Stochastic quasi-Newton molecular simulations

Chau, C.D.; Sevink, G.J.A.; Fraaije, J.G.E.M.

Stochastic quasi-Newton molecular simulations

C. D. Chau,* G. J. A. Sevink, and J. G. E. M. Fraaije
Leiden Institute of Chemistry, Leiden University, P.O. Box 9502, 2300 RA Leiden, The Netherlands

(Received 23 December 2009; revised manuscript received 22 April 2010; published 24 August 2010)

We report a new and efficient factorized algorithm for the determination of the adaptive compound mobility matrix B in a stochastic quasi-Newton method (S-QN) that does not require additional potential evaluations. For one-dimensional and two-dimensional test systems, we previously showed that S-QN gives rise to efficient configurational space sampling with good thermodynamic consistency [C. D. Chau, G. J. A. Sevink, and J. G. E. M. Fraaije, J. Chem. Phys. **128**, 244110 (2008)]. Potential applications of S-QN are quite ambitious, and include structure optimization, analysis of correlations and automated extraction of cooperative modes. However, the potential can only be fully exploited if the computational and memory requirements of the original algorithm are significantly reduced. In this paper, we consider a factorized mobility matrix $B=JJ^T$ and focus on the nontrivial fundamentals of an efficient algorithm for updating the noise multiplier J. The new algorithm requires $\mathcal{O}(n^2)$ multiplications per time step instead of the $\mathcal{O}(n^3)$ multiplications in the original scheme due to Choleski decomposition. In a recursive form, the update scheme circumvents matrix storage and enables limited-memory implementation, in the spirit of the well-known limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method, allowing for a further reduction of the computational effort to $\mathcal{O}(n)$. We analyze in detail the performance of the factorized (FSU) and limited-memory (L-FSU) algorithms in terms of convergence and (multiscale) sampling, for an elementary but relevant system that involves multiple time and length scales. Finally, we use this analysis to formulate conditions for the simulation of the complex high-dimensional potential energy landscapes of interest.

DOI: 10.1103/PhysRevE.82.026705 PACS number(s): 02.70.Ns, 05.10.Gg

I. INTRODUCTION

The development of general-purpose methods for large-scale molecular simulation is an important scientific goal. The area of application is large and diverse, but one may think of typical phase separation phenomena in hard and soft matter systems, including particle cluster optimization, membrane formation, protein folding, micelles, and polymer dynamics. The magnitude of the challenges involved cannot be underestimated, since in many chemical systems of relevance, the starting point for the model is necessarily atomic or molecular, while the emerging collective behavior on long length and time scales—determining the relevant material properties and function—essentially is not. A prime example is protein-folding, where the characteristic (length and time) scales associated with the smallest constitutive elements (electrons or atoms) deviate many orders of magnitude from those associated with the co-operative motion of protein domains, such as beta sheets or alpha helixes. Since the simulated system evolution, or, alternatively, the sampling rate on the complex energy hypersurface, is dictated by the *smallest* scale in the model description, this co-operative motion remains inaccessible even on present-day supercomputers.

A common strategy to overcome some of these problems is by going from high to lower resolution, i.e., by averaging over the smallest degrees of freedom. Our starting point [1] is such a coarse-grained model, the general position Langevin equation describing the Brownian behavior of N interacting particles in the high friction limit, written in Ito form as

$$d\mathbf{x} = -B\,\nabla\,\Phi(\mathbf{x})dt + \sqrt{2kTB}dW(t), \tag{1}$$

for a molecular potential energy Φ depending on the system state $\mathbf{x}\in\mathbb{R}^{3N}$, with Boltzmann constant k_B, temperature T and $W(t)$ a multivariate Wiener process with $\langle dW_i(t)dW_j(t)\rangle = \delta_{ij}dt$. The standard mobility B is constant and inversely proportional to the viscosity of the surrounding medium. Typically, such a (Brownian) dynamics model is simulated by an Euler scheme, over many, many time steps. The time step is determined by the fastest modes in the coarse-grained representation, associated with the steepest gradients in the energy landscape, and hence the scheme is again rather inefficient for slow modes associated with shallow gradients. Near phase boundaries these systems suffer from critical slowing down (we borrow the following arguments and notation from Dünweg [2]) due to the appearance of a very long correlation times. In these conditions, the system exhibits large correlated objects or "critical clusters" of typical size ξ (the correlation length), which can be made arbitrarily large by means of some control parameter. As a general feature very many configurations are easily accessible, since the typical energy to change, create or delete such an object is, at most, of order of the thermal excitation energy k_BT. The key challenge in simulating these systems is that the physical dynamics is usually *local*, whereas the collective behavior is not. The rearrangement on a larger scale depends on the spread of information through the smaller scales (for diffusive dynamics, with rearrangement time $\tau\propto\xi^2$) and thus consumes increasingly more time for increasing object size ξ. In other words: the hypersurface associated with conformational (re)arrangement on a larger scale is relatively flat, and the dynamics dictated by Eq. (1) slows down extremely due to it local nature, the constant mobility and (almost) vanish-

*c.chau@chem.leidenuniv.nl

ing gradients. The system becomes increasingly soft and sluggish—the true hallmark of soft matter.

Common strategies to circumvent the limitations in accessible time and length scales are based on selecting and updating large length scales with artificially high rates. In molecular modeling of (bio)molecule dynamics, one often relies on enhanced sampling via parallel tempering/replica exchange [3,4], simulated tempering [5], solute tempering [6], multicanonical molecular dynamics (MD) [7], and Wang-Landau [8]. Others, like metadynamics [9] and hyper-MD [10], introduce a bias on a small set of collective variables and recast the problem in terms of transition-state theory. Our S-QN method can also be seen as a method for enhanced sampling and (global) optimization [1]. Our focus here is its general applicability and on the multiscale features, in particular the multiplicity of time steps that is *automatically* introduced by including curvature information of the *unbiased* potential energy hypersurface. Our approach is essentially a real-space generalization of existing accelerated algorithms that use filtering for the separation of different length/time scales [11–13]. This Fourier acceleration (FA) technique [2] attempts to renormalize the characteristic times associated with different (Fourier) modes by using a mass matrix as a preconditioner to the forces in the Fourier domain. As such, it enables a *multiplicity* of time steps. Consequently, the determination of an appropriate mass matrix is key to the success of this technique. The mass matrix should be positive definite (due to the appearance of a square root in the noise term), and is often regularized to avoid problems associated with very small wave vectors. The renormalizing mass matrix can be determined analytically for a purely Gaussian model (a quadratic potential), where Fourier modes completely decouple and the integration can be carried out independently. For this model Hamiltonian, FA has indeed been shown to completely eliminate critical slowing down [14]. For general Hamiltonians with higher order terms, different modes may be *coupled*, and preconditioning with this mass matrix can easily fail. In particular, it is *a priori* uncertain if FA will work at all [15] and FA is also known to suffer from discretization artifacts [16]. We explicitly note that following or (re)constructing the actual "physical" dynamics of the system is not our purpose. In this sense, the S-QN method is very similar to FA and many of the other methods on different levels of description that are aimed at accelerated or enhanced sampling of energy landscapes. By construction, however, the characteristics of large-scale dynamics and important correlations will always be directly accessible.

We first clarify the central idea of S-QN. For simplicity, we omit the spurious drift term [1]. We consider one of the simplest systems possible, a Harmonic potential $\Phi(x)=\frac{h}{2}x^2$ ($x\in\mathbb{R}$), with h (in J/m^2) the force/spring constant. After introducing a second differential equation for the adaptive mobility B, the Langevin equation for this system is given by [1]

$$dx = -Bhxdt + \sqrt{2k_BTB}dW(t) \quad \text{initial state}, \tag{2}$$

$$\frac{dB}{dt} = -B + \frac{1}{h}, \tag{3}$$

$$B(0) = 1. \tag{4}$$

We note that the second equation is only used for the purpose of illustration: in the S-QN method, the constant inverse Hessian of Φ, $1/h$, is recursively determined using QN methodology. From Eq. (2), it is clear that the initial behavior at $t=0$ is the *same* as for Eq. (1), i.e., Langevin dynamics with a constant $B=1$. In this stage, the noise is decoupled from the energy landscape, and only the drift term acknowledges the local gradients on the potential energy hypersurface. In the stationary state ($B=1/h$), however, Eq. (2) simplifies to

$$dx = -xdt + \sqrt{2\frac{k_BT}{h}}dW(t) \quad \text{stationary state}. \tag{5}$$

We observe reversed roles: the drift term is no longer dependent on the gradient, but the random displacement is strongly dependent on the gradient through the proportionality to $\sqrt{1/h}$, and will therefore decrease in magnitude for increasing h. In addition, depending on the value of h, the drift term in Eq. (5) gives rise to an acceleration ($h<1$) or slowing down ($h>1$) compared to Eq. (1), or, alternatively, an effective scaling of the time by $1/h$. More general, the method was designed to automatically apply dense sampling in narrow basins with steep gradients containing minima and larger sampling steps in almost flat parts of the energy hypersurface where the gradients almost vanish. The noise term facilitates the escape from basins [1]. This differentiated sampling rate (for fixed dt) is obtained by acknowledging the topography of the hypersurface via curvature information. Hence, efficient incorporation of proper curvature information is vital, and we previously showed [1] that the standard QN framework for numerical optimization provides such methodology. In particular, for a constant time step dt, the inverse Hessian is iteratively constructed by a Broyden-Fletcher-Goldfarb-Shanno (BFGS) method using only gradient information in subsequent sampling points. However, we recognize that for the target systems involving multiple scales or, equivalent, large n, memory requirements and/or the computational load can still become limiting for the application of the S-QN method. The Cholesky factorisation, necessary for computing the noise term [1], represents a considerable [$\sim\mathcal{O}(n^3)$] computational burden for each iteration. Storage may become an additional burden, since several $n\times n$ matrices should be updated and/or stored at each step. Consequently, algorithmic improvements that leave the general properties of the method unaffected but substantially *reduce* the storage and computational requirements are of great importance for the value of S-QN as an efficient general-purpose simulation method. Here, we focus on the derivation of such new and efficient algorithms. Factorized QN methods using triangular matrices have been considered, but these methods were primarily designed to avoid positive semidefinite or negative definite updates due to rounding errors, i.e., to enhance numerical stability [17,18]. Since our aim is different, i.e., to update J via a direct procedure, we derive a factorized secant update scheme (FSU) for B_+ of the Brodlie form [19]. The same secant condition should now hold for the $B_+=(I+\mathbf{v}\mathbf{y}^T)B(I+\mathbf{v}\mathbf{y}^T)^T$ that is very similar to the one introduced by

TABLE I. Schematics of methodology: Complete update and Limited memory update in QN and S-QN methods.

	Update scheme	
Minimization method	Full	Truncated
QN method	BFGS	L-BFGS
S-QN method	FSU	L-FSU

matrix $B=JJ^T$ and B is, by construction, positive definite. We will show that this FSU scheme reduces the total computational costs to $\sim\mathcal{O}(n^2)$ for each iteration, which remains tractable even for large n. To further reduce the requirements and avoid matrix storage, we cast FSU in a new recursive scheme inspired by limited-memory BFGS (L-BFGS). The L-BFGS method was earlier developed [20] to address large-scale problems, and has the advantage that the amount of storage (and thus the cost per iteration) can be controlled by the user, while retaining good overall performance. Our limited-memory FSU (L-FSU) scheme stores only three vectors of length n per iteration, and provides means to further restrict the computational costs per iteration by limiting the number of stored corrections (m) incorporated in J (and thus B). The analogy between the QN and S-QN frameworks is illustrated in Table I.

The paper is organized as follows: in the theory section, we derive the factorized FSU and L-FSU scheme, and quantify how further reduction of computational costs and storage is possible with L-FSU. In Sec. III we consider the performance of FSU and L-FSU for a set of coupled harmonic oscillators. We consider this system for simple reasons: (a) the Hessian H is analytically known and the convergence properties of B_k, as determined by FSU and L-FSU, can be quantitatively analyzed, (b) multiple time and length scales play a role in the dynamics of this system, (c) the system is a starting point for coarse-grained protein modeling. In the analysis we focus on the convergence of $B\to H^{-1}$, the presence of cooperative motion along the sampling pathway and the sampling distribution at long time scales. In particular, we focus on the effect of truncation (L-FSU) on these properties and the determination of a good history depth m. We will shortly elaborate on additional properties of the method, e.g., how the mobility can be used for introducing a multiplicity of time scales and the efficient and automated calculation of correlations in local or global minima.

II. THEORY

A. S-QN

The S-QN method is based on a new stochastic Langevin equation for general n dimensional potentials Φ given by [1]

$$d\mathbf{x} = [-B(\mathbf{x})\,\boldsymbol{\nabla}\,\Phi(\mathbf{x}) + k_BT\,\boldsymbol{\nabla}\cdot B(\mathbf{x})]dt + \sqrt{2k_BT}J(\mathbf{x})dW(t), \tag{6}$$

where $J(\mathbf{x})$ is related to the mobility $B(\mathbf{x})$ through

$$B(\mathbf{x}) = J(\mathbf{x})J(\mathbf{x})^T. \tag{7}$$

The new second term in the right hand side of Eq. (6) is the spurious drift term or flux caused by the random force. The crucial ingredient of our S-QN method is the mobility $B(\mathbf{x})$ or, in the discrete form, the $n\times n$ matrix B. We have previously discussed that our choice for the mobility matrix is inspired by Newton methods, i.e., H^{-1}, the inverse Hessian of Φ [1]. We relied on the BFGS standard in Quasi-Newton numerical optimization for constructing a series of positive definite matrices B_k (with k the time index in the discretized Langevin equations), such that $B_k\to H^{-1}$ under specific conditions [1]. These B_k constitute the adaptive compound mobility matrix that responds to the energy landscape by a memory function. We note that the (inverse) Hessian for the potentials Φ considered in the examples section is always a constant. Consequently, the spurious drift term in Eq. (6) is negligible for all B_k due to the closeness property and vanishes completely when B_k has converged to the inverse Hessian. For simplicity, we have therefore disregarded the spurious drift term in the remainder. We validated this explicitly for the systems considered in the examples section. The S-QN method and our new update algorithms for the mobility are, however, not in any way restricted to this special case. In particular, the update scheme for Eq. (6) in Appendix, Sec. 1 shows how the general case is only a correction to this special case, at the expense of additional costs.

