared to RM—the convergence is clearly improved significantly. It is evident that curvature information is indeed essential. Figure 11 shows the predicted potential at various iteration steps for the case of \( t = 30 \) and \( S = 128 \). We observe, that the scheme quickly identifies the two wells as well their depths. Initially, there is some noise which is gradually smoothed out. It is remarkable, that even after 10 iterations (i.e., 128 ps of coarse-grained simulations), the predicted potential looks very similar to the reference one. The algorithm shows accelerated convergence similar to NR, while being free of its instabilities due to each RM-like averaging. Generally, speaking the bigger \( t \) is, the greater the acceleration we observe. One can clearly see this by observing that the RMSE lines in Fig. 10 are grouped according to \( t \). The theoretical upper limit of \( t \) for this problem is 39 (the dimension of \( \mathbf{\gamma} \)). This is because after 39 conjugate-gradient iterations the linear system of Eq. (26) is solved exactly. However, setting \( t \) to be exactly equal to 39 leads to problems when the Hessian is singular. The reason it works for finite \( t \) is because the \( t \) conjugate-gradient iterations for the solution of Eq. (26) provide an approximation to the optimal step in a \( t \)-dimensional Krylov space. Therefore, even if the Hessian is rank-deficient the method can make an effective use of the curvature information. Thinking in these terms, one might postulate that the optimal choice of \( t \) is directly connected to the rank of the Hessian. Giving a positive answer to this claim as well as developing effective criteria for the choice of the learning sequence \( \alpha_k \) and the optimal sample size per iteration \( S \) requires further numerical and theoretical investigation of MRM.

IV. CONCLUSIONS

We believe that the relative-entropy principle is a new way of thinking that is going to revolutionize the way research on multi-scale systems is carried out. The reason is simply that it provides a principled way of coarse-graining degrees of freedom on arbitrary systems. In order to make it available to practitioners, we need to develop efficient stochastic optimization schemes. This work is a step towards this goal. We introduced a concrete mathematical definition of the relative-entropy principle for the identification of coarse-grained potentials. We demonstrated how the optimization problem can be discretized and—using the idea of an affine basis—we
emphasized the generality of the method and its applicability beyond canonical ensembles. We extensively discussed the numerical issues of solving the resulting optimization problem which—despite its convexity—remains extremely difficult due to the presence of noise in the gradients. We introduced the Robbins-Monro stochastic optimization method and showed how it can lead to improved performance in the presence of noise, while demonstrating its limitations in high-dimensional settings. Furthermore, we developed a modified version of Robbins-Monro, which we call Modified Robbins-Monro, which effectively utilizes curvature information and leads to accelerated convergence. We demonstrated our claims using two numerical examples: (1) finding a mean field approximation to the lattice gas model and (2) coarse-graining the SPC/E water model. In our current work, we are investigating the mathematical nature of MRM, trying to find theoretical or practical criteria for the choice of the learning parameters. We believe that schemes which automatically adjust the learning sequence and the number of conjugate-gradient iterations per step can lead to accelerated convergence. Another important aspect of ongoing research is the development of convergence criteria for MRM. Both these goals, require in-depth mathematical analysis of the proposed algorithm. A final point that we wish to make is that additional computational savings can be achieved by employing “importance sampling” techniques for the estimation of the gradients of the relative entropy in the sense of Ref. 11. Such an approach essentially allows one to estimate the gradients at a new \( \gamma \) making use of a CG simulation on a nearby \( \gamma \). Therefore, it is expected to significantly reduce the total number of CG simulations. Since, such an endeavor deserves special attention, we postpone it for a future work.

ACKNOWLEDGMENTS

This research was supported by an OSD/AFOSR MURI09 award on uncertainty quantification, the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research and the Computational Mathematics program of the National Science Foundation (NSF) (Award No. DMS-1214282). This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy (Contract No. DE-AC02-05CH11231). Additional computing resources were provided by the NSF through TeraGrid resources provided by NCSA (Grant No. TG-DMS090007).