We note that efficiency, i.e., avoiding the computation of the exact Hessian for large n, is not the only reason for the choice of BFGS. In general n-dimensional problems, the Hessian can and will become negative definite or even singular in parts of the energy landscape. This results in conditions for B that are principally equal to the implicit condition for the mass matrix in FA: B should *always* be positive definite to guarantee the existence of J in Eq. (7). When the secant condition is satisfied the BFGS update method guarantees the construction of such a positive definite B. Nevertheless, the sampling path will have to cross over concave and flat regions of the energy landscape, and we need to somehow adapt the mobility in these regions [1] (see in the body of this paper). The most important feature of the general methodology is that the mobility B always exists and remains positive definite, since the BFGS update method constructs an *approximate* inverse Hessian, even when the Hessian itself is singular. This property is equivalent to the somewhat *ad hoc* regularisation in FA methods, but *automatic*. By using this nonsingular approximation, sampling of the longer wavelength modes associated with zero (and other very small) eigenvalues of the Hessian is enhanced, hence the automatic scaling in our system. It is this property that will allow one to introduce a multiplicity of time steps by taking differential steps in different directions while maintaining thermodynamic consistency. Hence, the S-QN method bridges between general directed search methods such as QN and random search methods such as Monte Carlo (MC) or simulated annealing (SA). The QN method only ensures that the sampling path on this hypersurface is always in the descending direction, and is therefore principally *local*. In particular, QN does not sample according to a distribution. The update schemes in MC/SA incorporate global hypersurface information only weakly in the form of rules for acceptance or rejection based on a sampling distribution that favors configurations $\mathbf{x}$ with lower $\Phi(\mathbf{x})$ [21,22]. A well-known drawback of this method is performance: for large systems (large n), the random sampling and subsequent substantial increase of the number of sampling points or function evaluations can make these algorithms computationally intractable.

B. Factorized secant update scheme

As mentioned in the introduction, a factorized update scheme was earlier developed to circumvent problems with the positive definiteness of B due to numerical errors [17,18]. The method of Goldfarb updates a lower triangular matrix L to $\bar{J}$, followed by a decomposition of the matrix $\bar{J}$ into an orthogonal Q and a right triangular matrix R, i.e., a QR factorization, to obtain the new L with $\bar{J}\bar{J}^T=B^{-1}$. Since $B^{-1}=LQ^TQL^T=LL^T$, the next QN direction $(\Delta\mathbf{x})$ can be determined by solving $LL^T\Delta x=-\nabla\Phi$. Here, the particular reason for developing a factorised scheme for the QN-matrix $J(\mathbf{x})$ is rather different. In particular, apart from the drift term

$$-B(\mathbf{x})\,\nabla\,\Phi(\mathbf{x})dt=-J(\mathbf{x})J(\mathbf{x})^T\,\nabla\,\Phi(\mathbf{x})dt, \tag{8}$$

such a scheme enables a *direct* calculation of the noise term

$$\sqrt{2k_BT}J(\mathbf{x})dW(t), \tag{9}$$

without an additional Choleski factorization of the matrix B. In the following, we consider a discrete set of equations, with equidistant time steps labeled by k (starting from $t=kdt=0$). In line with common practice in QN, the matrix B_k (the approximate of the inverse Hessian) is updated each iteration step to obtain a new matrix $B_{k+1}=B_k+\Delta B_k$, with ΔB_k a correction matrix. Suitable conditions for incorporating second-order information of Φ in this new matrix should be formulated, and we use the standard secant condition, based on expanding $\nabla\Phi$ in $\mathbf{x}_k-\mathbf{x}_{k+1}$ around the new point $\mathbf{x}_{k+1}$ as a Taylor expansion

$$\nabla\Phi(\mathbf{x}_k)\approx\nabla\Phi(\mathbf{x}_{k+1})+\nabla^2\Phi(\mathbf{x}_{k+1})(\mathbf{x}_k-\mathbf{x}_{k+1}). \tag{10}$$

The property that B^{-1}_{k+1} approximates $\nabla^2\Phi(\mathbf{x}_{k+1})$ is equivalent to the secant condition given by

$$B_{k+1}\mathbf{y}_k=J_{k+1}J^T_{k+1}\mathbf{y}_k=\mathbf{s}_k, \tag{11}$$

where $\mathbf{s}_k=\mathbf{x}_{k+1}-\mathbf{x}_k$ and $\mathbf{y}_k=\nabla\Phi(\mathbf{x}_{k+1})-\nabla\Phi(\mathbf{x}_k)$. Other properties that B_{k+1} should inherit from B_k include symmetry and positive definiteness. Unlike in standard QN these properties of B_{k+1} are automatic, since the product JJ^T is always symmetric positive definite for nonsingular matrices $J\in\mathbb{R}^{n\times n}$. To determine a *unique* update J_{k+1}, we have to impose an additional condition on J. From all matrices J that satisfy the secant condition (11), we determine the one that is *closest* to J_k in some sense (see below), such that useful information stored in J_k is not lost in the update. The proximity condition, giving rise to a unique J_{k+1}, is casted into

$$\min_{J_{k+1}}\|J_{k+1}-J_k\|, \tag{12}$$

$$J_{k+1}\mathbf{v}_k=\mathbf{s}_k, \tag{13}$$

$$J^T_{k+1}\mathbf{y}_k=\mathbf{v}_k. \tag{14}$$

where the last two equations express the secant conditions on J_{k+1}. The convergence property of J is hereby satisfied: if all curvature information is stored in J_k, it is automatically inherited by $J_{k+1}=J_k$. There exist a nonsingular J_{k+1} satisfying Eq. (11), if and only if the curvature condition $\mathbf{s}^T_k\mathbf{y}_k>0$ holds. When $\|\cdot\|$ is the Frobenius norm, the solution to Eqs. (12) and (13) is even unique and J_{k+1} is given in terms of vectors $\mathbf{s}_k$ and $\mathbf{v}_k$. The uniqueness and existence proof is analog to the proof [17] given for

$$\min_{\bar{J}_{k+1}}\|\bar{J}_{k+1}-L_k\|, \tag{15}$$

where L_k is a lower triangular matrix, with the secant condition on $B^{-1}_{k+1}=\bar{J}_{k+1}\bar{J}^T_{k+1}$. We note that the lower triangular matrix L in this scheme was chosen for convenience: the relation for the update Δx, $LL^T\Delta x=-\nabla\Phi$, is easily solved for such L by forward and backward substitutions. Here, we want the update J_{k+1} in closed form (in terms of $\mathbf{s}_k$ and $\mathbf{v}_k$) instead, because this allows us to cast the scheme into a recursive form. In the next section we show how this recursive scheme can be exploited for limited-memory purposes. Using Eq. (14), $\mathbf{v}_k$ can be determined (see appendix 2) and J_{k+1} is given by

$$J_{k+1}=J_k+\frac{\alpha_k\mathbf{s}_k\mathbf{y}^T_kJ_k-\alpha^2_kJ_kJ^T_k\mathbf{y}_k\mathbf{y}^T_kJ_k}{\mathbf{y}^T_k\mathbf{s}_k}, \tag{16}$$

with

$$\alpha^2_k=\frac{\mathbf{y}^T_k\mathbf{s}_k}{\mathbf{y}^T_kJ_kJ^T_k\mathbf{y}_k}. \tag{17}$$

We can take the square root if $\mathbf{y}^T_k\mathbf{s}_k>0$, the curvature condition, assuring the positive definiteness of the QN-matrix. In particular, we consider the positive root of α_k for which J_{k+1} is minimal in Eq. (12). We note that, in order to enhance the sampling of the potential energy hypersurface in concave or flat regions, we avoid the usual backtracking algorithms and use a constant increment dt instead. Gradient information is therefore always only calculated once during each update. It is essential to have a good strategy when the curvature condition is violated. In such case, we update $\mathbf{s}_k$ and $\mathbf{y}_k$ but keep the matrix B fixed, i.e., $J_{k+1}=J_k$, to ensure positive definiteness. As a tradeoff, the secant condition $B_{k+1}\mathbf{y}_k=B_k\mathbf{y}_k=\mathbf{s}_k$ is not necessarily satisfied. However, since the next step is effectively a reinitialization of the scheme with $J_0=J_k$, the secant condition is restored in step $k+2$ and can be disregarded at step $k+1$. We refer to our previous work [1] and the discussion section for more details on this particular choice.

We conclude that the factorized secant update (FSU) of Eq. (16) gives rise to a direct calculation of the drift [Eq. (8)] and noise [Eq. (9)] terms without further factorizations. Since the FSU update for B is equivalent to the update obtained from the Davidon-Fletcher-Powell (DFP) method of the convex BFGS family (see Appendix, Sec. 2) we shortly review some general properties of DFP for a quadratic objective function such as Eq. (21). Fletcher and Powell showed that, for a nonsingular Hessian H, the matrix B converges to H^{-1} and x_k finds the minimum of Φ in exactly n steps if *exact* line searches are used [23]. The proof is based on showing that $\mathbf{s}_k$ $(k=0,\ldots,n-1)$, with step size α_k selected based on exact line searches, are linearly independent eigenvectors of B_nH with eigenvalue unity (in other words, $\mathbf{s}_k$ $k=0,\ldots,n-1$ is an basis for $\mathbb{R}^n$ and $B_nH=I$). In reality, exact line searches are seldom used. They are considered inefficient, as often very many costly Φ-evaluations are required for the determination of the QN stepsize α_k in each iteration step. The preferred backtracking algorithms [24] are less efficient in terms of convergence, but the number of Φ-evaluations per iteration is also considerably reduced. In our procedure backtracking was not considered for various reasons, and the number of Φ-evaluations per iteration step is further reduced (to one). Returning to exact line searches [23], the update $B_{k+1}=B_k+A_k+A'_k$ consists of two rank-one contributions to B_k. The first contribution A_k ensures that the secant condition $B_{k+1}\mathbf{y}_k=\mathbf{s}_k$ is always satisfied, or, alternatively, that $\mathbf{s}_k$ is an eigenvector of $B_{k+1}H$ with unit eigenvalue. The second contribution A'_k deals with the convergence of B to H^{-1}, and one can show that $H^{-1}=\Sigma_{k=0}^{n-1}A'_k$. For an inexact or constant step size the vector $\mathbf{s}_k$ is in general not orthogonal to $g_{k+1}=\boldsymbol{\nabla}\Phi(x_{k+1})$ and, consequently, the buildup of information of H^{-1} in A'_k will be (much) slower. We note that the noise in Eq. (20) adds a 'random' displacement $\mathbf{d}$ to the QN-update, and this $\mathbf{d}$ is likely to also contain displacements orthogonal to g_{k+1}. For general inexact line searches, the update can be seen as successive rank reduction and rank restoration [25]. The first step A_k always reduces the rank of B_k by one. The second step A'_k restores the rank to the rank of B_k and gives rise to a positive definite B_{k+1}, all provided that the curvature condition $\mathbf{y}_k^T\mathbf{s}_k>0$ is satisfied. The difference $B_{k+1}-B_k$ is a symmetric matrix of rank at most two, with column and row spaces spanned by $\mathbf{s}_k$ and $\mathbf{h}_k=B_k\mathbf{y}_k$. A rank-two update is obtained only for linearly independent $\mathbf{s}_k$ and $\mathbf{h}_k$; otherwise the update is of rank one [26]. Davidon [27] showed that the generalized eigenvalue problem $B_{k+1}\mathbf{z}=\lambda B_k\mathbf{z}$ has $n-2$ unity eigenvalues and 2 eigenvalues that may differ from 1. These eigenvalues can be determined analytically as a function of $a=\mathbf{y}_k^TB_k\mathbf{y}_k$, $b=\mathbf{y}_k^Ts_k$, and $c=s_k^TB_k^{-1}s_k$ [27,26].

However, when minimizing a system of size n, the computational and storage costs may still be impractical for large n. In the next section we therefore propose an efficient implementation of FSU, which leads to a reduction of the n^2 storage and $\mathcal{O}(n^2)$ manipulations of the scheme described in Eq. (16).

C. Recursive limited-memory update

Limited memory methods in the Broyden family are in general based on truncating the history in the iterative update schemes for the approximate Hessian or inverse Hessian B. The direct advantage is in storage: at iteration k only a fixed number m (the history depth) of vector sets is stored, instead of the linearly growing number of sets $(\sim k)$ in the FSU update scheme. In particular, for $k>m$ the oldest information contained in B is discarded and replaced by the new one. Although one could intuitively expect this truncation to affect the performance of the method, numerical evaluations show that this procedure for BFGS (the L-BFGS method) gives rise to good convergence properties in practice, even for small m [20]. A theoretical understanding of this property for general cases is still lacking.

In Appendix, Sec. 4, we provide the details of a recursive scheme that is suited for both FSU and limited-memory FSU (L-FSU). The strategy is to avoid the use of expensive matrix-vector products by loop unrolling. Instead of n^2 storage for the matrix J in Eq. (16), this scheme requires storage of three vectors $\{\mathbf{s}_k,\mathbf{y}_k,\mathbf{h}_k\}$, each of length n. For L-FSU, this results in a total storage of $3mn$, with m in general small (see appendices). The starting matrix for the limited-memory updates, J_0, can be freely chosen, but should reflect the discarded information for the problem at hand. A standard choice is the same initial value for J_0 as considered for FSU, i.e., $J_0=I$, but scaling $J_0=\gamma_kI$ with

$$\gamma_k=\mathbf{y}_k^T\mathbf{s}_k/\|\mathbf{y}_k\|^2 \tag{18}$$

was identified a simple and effective way of introducing a scale in the algorithm and improving its performance [20]. However, the analysis and the numerical examples (see next section) use $J_0=I$. Just like in the L-BFGS the starting matrix J_0 can be freely adjusted in our scheme during the iterative process.

In contrast to FSU, one should take special care when the update is temporarily stalled due to a violation of the curvature conditions $\mathbf{y}_k^T\mathbf{s}_k>0$. The limited-memory algorithm employs a shifting window of m vector-triplets $\{\mathbf{y},\mathbf{s},\mathbf{h}\}$ to update the B matrix at each k step, starting from the initial condition $J_0=I$ (all eigenvalues$=1$). Since the update scheme is of rank two at most, a maximum of $2m$ eigenvalues of B will deviate from $\lambda=1$ (see also previous section) at *any* stage of the simulation. All local-curvature information, or, alternatively, information about the different modes in the system, is contained in the eigenvectors with eigenvalue $\lambda\neq1$. The history contained in this window may stretch over a much longer range than simply anticipated from the recursive schemes, since B is not always updated. For the sake of the argument, we suppose $k=\bar{k}\geq m$ when the curvature condition is first violated and that the second violation takes place after an additional number of steps $\geq m$. The general case is more involved but straightforward. In Appendix, Sec. 3 we have shown that, in order to satisfy the secant condition at all times, adapted V_k based on $\tilde{\mathbf{h}}$ instead of $\mathbf{h}$ are required after m updates. Upon updating B at step $\bar{k}+1$, one option is to disregard the information contained in B and restart the L-FSU scheme with $J_0=I$, since by definition this information can be restored in m steps. As in FSU (see Sec. II B) one can also see the next step as an effective re-initialization of the scheme with $J_0=J_{\bar{k}}$, based on the logic that especially for large m the matrix $J_{\bar{k}}$ contains valuable Hessian information. In the latter case, one can either store the $n\times n$ matrix $J_{\bar{k}}$ or build $J_{\bar{k}}$ recursively from its particular window of past vector-triplets (a maximum of m). In both cases, the use of $J_{\bar{k}}$ introduces additional memory and/or computational requirements. The secant condition is restored in step $\bar{k}+2$ and can be disregarded at step $\bar{k}+1$. Whatever restart method is most efficient depends on the values of m and n. For both J_0, however, it is important to note that the L-FSU scheme should not truncate during the first m step, i.e., L-FSU and FSU after restart are equal and V_k based on $\mathbf{h}$ should be considered in the update.

We shortly analyze the computational load by considering the total number of elementary operations for one update cycle [Eq. (6)]. As a reference, the FSU update [see Eq. (16)] requires $4n^2$ multiplications (first calculate $J_k^T\mathbf{y}_k$ and reuse this vector in the update). Calculating the drift part directly from $J_kJ_k^T\boldsymbol{\nabla}\Phi$ requires $2n^2$ multiplications. A total of n^2 multiplications are needed for the noise part J_kdW. Summing all up, the total number of multiplications using the FSU update is $7n^2$. In the recursive L-FSU update scheme, $10mn+2n$ and $2mn+n$ multiplications are required for the calculation of the drift and noise term, respectively (see Appendix, Sec. 4 for details). A total of $12mn+3n$ multiplications is therefore required for the L-FSU update. A conclusion from this simple analysis is that L-FSU is more efficient than the FSU update scheme, in terms of computational as well as memory costs for $m\leq n/2$ and definitely better than using Cholesky decomposition. Since memory and computational constraints are typically problematic for large n, this is a much desired property.

III. RESULTS AND DISCUSSION

In the following examples, we have discretized Eq. (6) into

$$\mathbf{x}_{k+1}=\mathbf{x}_k+\Delta\mathbf{x}_k, \tag{19}$$

$$\Delta\mathbf{x}_k=-J(\mathbf{x}_k)J(\mathbf{x}_k)\,\boldsymbol{\nabla}\,\Phi(\mathbf{x}_k)\Delta t+\sqrt{2k_BT}J(\mathbf{x}_k)\Delta W. \tag{20}$$

using an explicit Euler scheme. Here $\Phi=\Phi_{\text{spring}}$ is a harmonic potential, describing the behavior of n connected particles at $\mathbf{x}=(x_1,\ldots,x_n)^T$ on a line, with

$$\Phi_{\text{spring}}=\sum_{i=1}^{n-1}(x_{i,i+1}-1)^2; \tag{21}$$

and $x_{i,i+1}=|x_i-x_{i+1}|$ the distance between particle i and $i+1$. For simplicity, we have set the equilibrium distance and spring constants to unity. We note that the considered model is related to the well-known Rouse model in polymer dynamics, which we can obtain by taking a spring constant $3k_BTb^{-2}$, with b the Kuhn segment length, and a vanishing equilibrium length in Eq. (21) and constant mobility $B=JJ^T=\zeta^{-1}$ in Eq. (20), where ζ is the friction coefficient due to the surrounding medium. Although our one-dimensional (1D) particle-spring system seems trivial at first sight, it possesses many features, in particular critical slowing down, that are present in nontrivial models, which automatically require simulation. An clear advantage is that the analytic Hessian H is easily calculated, and H can thus be directly compared to B_k^{-1}. In particular, this Hessian $H_{\Phi_{\text{spring}}}$ is a tridiagonal matrix,

$$H=\begin{bmatrix} 2 & -2 & 0 & & \\ -2 & 4 & -2 & & \\ & \ddots & \ddots & \ddots & \\ & & -2 & 4 & -2 \\ & & 0 & -2 & 2 \end{bmatrix}, \tag{22}$$

which is singular. The kernel or null-space of H, Null (H), is spanned by the n dimensional vector $1=[1\ldots1]^T$, corresponding to the translational invariance of the energy potential Φ_{spring}, i.e., the insensitivity of the potential to a translation of the string of particles as a whole. An equivalent Gaussian model was analytically considered by Dünweg [2], who derived it from the well-known Landau-Ginzburg Hamiltonian by omitting the ϕ^4 term. Dünweg showed that the correlation times of longer-wavelength modes (smaller p) become increasingly large. With an increasing length scale, the system therefore becomes increasingly soft or—correspondingly—increasingly sluggish.

We concentrate on three features: (1) the similarity between B_k and the inverse Hessian, (2) the sampling performance, and (3) the sampling distribution.

A. Comparison of the adaptive mobility to the inverse Hessian

One consequence of the singularity of the Hessian is that a direct comparison between H^{-1} and B_k is impossible. We can use the *generalized* inverse matrix H^- [28], that exists for any H such that $HH^-H=H$, but for a singular matrix it will not be unique. In particular, for any given generalized inverse H^-, the class can be generated by $H^-+U-H^-HUHH^-$ with U any arbitrary matrix. For a nonsingular H the inverse is unique and $H^-=H^{-1}$. The adaptive mobility B_k itself is positive definite by construction, and the inverse B_k^{-1} can be obtained in $O(n^3)$ operations. However, we circumvent the additional inversion at each iteration step k, and focus on the properties of B_k itself, starting with the eigenvalues of B_k. Rates of convergence can be determined from $\|HB_kH-H\|_{\text{F}}$, the Frobenius norm, as B_k itself may converge to any member of the class of generalized inverses of H. In the discussion we show that for the considered system it is in principle possible to use B_kH-I instead. In addition, we deal with the singularity of H by constraining the system. The first and rather crude option is to reject displacements associated with Null (H) in the update of B_k, by applying an affine translation after each step such that the chain's center of mass $c_{k+1}=c(\mathbf{x}_{k+1})=\Sigma_{i=1}^N x_i$ is always reset to the initial value c_0. A better option, that resolves the singularity itself, is to regularize the system by adding a penalty function containing $c(\mathbf{x})$ to the harmonic potential [Eq. (21)]. First, we evaluate the properties of B_k for the unconstrained case. Consequently, we relate the eigenvalue spectrum for this B_k to the eigenvalue spectrum of the regularized H, using elementary linear algebra.

For a quadratic potential, the buildup of information in B_k is *independent* of the variables Δt and k_BT in Eq. (20), as long as B_k is updated ($\mathbf{y}_k^T\mathbf{s}_k>0$) for each k. Moreover, this process is insensitive to $\mathbf{x}_0$, the initial state in Eq. (20). For these reasons, the spectral properties and rates of conver-

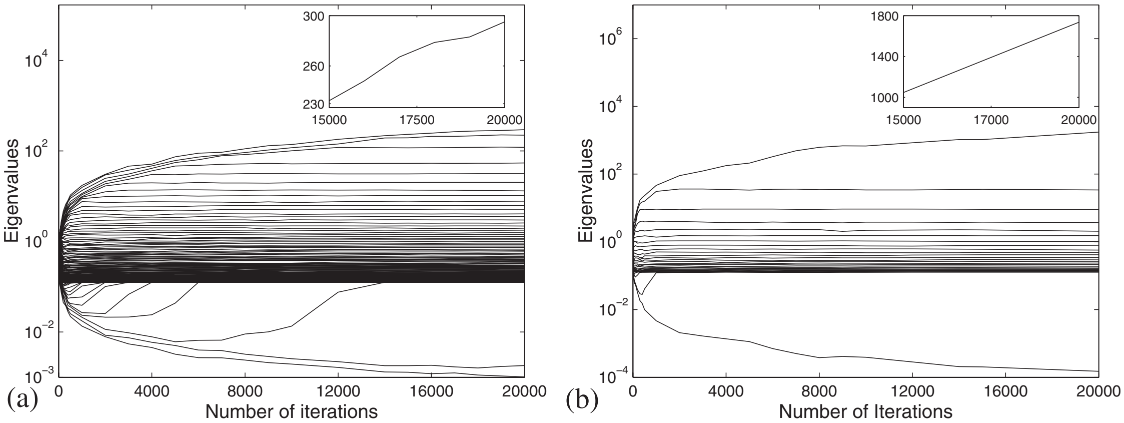

FIG. 1. Eigenvalue spectrum of the mobility matrices constructed by FSU for the spring potential. The insets in (a) and (b) display the evolution of the largest eigenvalue. (a) Spectrum vs iteration index k for $n=100$. (b) Spectrum vs iteration index k for $n=27$.

gence of B_k considered in the next subsections are universal. The sampling pathway itself depends on the values of k_BT and Δt, as well as the initial state, and should be chosen with care. In particular, we have selected $k_BT=0.01$ in Figs. 1–5. For the results shown in Fig. 1 and 3 $\Delta t=0.0001$ and $\Delta t=0.01$ for Fig. 2. In Figs. 4 and 5 we chose $\Delta t=10^{-6}$ since a small time step promotes $\mathbf{y}_k^T\mathbf{s}_k>0$.

1. FSU

We focus on the convergence of B_k, obtained by FSU, for the spring potential Eq. (21) using Eq. (20). We first consider the eigenvalues of the mobility matrices B_k (we omit k dependence for simplicity of notation). Multiple length and time scales should play an important role in systems with a larger number of degrees of freedom, and we start with $n=100$. The simulated k-evolution of the eigenvalue spectrum of B is shown in Fig. 1(a). It is clear that the eigenvalues, apart from the extremes, approach a constant value with increasing k within the limited time of simulation (20 000 steps). The largest (smallest) eigenvalues continue to increase (decrease) with increasing k: focusing on the maximum $\lambda_{\max}$ this increase is approximately linear with k. Visual inspection of the spring dynamics (see next section for details) shows a dominant and random movement of the spring as a whole at later k stages, signaling that the smallest eigenvalue of the *inverse* of the nonsingular B indeed converges to zero with increasing k (or alternatively, $\lambda_{\max}\to\infty$ for B). Force contributions along the eigenvector $\mathbf{1}$, the null space of H, will thus be amplified with increasing k, leading to the dominance of coordinated but diffusive movement of the string at later stages. Figure 1(b) shows the eigenvalue spectrum for a system with a smaller number of degrees of freedom ($n=27$) and is qualitatively very similar. From a comparison between both spectra we observe that the time scale at which most eigenvalues approach constant values depends on the number of degrees of freedom n. We quantify the simulated rate of convergence further by considering the evolution of $\|HBH-H\|_{\mathrm{F}}$ [Fig. 2(a)] and the scaled norm $(1/n^2)\|HBH-H\|_{\mathrm{F}}$ [Fig. 2(b)] for varying $n=10,20,\ldots,100$. Since the Frobenius norm sums over all elements of the $n\times n$ residual matrix $HBH-H$, the scaled norm in Fig. 2(b) provides the average residual per matrix element. From Fig. 2, we observe that for all considered n the matrix B_k converges to a generalized inverse H^{-}, within certain error bounds, and that the residual norm has a typical S-shape. In

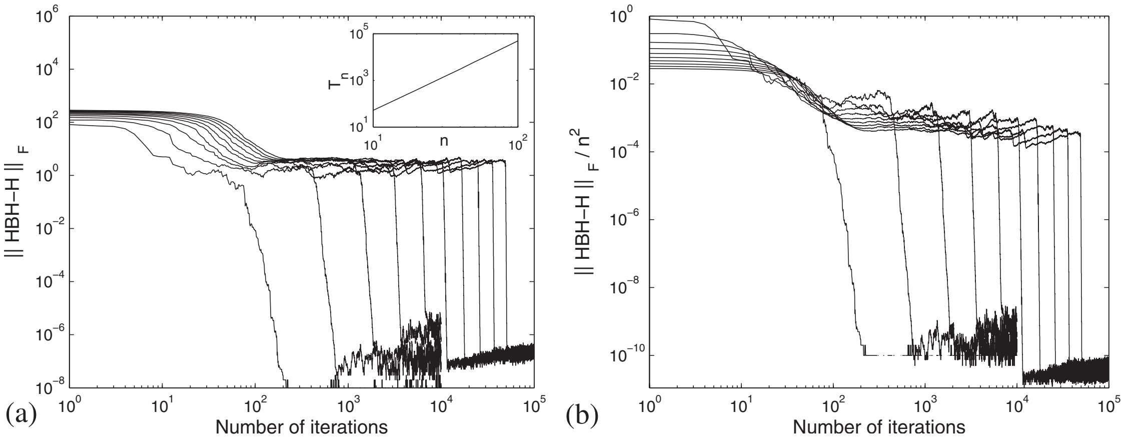

FIG. 2. Rate of convergence of the mobility matrix B_k to the generalized inverse Hessian H^{-1} for $n=10,20,30,\ldots,100$. For all cases the initial guess $B_0=I$, as a result the residual norm at $k=0$ increases with n. The inset in Fig. 2(a) shows the plateau length T_n vs n in a log-log scale. (a) $\|HB_kH-H\|_{\mathrm{F}}$ vs iteration index k for various n. (b) $\frac{1}{n^2}\|HB_kH-H\|_{\mathrm{F}}$ vs iteration index k for various n.

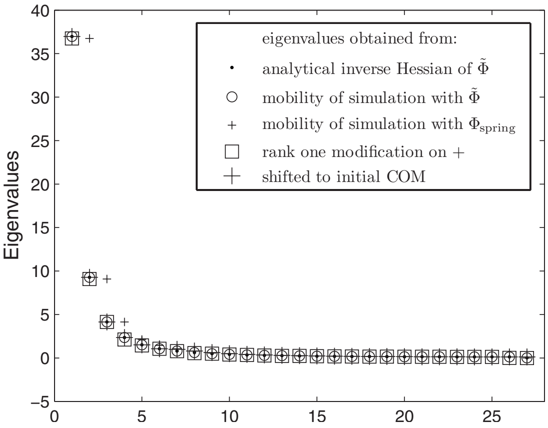

FIG. 3. Comparison of the eigenvalues of the analytic inverse Hessian and the mobility matrix as constructed by FSU for the constrained $\tilde{\Phi}$ and Φ_{spring}.

the initial stages, the FSU scheme starts updating $B_0 = J_0 = I$ (all eigenvalues$=1$) if and only if the curvature condition is satisfied. Focusing on selected simulations shown in Fig. 2, we indeed observed that either one or two eigenvalues become $\neq 1$ after each update. In addition to these two eigenvalues, all other eigenvalues $\neq 1$ are adjusted in each update. All eigenvalues are therefore $\neq 1$ for $k \in [n/2, \ldots, n]$ steps. From Fig. 2(b), we can identify this process as the first smooth and slowly decreasing part of the S-curve that terminates in a plateau value after approximately n steps. The second stage, the noisy plateau region, deals with correction and further buildup of linear independent curvature information via new $\mathbf{s}_k$ and $\mathbf{h}_k$. The plateau value of the residual norm itself seems independent of n [Fig. 2(a)]. This plateau could be seen as an exited state with a lifetime T_n that depends on the dimensionality n of the system. From the T_n vs n plot, shown as insets on a log-log scale in Fig. 2(a), we determined the life time as $T_n = 0.05n^3$. The noisy plateau region terminates via a drop, signaling that the construction of a (generalized) inverse Hessian H^- becomes completed. The constant slope associated with this drop was used in the determination of T_n. Combining Figs. 1 and 2 for $n=27$, we find that prior to the drop most intermediate eigenvalues have almost reached their final and constant value (Fig. 1). The drop at $k \approx 1000$ (Fig. 2) corresponds to the convergence of all $n-2$ eigenvalues to constant values, although small fluctuations around these values can be observed at later stages. Only the two extreme eigenvalues are still increasing ($\lambda_{\max}$) and decreasing ($\lambda_{\min}$) for $k > 1000$. The fact that $\lambda_{\min}$ and $\lambda_{\max}$ do not level off with increasing k shows that the condition number of B_k increases with k and is consistent with the rank-two nature of the update scheme. Moreover, these eigenvalues are inversely proportional to the extremes for H, corresponding to the fastest and slowest modes in the system. This observation is consistent with the simple analysis in the discussion section, which anticipated that the relaxation rates associated with the slowest (fastest) modes is increased (decreased) with respect to standard Langevin dynamics ($B_k = I$), respectively. For $n=100$ the instantaneous drop occurs at $k \approx 50\,000$ (Fig. 2). We conclude that the earlier observation, i.e., all $n-2$ eigenvalues have converged at $k = 20\,000$, is not strictly valid. Close re-examination of Fig. 1 shows indeed that the one but lowest eigenvalue has apparently not converged at $k = 20\,000$. Also for $n=100$ most

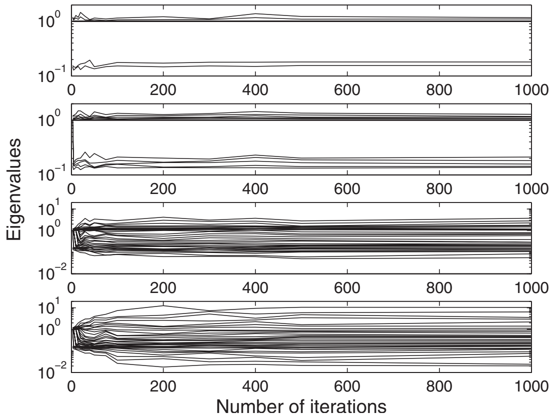

FIG. 5. Eigenvalues of B_k vs iteration index k for $n=30$, where B_k is obtained by L-FSU. From top to bottom $m=2$, 4,20 and 50.

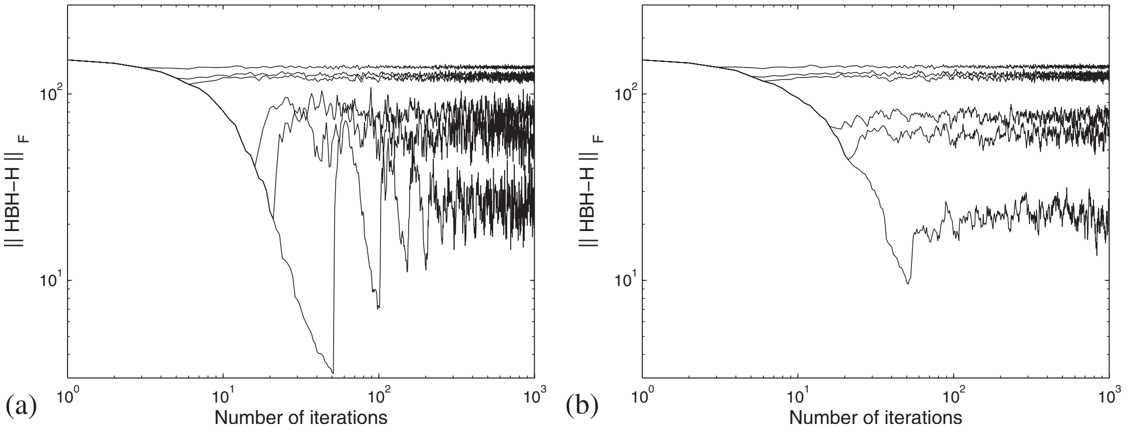

FIG. 4. Comparison of the rate of convergence of the mobility matrix obtained by L-FSU and L-BFGS with various m to the generalized inverse for $n=30$ and the spring potential. The first m updates updates of B_k in L-FSU and FSU coincide and the deflection point is decreasing with increasing m. (a) $\|HB_kH - H\|_{\mathrm{F}}$ vs iteration index k, where B_k is obtained by L-FSU for $m=2$, 4, 5, 15, 20, and 50. (b) $\|HB_kH - H\|_{\mathrm{F}}$ vs iteration index k, where B_k is obtained by L-BFGS for $m=2$, 4, 5, 15, 20 and 50.

intermediate eigenvalues have (almost) reached their steady-state value for rather small k.

2. Regularization

A nonsingular $\tilde{H}$ is obtained for the constrained potential given by

$$\tilde{\Phi} = \Phi + \Phi_{\mathrm{com}}, \tag{23}$$

with $\Phi_{\mathrm{com}}=(c_k-c_0)^2$ and $c_k=c(\mathbf{x}_k)=\Sigma_{i=1}^{N}x_i$. Adding this term to the potential has the effect of conditioning the eigenvalues. We focus on the smallest system from the previous section, $n=27$, for illustration purposes. The eigenvalues of $\tilde{B}$ obtained by FSU (not shown here) level off to constant limiting values for increasing $k(k>1000)$, including $\lambda_{\max}$ that is now bounded. Owing to the nonsingular $\tilde{H}$, we can directly compare the steady-state eigenvalue spectrum of $\tilde{B}$ to the analytic spectrum for $\tilde{H}^{-1}$. In addition, we can compare this analytic spectrum to the steady-state eigenvalue spectrum of B itself [see Fig. 1(b)]. Linear algebra provides the tools for such a direct comparison, in terms of an expression for the change in eigenvalues of a matrix when a rank one matrix is added. Let $\mathbf{u}$ and $\mathbf{v}$ be two n-dimensional vectors and $\mathbf{u}$ an eigenvector of the $n\times n$ matrix A. The eigenvalues of $A+\mathbf{u}\mathbf{v}^T$ are then given by:

$$\sigma(A+\mathbf{u}\mathbf{v}^T)=\{\sigma(A)\setminus\lambda_{\mathbf{u}}\}\cup\{\lambda_{\mathbf{u}}+\mathbf{u}^T\mathbf{v}\}, \tag{24}$$

where $\lambda_{\mathbf{u}}$ is the eigenvalue from corresponding eigenvector $\mathbf{u}$. Taking $A=B^{-1}$, $\mathbf{u}=[1\ldots1]^T$ and $\mathbf{v}^T=[2\ldots2]$, and thus $\mathbf{u}\mathbf{v}^T$ equal to the Hessian of Φ_{com} (H_{com}), we can directly compare the spectrum of $B^{-1}+H_{\mathrm{com}}$ to the analytic spectrum of $\tilde{H}$. For consistency, we plot in Fig. 3 the inverse of the eigenvalues calculated using this relation. We note for completeness that the eigenvalue $\lambda_{\mathbf{u}}$ in Eq. (24) is the previously mentioned $\lambda_{\max}^{-1}$ of B.

From the spectra in Fig. 3, we observe that the smallest eigenvalues of $B(+,k=10\,000)$ and $\tilde{B}(\circ,k=2000)$ coincide, while the largest eigenvalues deviate considerably. The eight largest eigenvalues of B are increasingly exceeding the ones for $\tilde{B}$, and we note that the highest eigenvalue ($\lambda_{max}\approx351$) of B even exceeds the chosen vertical axis limit. Conditioning the eigenvalue spectrum by the penalty function Φ_{com} in Eq. (23) thus only acts on the smallest eigenvalue of $H\sim B^{-1}$, as could be expected. Comparing the spectrum of $\tilde{B}$ ($\circ$) and the analytic spectrum for $\tilde{H}^{-1}$ ($\cdot$) shows that they coincide in detail. We conclude that for $\tilde{\Phi}$, FSU constructs a perfect approximate of the inverse Hessian starting from $k\approx1000$. Finally, we used the relation (24) to project the spectrum of B onto the analytic values for $\tilde{H}^{-1}$ ($\square$). From the perfect match, we conclude again that B^{-1} is an accurate approximate of H, in particular at the considered later k-stages ($k=10\,000$). Finally we carried out an additional FSU simulation for the unconstrained potential Φ. After each k update, the chain center of mass is reset to the original position by an affine translation. The eigenvalue spectrum (denoted by large +) coincides with the analytic values, and we conclude that even with this *ad hoc* regularization procedure FSU generates a very accurate inverse Hessian.

B. L-FSU

From numerical evaluation on a range of test problems, L-BFGS is known to be an efficient general-purpose method for determining optimal solutions in nonlinear problems, i.e., $\mathbf{x}'$ such that $\Phi(\mathbf{x}')$ is optimal, even for relatively small m [20]. L-BFGS is therefore the most popular member of the (convex) Broyden family of limited-memory methods, but the performance of L-DFP is known to be comparable [29]. As far as we know, some properties, like the convergence of B_k to H^{-1}, have not been considered in detail. The convergence to optimal solutions suggests that at least some Hessian information is stored in B. The important issue considered here is which, and how much, Hessian information is stored in B, and which m is optimal in this sense.

We consider L-FSU for $n=30$. The history depth m is varied from small (2, 4, and 5) to larger values (15, 20, and 50). Figure 4(a) shows the (unscaled) residual norm as a function of k and m. For all m, the norm in the initial m steps is equal to the FSU norm (see Fig. 2). Starting from $k=m$, older information is disregarded in favor of new information. For small m ($m=2,4,5$), the residual norm is observed to oscillate around a constant value that is roughly equal to the residual norm of B_m. The effect of truncation is more distinct for larger m and gives rise to large oscillations that damp out while the residual norm reaches an almost constant value. The apparent periodicity of these oscillations shows an intricate effect. Since $n=30$, all eigenvectors/eigenvalues are affected by the update process for $m>15$ and H^- will be increasingly approximated by B_m. As shown before in Fig. 2, the initial stages ($k<n/2$) give rise to a fast decrease of the residual norm relative to the correction at later stages, which is associated with the noisy plateau region. Disregarding the V_k of these initial stages will naturally give rise to a jump in the residual norm. This information will be restored by the iterative process and give rise to apparent periodicity. As the information that can be incorporated in B_k is limited to m vector triplets, also this residual norm will converge to a constant value, but this value may be slightly higher than the residual norm of B_m if the process of reiterating the disregarded information cannot keep up with the truncation process. For this constant Hessian, one could therefore choose to keep $B_k=B_m$ fixed during the iterative process for $k>m$. For general potentials, the Hessian will vary and the B_k obtained by L-FSU will be slaved by the pathway on the energy hypersurface. The k evolution of the eigenvalues of B_k is shown in Fig. 5, for varying m. Except for the eigenvalues=1, originating from $J_0=I$, none of the eigenvalues levels off to a constant value. Instead, and although hidden by the logarithmic scale used in Fig. 5, they show tiny oscillations around a constant value due to the truncation process. It shows the convergence of B_k to a stationary state $\bar{B}(m)$. It is clear that $2m$ eigenvalues differ from unity and that the range of the eigenvalue spectrum increases with increasing m. Since these eigenvalues directly correspond to a *multiplicity* of time steps, we conclude that the multiscale nature is enhanced by considering larger m.

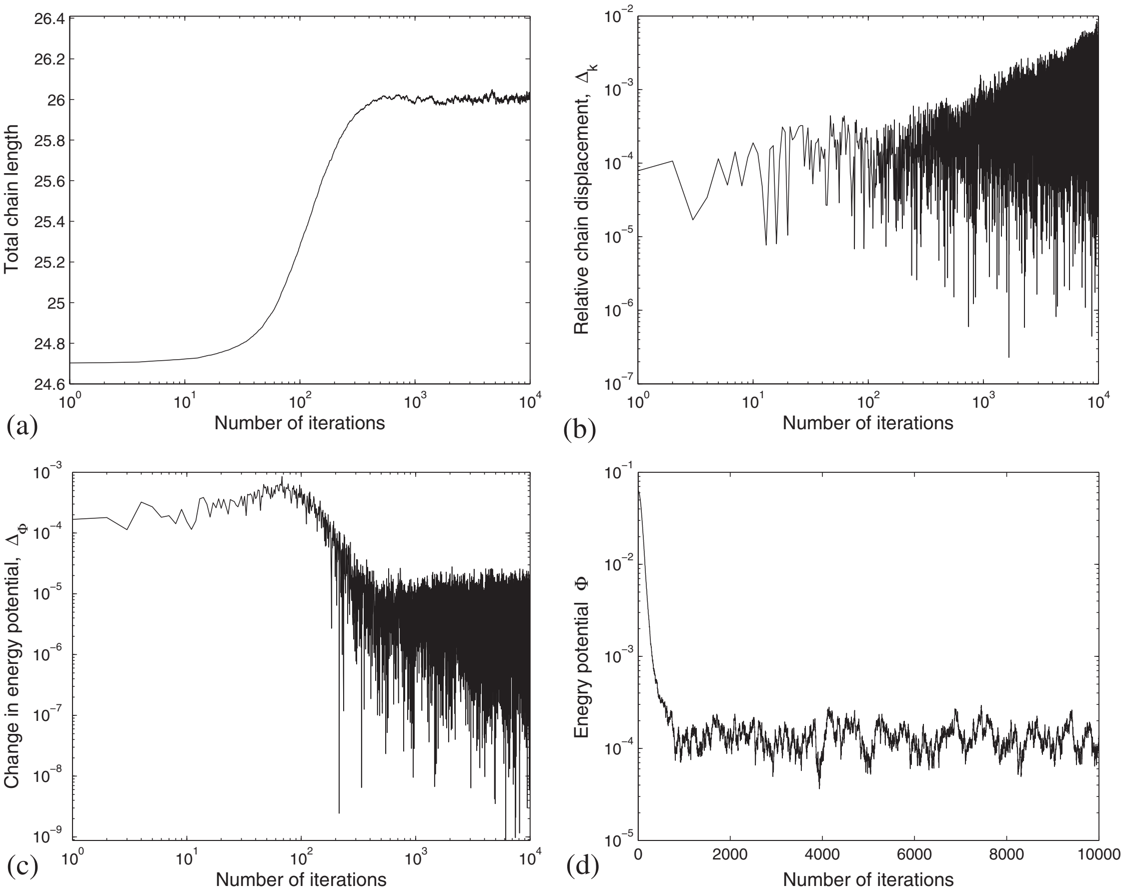

FIG. 6. Sampling properties obtained using FSU for the spring potential with $n=27$ and equidistant initial particle distances; $x_{i,i+1}=0.95$. (a) Total chain length l vs iteration index k. (b) Relative chain displacement Δ_k vs iteration index k. (c) Potential energy difference $\Delta\Phi$ vs iteration index k. (d) Potential energy $\Phi_{(x_k)}$ vs iteration index k.

Finally, we validate the new L-FSU method by considering L-BFGS for the same system and varying m [see Fig. 4(b)]. We used Choleski decomposition for the noise factor. We identify the same type of oscillation around a constant value as in L-FSU for small m [Fig. 4(a)]. Apparently, the L-BFGS scheme is more stable against truncation for large m by the particular form of the update scheme. It only shows that the update schemes are conceptually different. From Fig. 4 and $k<m$, one can conclude that FSU gives a better approximation for the inverse Hessian than the standard BFGS for this particular potential. Nevertheless, the residual norms of the stationary states $\bar{B}(m)$ are roughly equal to the ones obtained from L-FSU.

C. Multiscale simulation

Although some information is provided in the previous sections, we further concentrate on the multiscale nature of FSU and L-FSU. Analogous to effective diffusion, the mobility on larger length and time scales (coordinated movements) can be determined from the root mean square displacement of the chain's center of mass. Throughout the rest of this paper, the simulation variables are set to $\Delta t=0.01$ and $k_BT=10^{-5}$. We have chosen $n=27$, in order to complement the results of the previous section. A relatively small n allows us to concentrate on the role of the adaptive mobility when one could assume that larger modes are less important. At the end of this section, we consider the performance for $n=100$ as well. Another important choice for the sampling behavior is the initial state $\mathbf{x}_0$. For $\mathbf{x}_0$ close (far) from the optimal state $\mathbf{x}'$ of Φ, with $\Phi(\mathbf{x}')=0$, cooperative motion will play a less (more) significant role. We consider two different initial states. In the first scenario, we initially position all particles at a distance $x_{i,i+1}=0.95$, rather close to the equilibrium distance. We note that the topography of the potential energy hypersurface around the optimal state is symmetric, since all spring constants were set to unity.

Starting with FSU, the chain's contour length and the relative chain displacement $\Delta_k=c_k-c_{k-1}$ are shown in Figs. 6(a) and 6(b), respectively. The potential energy difference between successive steps $\Delta\Phi_k=\Phi(\mathbf{x}_k)-\Phi(\mathbf{x}_{k-1})$ and the potential energy value are shown in Figs. 6(c) and 6(d). From Figs. 1 and 2, we concluded that B_k is increasingly approximating H^{-}, while the largest eigenvalue continues to rise approximately linear with k. Figure 6(a) shows that the chain's contour length l rises to the equilibrium length $l_0=26$ very fast due to larger modes in the system. The line intersects with l_0 after ≈500 steps and fluctuates around l_0 at later stages. From Figs. 6(c) and 6(d) one can see that the system has to overcome a small barrier during the equilibration process and that the potential energy hardly changes after the equi-

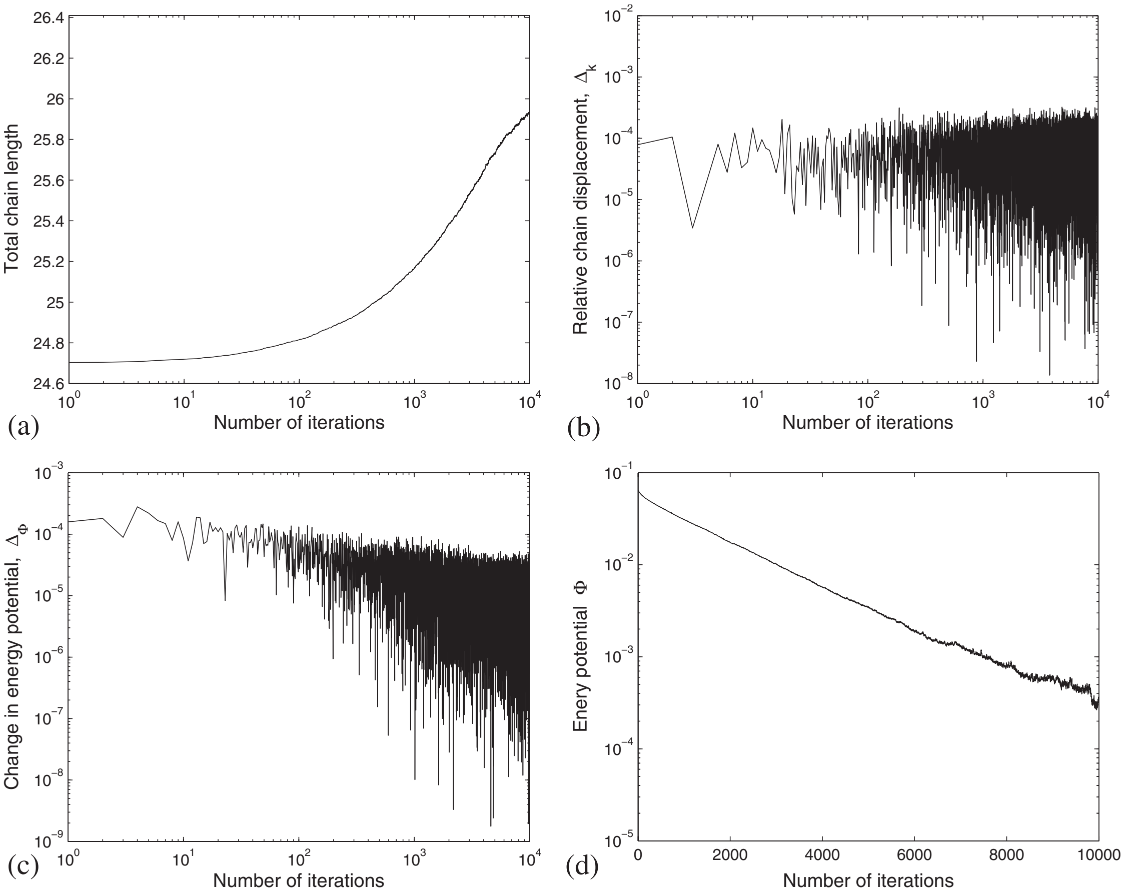

FIG. 7. Sampling properties obtained using conventional Langevin dynamics for the spring potential with $n=27$ and equidistant initial particle distances; $x_{i,i+1}=0.95$. (a)Total chain length l vs iteration index k. (b) Relative chain displacement Δ_k vs iteration index k. (c) Potential energy difference $\Delta\Phi$ vs iteration index k. (d) Potential energy $\Phi_{(x_k)}$ vs iteration index k.

librium contour length is reached. At $k\approx 1000$, the optimal state $\mathbf{x}'$ is reached and Eq. (20) starts sampling the basin around $\mathbf{x}'$ with a decreased and rather small noise amplitude. Apparently, the contour length l_0 is reached before all particles arrive at their equilibrium distances. From Fig. 6(b) shows that Δ_k changes very rapidly between successive steps and that the average Δ_k increases monotonically with k. At later stages, Δ_k becomes much larger than Δt. We conclude from Fig. 6 that the chain moves as a whole during the later stages, due to the increasing λ_{max} associated with displacements in the null-space of H. This chain movement is random, the true hallmark of diffusion.

We also compared the performance of Eq. (20), with J_k determined by FSU, and conventional Langevin dynamics (CLD). The value of the scalar mobility in $B=MI$, resulting from friction due to the surrounding medium, depends to a high degree on the unknown topography of the energy hypersurface. This M constitutes an effective scaling of the time step Δt ($M\Delta t$ for all modes), which is automatic and mode-dependent in Langevin dynamics with curvature-dependent mobility. For the special case considered here, this topography is quite simple since all spring constants=1, and $M=\|\widetilde{H}^{-1}\|_F/\|I\|_F=1/n\|\widetilde{H}^{-1}\|_F=7.4068$ would be a good value, but one need to explicitly include information of the analytic inverse Hessian for the constrained Φ. In general, and even for a Harmonic potential with different spring constants, the determination of an optimal M is much more difficult. In those cases M is determined on physical grounds or by trial simulations in the vicinity of the starting state. The starting point for FSU is $B_0=I$, and we therefore consider $M=1$ here. All motion is local and the noise samples around the steepest descent direction, i.e., the drift term for $B_k=I$. From Fig. 7(a) we find that the contour length l approaches the equilibrium value l_0 but has not reached it at $k=10^4$. The relative chain displacement Δ_k and potential energy difference between successive steps $\Delta\Phi_k$ for CLD are shown in Fig. 7(b) and 7(c), respectively. We find that $\Delta_k \ll \Delta t$ and the average Δ_k is apparently a very small constant. In other words, the chain center of mass moves only marginally due to displacements of individual particles and cooperative displacements are absent. The barrier of Fig. 6(c) is absent in Fig. 7(c), signaling that also the pathway on the potential energy hypersurface is different. The potential energy Φ has almost reached the optimal $\mathbf{x}'$ at $k=10^4$, but the amplitude of the noise is roughly equal during the whole simulation pathway.

For L-FSU, we considered the sampling performance for $m=5$ and $m=50$. Figure 8(a) shows the contour length l, relative chain displacement Δ_k [Fig. 8(b)] and potential energy difference between successive steps $\Delta\Phi_k$ [Fig. 8(c)]. It is clear that for small $m=5$ the sampling performance is considerably reduced compared to FSU. The convergence of l and $\mathbf{x}_k$ to l_0 and $\mathbf{x}'$, respectively, are comparable with CLD

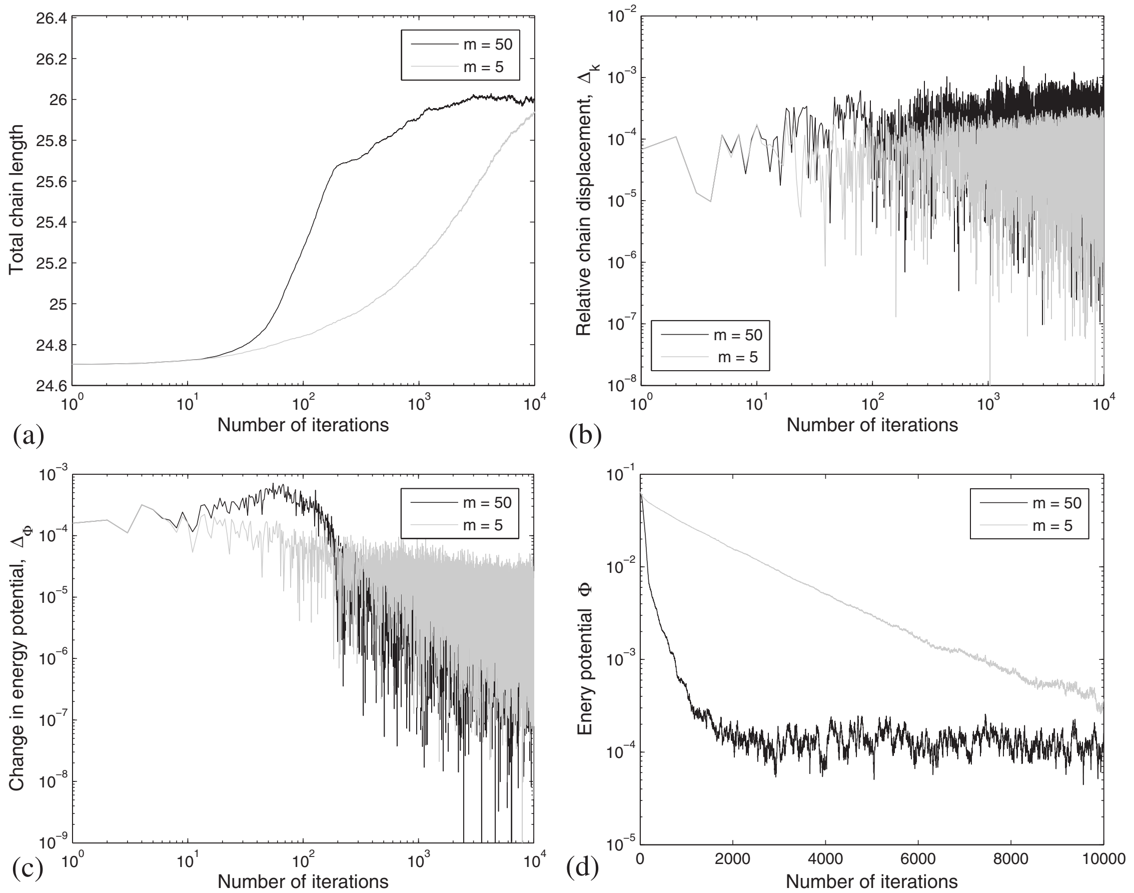

FIG. 8. Sampling properties obtained using L-FSU, $m=5$ and $m=50$, for the spring potential with with $n=27$ and equidistant initial particle distances; $x_{i,i+1}=0.95$. (a)Total chain length l vs iteration index k. (b) Relative chain displacement Δ_k vs iteration index k. (c) Potential energy difference $\Delta\Phi$ vs iteration index k. (d) Potential energy $\Phi_{(x_k)}$ vs iteration index k.

(Figs. 7 and 8). From the analysis of eigenvalues (Fig. 5, for $m=4$) it is clear that the spectrum is very narrow and that the eigenvalues only slightly deviate from unity. In certain parts of the potential hypersurface, the direction of search will therefore be rather similar to the one in steepest descent, and some modes will be accelerated while others will be slightly damped. This effect can also be observed in Δ_k [Fig. 8(b)]. The spread of Δ_k marginally increases with k and the noise amplitude in $\Delta\Phi_k$ decreases only very mildly while approaching the optimal state of the system [Figs. 8(c) and 8(d)]. For larger $m=50$, the performance improves. The equilibrium length l_0 is reached at $k\approx 2600$ and l fluctuates around l_0 at later stages. The optimal state is first found at roughly the same $k\approx 2500$, after which the basin is sampled. The contribution of larger modes in the displacement is reduced compared to FSU due to truncation (compare Figs. 1(a) and 5 for $m=50$). Although not very significant, the spread of Δ_k increases with increasing k and the maximum displacement is larger than for $m=5$ [Fig. 8(b)]. From Fig. 8(c), one can observe that also for $m=50$ the simulation pathway crosses a small barrier. Note again the reduced noise amplitude close to the optimal state.

We shortly consider the sampling performance for $\mathbf{x}_0$ further from the optimal $\mathbf{x}'$, where $x_{i,i+1}$ is randomly chosen between 0.5 and 5. In Fig. 9(a) the contour length l and relative chain displacement Δ_k [Fig. 9(b)] are compared for J_k derived by FSU, $J=I$ (CLD) and J_k derived by L-FSU for $m=5$ and $m=15$. In Fig. 9(a), l can be seen to drop very fast toward l_0 for both FSU and L-FSU, within ≈ 500 steps, and rather independent of m. In the case of L-FSU, the incorporation of curvature information speeds up this process considerably (compare to CLD) even for small m. When the chain has (almost) contracted to the equilibrium contour length l_0, cooperative chain movements in the drift term become less important, with an exception for the movement of the whole chain. The noise term in Eq. (20) gains importance since $\mathbf{x}_k$ is close to the optimal state $\mathbf{x}'$ at $k\approx 500$ in both schemes. As the noise contribution is fairly small ($k_BT=10^{-5}$), the result is a distinct kink in the l-curve. We find that the overall convergence to the optimal state $\mathbf{x}'$ in (L-)FSU is one order of magnitude faster than for CLD. The information in Fig. 9(b) supports this analysis. The Δ_k for FSU is again governed by the increasing λ_{max} or movements of the whole chain. For L-FSU and small m, full-chain movements are considerably damped due to the truncation process. We observe rather large Δ_k in the initial stages for both values of m due to cooperative motion in the drift term. At later stages, when for L-FSU the contribution due to the drift term is reduced and the basin around the optimal state is mainly sampled due to the noise term, the features of Δ_k for L-FSU ($m=5,15$) and CLD only marginally differ. This confirms the earlier findings (see previous paragraphs) that the

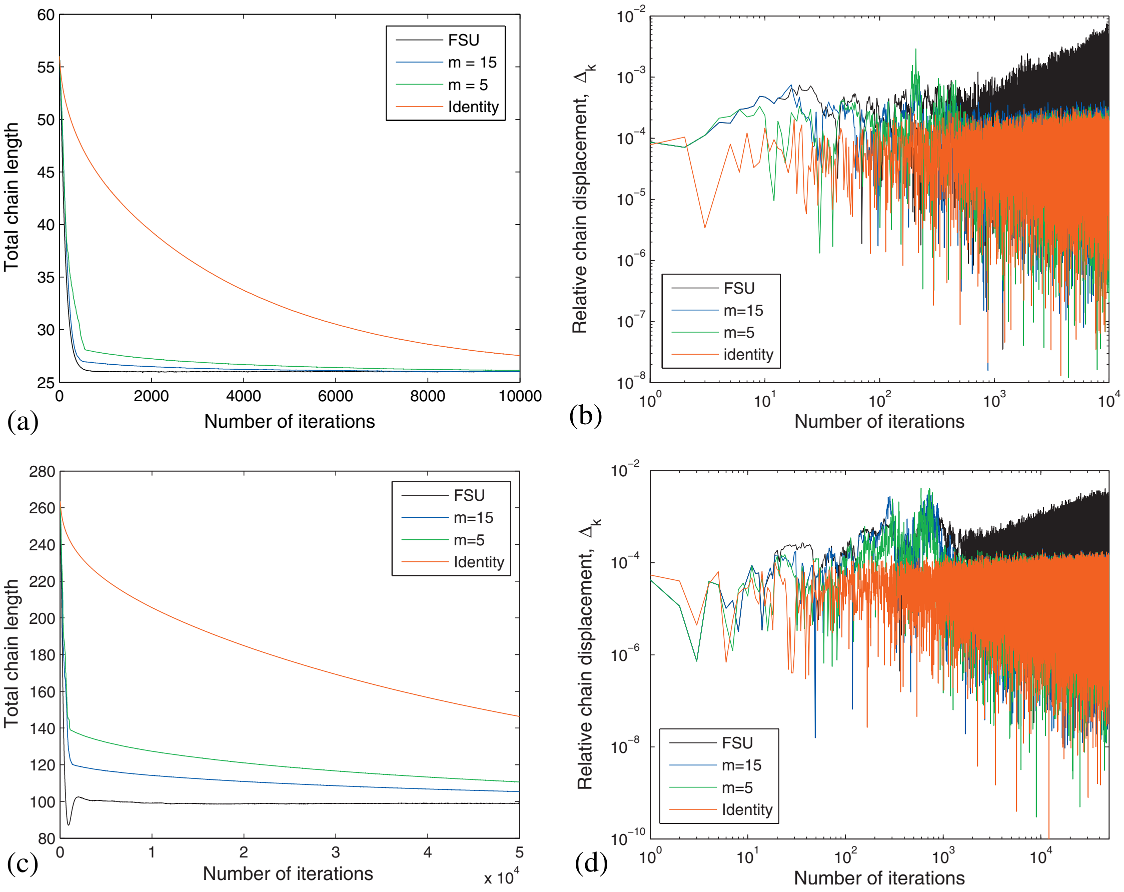

FIG. 9. (Color) Sampling properties obtained using FSU, L-FSU (m=5 and 15) and conventional Langevin dynamics for the spring potential where the particles are initially placed at random interparticle distances between 0.5 and 5. (a)Total chain length l vs iteration index k for n=27. (b) Relative chain displacement Δ_k vs iteration index k for n=27. (c) Total chain length l vs iteration index k for n=100. (d) Relative chain displacement Δ_k vs iteration index k for n=100.

sampling of this basin is rather independent of curvature information. Since all spring constants were set to unity, this behavior could be expected (see discussion of the time step). Finally, we compare the sampling performance of FSU, L-FSU and CLD using the same $\mathbf{x}_0$, i.e., random interparticle distances between 0.5 and 5, for a much larger system, n =100. We find very similar features for the contour length l [Fig. 9(c)] and the relative chain displacement [Fig. 9(d)] compared to n=27 in Figs. 9(a) and 9(b). However, there are also subtle differences. For FSU, the chain length l mildly overshoots the equilibrium length l_0, directly followed by a correction. This overshoot shows the significance of larger modes. When the length of the chain is close to the equilibrium value, the contributions of larger modes to the particle displacements becomes much smaller but do not completely disappear. Due to the amplification of these modes in FSU, the chain continues to shorten fast. When the chain becomes too short, the forces associated with these modes change sign. Consequently, the chain extends and the chain length convergences to the equilibrium value very fast. For L-FSU (m=5 and m=15) the contribution of the larger modes is damped due to truncation. As a result, the fast decrease of l in the initial stages (first 10^3 steps) due to the larger modes is followed by slower contraction when the contribution due to these modes becomes less significant, and we again observe a distinct kink. The details of Fig. 9(d) support this analysis. In particular, the graphs indicate that in the initial stage the relative displacements for L-FSU are considerably larger than for the equivalent simulations with n=27. After this stage, Δ_k for L-FSU and CLD again become rather comparable.

D. Sampling distribution

We have written $J(\mathbf{x}_k)=J_k$ in Eq. (20) so far. In reality, the iterative update scheme for J introduces memory effects and J contains also previous system states, $J_k=J(\mathbf{x}_0,\mathbf{x}_1,\ldots,\mathbf{x}_k)$. The usual derivation for the Fokker-Planck equation from a stochastic differential equation (sde) like Eq. (6) assumes a Markov process, i.e., no memory effects, but it can be shown that this sde implies the Fokker-Planck equation even when the mobility depends on a finite history of earlier states [30,31]. In our case, $B_k=J_kJ_k^T$ always contains a finite history of earlier states, either due to convergence or truncation. More generally, the number of updates will always be finite. We have previously shown that our general sde indeed gives rise to thermodynamically consistent sampling, i.e., the Boltzmann distribution in equilibrium, for low dimensional systems [1]. We note that the spurious drift term in this general sde gives rise to a predictor-corrector scheme that requires

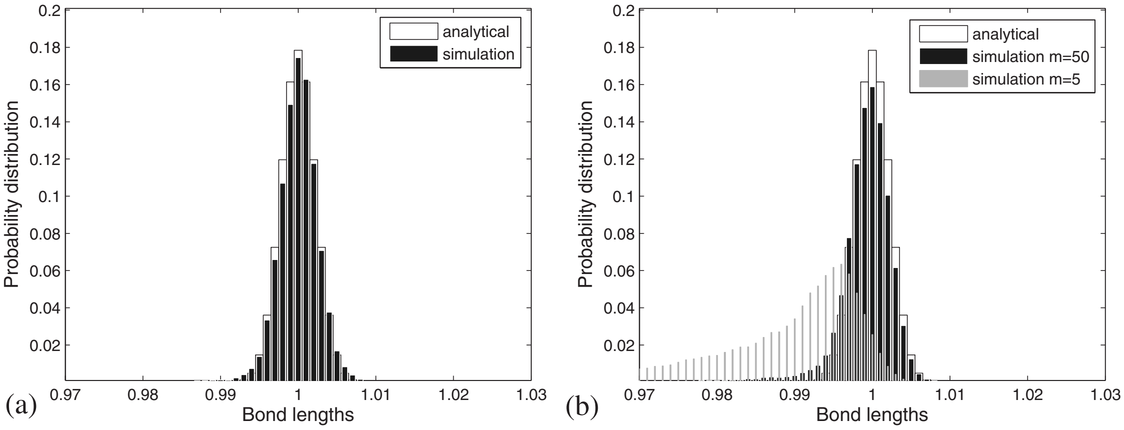

FIG. 10. Bond lengths distributions of the spring obtained using FSU and L-FSU ($m=5$ and 50) for the spring potential with $n=27$ and equidistant initial particle distances; $x_{i,i+1}=0.95$. (a) Distribution of the equilibrium bond lengths according to the Boltzmann distribution and calculated distribution after 10^4 iterations using FSU. (b) Distribution of the equilibrium bond lengths according to the Boltzmann distribution and calculated distribution after 10^4 iterations using L-FSU, $m=5$ and $m=50$.

B_k^{-1} (see Appendix, Sec. 1). However, this general sde reduces to the discrete Eq. (20) for systems with constant mobility. Although the (inverse) Hessian is a constant for the considered systems, the adaptive mobility B_k is only strictly constant when B_k has converged. It seemed, however, justified to always omit the spurious drift term for the considered quadratic potentials, since either B_k converges fast to a constant value or B_k is very slowly varying along the pathway, in which case the effect of spurious drift is minute and can be neglected (see the results of the previous subsections). Here, we consider the validity of the assumption by analyzing the actual sampling distributions for the systems in the preceding subsection.

In Fig. 10, we compare the theoretical equilibrium bond-length sampling distribution (TESD) to the simulated sampling distribution using Eq. (20) for 10^4 steps, unless otherwise mentioned. Since all equilibrium bond-lengths are unity, we expect a single peak centered around one. The J_k was obtained by FSU (Fig. 6) or by L-FSU, with $m=5$ and $m=50$ (Fig. 8). The simulated sampling distribution using FSU in Fig. 10(a) fits the TESD very well. Both the spread and the height match perfectly. For L-FSU we expect that an increased number of iterations are required to obtain close to the equilibrium distribution. For $m=50$ [Fig. 10(b)] it is clear that the simulated distribution converges to the TESD, despite a small remaining shoulder due to the limited number of samples. For $m=5$ this shoulder is more pronounced after 10^4 steps and bond-lengths >1 are clearly undersampled. Increasing the number of samples will reduce the height of the shoulder in favor of a peak centered around one. We conclude that, also in this case, the TESD will be obtained with increased sampling. Finally, we considered the simulated sampling distribution for $\mathbf{x}_0$ further away from the optimal state using L-FSU ($m=15$, see Fig. 9). Figure 11 shows the sampling distribution at different stages: $k=4000$, 6000, and 10^4. Increasing the number of samples in the simulated sampling distribution leads to a shift of the peak value to one and a decrease of the shoulder on the right. We conclude that also this sampling distribution will converge to the TESD.

E. Discussion

1. Inverse Hessian for the unconstrained case

One important issue that is often disregarded in the QN literature is the actual convergence of B_k to the inverse Hessian. As mentioned, the potential Φ may be invariant under particular translations or rotations and H can thus be singular. The mobility B_k may converge to any of the generalized inverses of H, and we should consider $\|HB_kH-H\|_{\mathrm{F}}$ to determine convergence properties. Here, we show that it is instructive to consider $\|B_kH-I\|_{\mathrm{F}}$ nevertheless, after special taken care of the null-space of H. By definition, $B_kH=I$, or alternatively $H=B_k^{-1}$, if $B_kH\mathbf{x}=I\mathbf{x}=\mathbf{x}\ \forall \mathbf{x}\in\mathbb{R}^n$. One can easily see that since $B_kH(\mathbf{x}+c1)=B_kH\mathbf{x}$ for any scalar c, this equality does not have a solution. For the 1D chain considered in the examples, we can resolve this problem. Since the matrix H is Hermitian positive semidefinite, we can write H as $H=U\Sigma U^T$, with U a unitary matrix and Σ a diagonal matrix containing the singular values σ_i $(i=1,\ldots,n)$. The col-

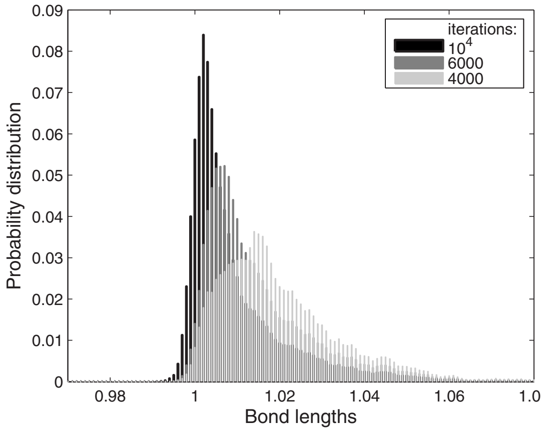

FIG. 11. Evolution of the bond-length distribution at different stages during a simulation with the spring potential using L-FSU, $m=15$ and $n=27$. The particles are initially placed at random interparticle distances between 0.5 and 5.

umns $\mathbf{u}_i(i=1,\ldots,n)$ of U form a orthonormal basis for $\mathbb{R}^n$, i.e., $\mathbf{u}_i^T\mathbf{u}_j=0$ for $i\neq j$ and $=1$ for $i=j$. Let $\mathbf{u}_n$ be the basis vector associated with the singular value 0 (the null space). We note that for our system $\mathbf{u}_n=n^{-1}\mathbf{1}=n^{-1}[1\ldots1]^T$, due to the invariance of the potential Φ under translation of the whole chain. Writing $\mathbf{x}=\Sigma_{i=1}^N a_i\mathbf{u}_i$ one easily see that when we add a matrix $B=[\mathbf{u}_n\mathbf{u}_n\ldots\mathbf{u}_n]^T=n^{-1}[\mathbf{1}\ldots\mathbf{1}]^T$ to the left hand side of $B_kH=I$, i.e., $B_kH+B=I$ or equivalent $(B_kH+B)\mathbf{x}=\mathbf{x}$, $\forall\mathbf{x}\in\mathbb{R}^n$, this equation can have a solution B_k. We can even make B more general by replacing it by $B=(\Sigma_i^n\sigma_i')^{-1}\Sigma'\mathbf{1}_M$, where $\mathbf{1}_M$ is a matrix with all unit elements and Σ' is a diagonal matrix. We note that the n values of σ_i' on the diagonal of Σ' can be freely determined, for instance by the requirement that the diagonal elements in B_kH+B are unity. In the examples, we have therefore always considered $\|B_kHB_k-I\|_{\text{F}}$ for the unconstrained cases, with σ_i' appropriately chosen. Returning to the original issue of the singularity of H, we see that this procedure essentially allows us to determine an *equivalent* B_kH in which the contribution associated with the null space of H, i.e., $\mathbf{u}_n$, is removed from each column of B_kH. In principle, these contributions can also be obtained from $A=U^TB_kU$, by resetting the entries associated with $\mathbf{u}_n$ to zero in a new matrix $\tilde{A}$. We can then compute $\tilde{B}_k=U\tilde{A}U^T$, followed by an evaluation of $\|\tilde{B}_kH-I\|_{\text{F}}$.

2. Accelerated sampling and regularization

The idea of reducing or even eliminating critical slowing down by scaling is also present in the Fourier acceleration method for lattice field theory [14]. The application of Fourier acceleration has been extended to Langevin molecular dynamics [15], where acceleration is accomplished by amplifying the slow modes, after transforming the equation of motion to the Fourier domain. Our proposed update does not require any transformation and decoupling of the modes, which is also not always possible in nontrivial models. From the results in the previous sections, we conclude that collective motion, or an effective multiplicity of time steps, is introduced by the new mobility as determined by FSU. Truncation (L-FSU) leads to a slowing down of the collective modes. Although $m=5$ showed good performance for our particular system, one should take care in determining the optimal history depth m. Since all columns and rows of $\bar{B}(m)$ contain curvature information for $n/2\leq k\leq n$, a good choice is $m=n/2$. Using this m, the computational costs for L-FSU $(6n^2+3n)$ are less than FSU $(7n^2)$ for $n>1$. Using (L-)FSU, we obtain an *automated* treatment of different time and length scales in the system. Although the acceleration due to the translational degrees of freedom can be desired in some cases, we showed that it can easily be reduced or even avoided by introducing translational constraints. In general, the null space related to singular values of H can contain more degrees of freedom than only translation, and it makes sense to constrain them by regularization. We will consider this issue in more detail in a future publication.

3. Time step

Although the time step Δt in S-QN is at first sight equivalent to the step size α_k in QN methods [1], its role is significantly different. The step size α_k in QN depends on hypersurface information and *varies*. Langevin dynamics, however, is a coarse-grained description for the purely diffusive motion of n particles on a time scale that is large enough to integrate out the rapid modes and replace them by friction and noise. Although this concept can be further exploited for a separation of time scales in the coarse-grained description [32], most conventional Langevin simulations use a single constant time step Δt. In particular, there is no *a priori* measure based on which one could vary the time step *during* simulation. The time step Δt is lower bounded by the requirement that these rapid modes in the detailed description are always in local thermodynamic equilibrium, and upper bounded by the fastest modes in the diffusive description, in particular by the requirement that these modes are sampled with the correct canonical distribution. In our examples section, we have also used a fixed Δt. The question remains: what is a good value for Δt in S-QN in terms of accurate and efficient sampling? Analysis for a single one-dimensional harmonic oscillator, $\Phi=\frac{k}{2}x^2$, with spring constant k provides some insight. An analysis of the time scales involved in the problem using conventional Langevin dynamics gives

$$\frac{d\langle\mathbf{x}\rangle}{dt}=-k\langle\mathbf{x}\rangle \tag{25}$$

with $\langle\cdot\rangle$ the ensemble average. We have scaled the time by setting the friction coefficient η to unity. Equation (25) shows that when a harmonic degree of freedom is disturbed from equilibrium, it will relax exponentially with a characteristic time k^{-1}. Using S-QN for the same system, i.e., $d\langle\mathbf{x}\rangle/dt=-\langle\mathbf{x}\rangle$, shows that the characteristic time for relaxation that is now the same for any spring constant k, i.e., accelerated for $k<1$ and slowed down for $k>1$ with respect to Eq. (25). Further simple analysis of the stability conditions for the discrete time step in Eq. (20) shows that a stable scheme is obtained for $\Delta t<1$. In comparison, the conventional Langevin equation is stable for $\Delta t<k^{-1}$. The same analysis can be carried out for coupled harmonic oscillators and provides similar results [33]. We note, however, that we have assumed $B=H^{-1}$ and that the analysis is linear. For systems where the higher moments of Φ are significant, a large time step may result in numerical instability and the optimal time step should be determined by numerical testing. Also in the early stages for a quadratic Φ, when the inverse Hessian information stored in B_k is still very sparse, a large time step may give rise to erratic behavior and increased violations of the curvature condition. Since violation of the curvature condition leads to a truncation of updating local Hessian information in B_k, this is very undesired from an algorithmic point of view. In the numerical testing, we found that a large time step (but still $\Delta t<1$) can indeed initially give rise to features of instability, but that proper behavior is automatically restored at later stages, when the inverse Hessian in increasingly approximated by B_k. We therefore conclude that the time step Δt can in general be much larger than in conventional Langevin dynamics (see also results section). The incorporation of backtracking, similar to QN, could provide a solution to the mentioned problems, but will seriously affect the computationally efficiency and is, if restricted to part of the sampling pathway, rather *ad hoc*. To resolve the time step issue in the early stages of simulation, it is much simpler to build Hessian information by local sampling, i.e., using Eq. (20) for $\nabla\Phi=0$ (only noise), and to input the obtained J as J_0 in the FSU method. We believe that the main benefit of our approach originates from introducing a genuine *multiplicity of time steps*, i.e., different scaling for modes in the system depending on information of the inverse Hessian.

4. Other energy landscapes

The considered system represents a special case, since H^{-1} is constant. For such systems, the properties of B_k are independent of the sampling part of the energy landscape. For proteins, the potential energy landscape is more complex and commonly referred to as the folding funnel. In particular, it is locally rugged but with an overall funnel shape of the low-energy part, so that most initial conditions are driven toward a "natural" native state [34]. One may argue that the fast degrees of freedom, such as bond lengths and angles, can be reasonably well described by a harmonic potential energy with different spring constants. However, this is typically not true for all degrees of freedom in a Langevin description. The inverse Hessian will in general significantly change along the sampling pathway, and the approximate B_k has to follow this change accordingly. Consequently, the typical number of steps $k(B)$ for which this memory function can adapt itself to this new situation becomes important. Since the sampling density of the energy landscape is not uniform, the rate $\tau=k(B)/k(H^{-1})$ rather than $k(B)$ itself provides a good estimate, where $k(H^{-1})$ is the typical number of steps in which H^{-1} changes significantly. We are aware that the definition of these variables is unprecise, but they suffice for our current discussion. We conclude that for $\tau \gg 1$, the incorporation of new Hessian information falls behind, and we expect suboptimal B_k. For the considered harmonic system, $k(B)$ can be seen as the rate of convergence, which depends on the chosen threshold value for the difference norm, see the examples section $[k(B)\sim n^3]$. Although $k(H^{-1})$ is infinite in our case, and τ is therefore always smaller than 1, the value of $k(B)$ for a piecewise quadratic Φ may be taken much smaller. Most eigenvalues/eigenvectors of B_k are reasonably approximated in the earlier stages, long before the drop of $\|HB_kH-H\|_F$, and one may argue that the actual $k(B)\sim n$. Our results for $n=27$ and $m=5$ support this view.

Truncating the history in the L-FSU scheme for constant H^{-1} can have a profound effect, as discarded older information has to be *reconstructed* by the new iterate(s). This is most apparent for small n, where convergence is fast and the majority of information about the inverse Hessian is stored in only a few V_k. Discarding such matrices V_k gives rise to a loss of information, to an extent that the value of m can be detected from periodic steps in the difference norm. The periodicity stems from the periodic removal and restoration of information, since the discarded information is swiftly recovered in the next few steps. For varying H^{-1}, however, memory of unrelated parts in the sampling pathway *should* be removed from B_k, and thus L-FSU for a "good" choice of m could be more appropriate than FSU and lead to a reduced value of k_B. Since this property only applies for more complex energy landscapes, we will consider this issue in greater detail elsewhere.

IV. CONCLUSION

We have constructed a factorized secant update (FSU) for the adaptive compound mobility matrix B in an existing stochastic quasi-Newton (S-QN) method [1]. The S-QN method is related to the conventional Langevin equation but differs by the fact that curvature information is contained in the mobility via the inverse Hessian, thereby enabling an automated separation of time scales for the different modes in the system. By updating the factorized J instead of $B=JJ^T$ itself, we avoid the cost expensive Choleski factorization for the noise term in the original scheme and B is always positive definite. The computational costs of FSU are restricted to $7n^2$ multiplications per time step for updating J. In particular, FSU does not require additional evaluations of the potential of interest Φ. For very large n, a limited-memory (L-FSU) update method was derived that allows the user to limit both computational and memory requirements. The approach is based on truncating the memory to m previous updates, similar to the approach in L-BFGS, and requires $12nm+3n$ multiplications and no matrix storage. The arithmetics of the new L-FSU method is found to be even more optimal than the L-BFGS method. The recursive scheme in L-FSU has the additional advantage to FSU that the initial matrix J_0 is isolated from the rest of the computations, allowing this matrix to be chosen differently in every iteration.

We have in detail evaluated the FSU and L-FSU method for a simple but appropriate 1D system of N particles connected by harmonic springs. This system has the advantage of a known analytic Hessian H and involves multiple length and times scales. We analyzed FSU and L-FSU in terms of convergence of B_k to the inverse Hessian, multiscale sampling performance and equilibrium sampling distribution. We found that the B_k determined by FSU converges to a generalized inverse H^- (since H is singular) within $k\sim n^3$ steps. Due to the truncation, the matrix B_k from L-FSU converges to a stationary $\bar{B}(m)$. Analysis of the eigenvalue spectra of B_k suggests that a reasonable good approximate of H^- in FSU is obtained for $k\sim n$. From the analysis of the sampling performance we find that the adaptive mobility indeed gives rise to automated separation of time and length scales. Collective motions, captured by the J_k in both FSU and L-FSU, lead to at least an order of magnitude faster convergence to the lowest energy potential compared to conventional Langevin dynamics. Although for a quadratic Φ good performance was observed for $m=5$, our analysis suggests that $m=n/2$ is optimal in L-FSU for general Φ. The theoretical (Boltzmann) equilibrium distribution was obtained as a limiting case for both FSU and L-FSU.

ACKNOWLEDGMENTS

We thank Simon de Leeuw for reading the paper prior to submission and his valuable suggestions. We thank an anonymous referee for his critical remarks that proved valuable for improving the clarity of presentation.

APPENDIX

1. Predictor-corrector scheme for the spurious drift term

The generalized S-QN equation is given by

$$d\mathbf{x} = [-B(\mathbf{x})\,\boldsymbol{\nabla}\,\Phi(\mathbf{x}) + k_BT\,\boldsymbol{\nabla}\,\cdot B(\mathbf{x})]dt + \sqrt{2k_BT}J(\mathbf{x})dW(t). \tag{A1}$$

We have previously shown [1] that Eq. (A1) can be discretized using the predictor-corrector scheme introduced by Hütter and Öttinger [37] as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k, \tag{A2}$$

$$\Delta\mathbf{x}_k = -\frac{1}{2}[B(\mathbf{x}_k + \Delta\mathbf{x}_k^p)\,\boldsymbol{\nabla}\,\Phi(\mathbf{x}_k + \Delta\mathbf{x}_k^p) + B(\mathbf{x}_k)\,\boldsymbol{\nabla}\,\Phi(\mathbf{x}_k)]\Delta t$$
$$+ \frac{1}{2}[B(\mathbf{x}_k + \Delta\mathbf{x}_k^p)B^{-1}(\mathbf{x}_k) + I]\sqrt{2k_BT}J(\mathbf{x}_k)\Delta W_t,\Delta\mathbf{x}_k^p\Delta,$$

$$\Delta\mathbf{x}_k^p = -B(\mathbf{x}_k)\,\boldsymbol{\nabla}\,\Phi(\mathbf{x}_k)\Delta t + \sqrt{2k_BT}J(\mathbf{x}_k)\Delta W_t, \tag{A3}$$

where Eq. (A3) is the predictor step and Eq. (A2) is the correction. Direct inversion of B would be costly and should therefore be avoided. Using the Sherman-Morrison theorem, the exact inverse $G_k = G(\mathbf{x}_k) = B^{-1}(\mathbf{x}_k)$ of $B(\mathbf{x}_k)$ in dual space can be calculated explicitly by

$$G_k = \left(I - \frac{\mathbf{y}_{k-1}\mathbf{s}_{k-1}^T}{\mathbf{y}_{k-1}^T\mathbf{s}_{k-1}}\right)G_{k-1}\left(I - \frac{\mathbf{s}_{k-1}\mathbf{y}_{k-1}^T}{\mathbf{y}_{k-1}^T\mathbf{s}_{k-1}}\right) + \frac{\mathbf{y}_{k-1}\mathbf{y}_{k-1}^T}{\mathbf{y}_{k-1}^T\mathbf{s}_{k-1}} \tag{A4}$$

reusing the vectors $\mathbf{y}_{k-1}$ and $\mathbf{s}_{k-1}$ stored for updating B_{k-1}. Disregarding the costs associated with the computation of $\boldsymbol{\nabla}\Phi(\mathbf{x}_k + \Delta\mathbf{x}_k^p)$ and the storage of G, we can calculate the costs of this predictor-corrector scheme employed for general Φ. For quadratic potentials, when the predictor Eq. (A3) suffices and $\Delta\mathbf{x}_k = \Delta\mathbf{x}_k^p$, the total costs are $7n^2$ (see the theory section). Due to the very related structure, the corrector Eq. (A2) is $7n^2$ as well (if we reuse terms) plus an additional $2n^2$ for $B^{-1}(\mathbf{x}_k)$ using Eq. (A4). The additional costs for Eq. (A2) are thus $9n^2$ and the total costs for the predictor-corrector scheme using FSU are $16n^2$. The Sherman-Morrison theorem can also be applied to derive an analytic expression for $D_k = J_k^{-1}$ from Eq. (16), providing an efficient method for determining $B^{-1}(\mathbf{x}_k)$ for L-FSU. Again, the total costs of the full scheme are roughly doubled compared to using only the predictor term. Since this calculation is straightforward but involved, the full technical details are given in future publications for general Φ. As a concluding remark, we note that the calculation of the divergence itself may actually be more efficient than the predictor-corrector scheme, because of the special nature of the update $B(\mathbf{x}_k) = B(\mathbf{x}_{k-1}) + V$, with V a rank-two correction.

2. Derivation of the FSU algorithm

The derivation of the update for J is equivalent to the update for the lower triangular matrix L [17]. By interchanging $\mathbf{s}$ and $\mathbf{y}$ and replacing L with J, the matrices LL^T and JJ^T become approximates of the Hessian and the inverse Hessian, respectively. Here we focus on the derivation of the update scheme for J.

Given $\|\cdot\|$ is the Frobenius norm and

$$\min_{J_{k+1}}\|J_{k+1} - J_k\|, \tag{A5}$$

$$J_{k+1}\mathbf{v}_k = \mathbf{s}_k, \tag{A6}$$

J_{k+1} is uniquely given by

$$J_{k+1} = J_k + \frac{\mathbf{s}_k - J_k\mathbf{v}_k}{\mathbf{v}_k^T\mathbf{v}_k}\mathbf{v}_k^T. \tag{A7}$$

Substitute J_{k+1} into

$$J_{k+1}^T\mathbf{y}_k = \mathbf{v}_k, \tag{A8}$$

gives

$$\mathbf{v}_k = J_{k+1}^T\mathbf{y}_k = \left(J_k + \frac{\mathbf{s}_k - J_k\mathbf{v}_k}{\mathbf{v}_k^T\mathbf{v}_k}\mathbf{v}_k^T\right)^T\mathbf{y}_k,$$

$$= J_k^T\mathbf{y}_k + \frac{(\mathbf{s}_k - J_k\mathbf{v}_k)^T\mathbf{y}_k}{\mathbf{v}_k^T\mathbf{v}_k}\mathbf{v}_k \tag{A9}$$

$$\Rightarrow\left(1 - \frac{(\mathbf{s}_k - J_k\mathbf{v}_k)^T\mathbf{y}_k}{\mathbf{v}_k^T\mathbf{v}_k}\right)\mathbf{v}_k = J_k^T\mathbf{y}_k. \tag{A10}$$

Hence, $\mathbf{v}_k = \alpha_k J_k^T\mathbf{y}_k$ and after substituting this into Eq. (A9) gives

$$\alpha_k^2 = \frac{\mathbf{y}_k^T\mathbf{s}_k}{\mathbf{y}_k^TJ_kJ_k^T\mathbf{y}_k}, \tag{A11}$$

which has a real solution for α_k due to the curvature condition and positive definiteness of $J_kJ_k^T$. The update scheme for J_{k+1} is now given by

$$J_{k+1} = J_k + \frac{\alpha_k\mathbf{s}_k\mathbf{y}_k^TJ_k - \alpha_k^2J_kJ_k^T\mathbf{y}_k\mathbf{y}_k^TJ_k}{\mathbf{y}_k^T\mathbf{s}_k}. \tag{A12}$$

Using this update we find after some algebraic operations that JJ^T is equal to the update derived from the DFP scheme

$$J_{k+1}J_{k+1}^T = J_kJ_k^T - \frac{J_kJ_k^T\mathbf{y}_k\mathbf{y}_k^TJ_kJ_k^T}{\mathbf{y}_k^TJ_kJ_k^T\mathbf{y}_k} + \frac{\mathbf{s}_k\mathbf{s}_k^T}{\mathbf{y}_k^T\mathbf{s}_k},$$

$$= B_k - \frac{B_k\mathbf{y}_k\mathbf{y}_k^TB_k}{\mathbf{y}_k^TB_k\mathbf{y}_k} + \frac{\mathbf{s}_k\mathbf{s}_k^T}{\mathbf{y}_k^T\mathbf{s}_k}. \tag{A13}$$

3. Limited-memory update scheme

We consider our L-FSU method in the framework of limited-memory approaches. To arrive at a limited-memory

BFGS method, two different strategies have been used. The L-BFGS method of Liu and Nocedal [20] recasts BFGS into a multiplicative form $B_{k+1}=V_k^T B_k V_k+\rho_k s_k \mathbf{s}_k^T$, and truncates by only using the information stored in V_k and $\mathbf{s}_k$ during the last m updates. In particular, given a (often diagonal) B_0, the L-BFGS update is provided by

$$\begin{aligned} B_{k+1} = {} & (V_k^T \dots V_{k-m+1}^T)B_0(V_{k-m+1}\dots V_k) \\ & + \rho_{k-m+1}(V_k^T\dots V_{k-m+2}^T)\mathbf{s}_{k-m+1}\mathbf{s}_{k-m+1}^T(V_{k-m+2}\dots V_k) \\ & + \rho_{k-m+2}(V_k^T\dots V_{k-m+3}^T)\mathbf{s}_{k-m+2}\mathbf{s}_{k-m+2}^T(V_{k-m+3}\dots V_k) \\ & + \dots + \rho_{k-1}V_k^T\mathbf{s}_{k-1}\mathbf{s}_{k-1}^T V_k + \rho_k s_k \mathbf{s}_k^T. \end{aligned} \tag{A14}$$

This approach was recently generalized by Reed [29] for the convex Broyden family of Quasi-Newton updates. The variable storage conjugate gradient (VSCG) method of Buckley and LeNir [35] is based on the BFGS formula in the additive form and overwrites the most recent update once m is reached. If only the current update is stored, both algorithms reduce to the memoryless QN method of Shannon and Phua [36]. It is generally recognized that L-BFGS with Shanno scaling is the most efficient and reliable method across a range of test problems.

We can rewrite the update scheme for J_{k+1} in Eq. (16) as $J_{k+1}=V_kJ_k=(\Pi_{j=0}^{k}V_{k-j})J_0$ with

$$V_k = \left(I - \frac{1}{\nu_k}\mathbf{v}_k\mathbf{y}_k^T\right) \tag{A15}$$

with $\mathbf{v}_k=\mathbf{h}_k-\mathbf{s}_k/\alpha_k$, $\mathbf{h}_k=J_kJ_k^T\mathbf{y}_k$ and $\nu_k=\mathbf{h}_k^T\mathbf{y}_k$. Using the additional condition, $B_{k+1}=J_{k+1}J_{k+1}^T$, we obtain

$$B_{k+1}=J_{k+1}J_{k+1}^T = V_kV_{k-1}\dots V_0J_0J_0^TV_0^T\dots V_{k-1}^TV_k^T \tag{A16}$$

$$=V_kJ_kJ_k^TV_k^T = V_kB_kV_k^T. \tag{A17}$$

Rewriting this expression in the additive form, several terms cancel and we obtain exactly the additive Davidon-Fletcher-Powell (DFP) formula (see also Appendix, Sec. 2) [29]. Hence, the multiplicative DFP formula

$$B_{k+1}=V_k^TB_kV_k+\rho_k\mathbf{s}_k\mathbf{s}_k^T \quad \text{with} \quad V_k=\left(I-\frac{1}{\nu_k}\mathbf{y}_k\mathbf{h}_k^T\right) \tag{A18}$$

and the update scheme in FSU are equivalent. The principle difference is that we casted Eq. (A18) into a *factorized* form Eq. (A17). The recursive expression (A16), obtained by loop unrolling, can serve as a basis for limited-memory implementation. The recursive algorithm also allows for a limitation of the memory requirements of FSU, by storing at each k step vectors $\{\mathbf{y}_k,\mathbf{s}_k,\mathbf{h}_k\}$ instead of matrices J_k and B_k in the original scheme [Eq. (A12)], however, at the expense of an additional computational load.

Since Eq. (16) is multiplicative, we adapt the L-BFGS strategy for limited-memory implementation of FSU (L-FSU). However, instead of truncating the incorporation of V_k in B, we truncate in J, i.e.,

$$J_{k+1}=V_kV_{k-1}\dots V_{k-m+1}J_0 \tag{A19}$$

for $k\geq m$, and apply the second relation to update the mobility B

$$J_{k+1}J_{k+1}^T = V_kV_{k-1}\dots V_{k-m+1}J_0J_0^TV_{k-m+1}^T\dots V_{k-1}^TV_k^T, \quad \text{for} \quad k\geq m. \tag{A20}$$

For $k<m$, the FSU relations apply. Upon comparing L-FSU to L-BFGS in Eq. (A14), with V_k as in Eq. (A18), we note three important properties: (a) L-FSU is factorized, (b) the memory requirements of L-FSU are the same as in L-BFGS, (c) assuming $B_0=I$, the number of matrix-vector products in L-FSU $(2m)$ is of a different order than L-BFGS $[2m+m(m-1)$, or $2m+m(m-1)/2$ if case of reusing information]. One remaining issue is whether the secant condition is satisfied by L-FSU for $k\geq m$.The L-Broyden family [29] was specially designed to satisfy the secant condition $B_{k+1}\mathbf{y}_k=\mathbf{s}_k$ for all k, since $V_k\mathbf{y}_k=0$. By construction, the L-FSU method satisfies the secant condition for $k<m$. Let $m>1$ and $k\geq m$, we define a matrix $\tilde{B}_k=\tilde{J}_k\tilde{J}_k^T$ by

$$\tilde{J}_k=V_{k-1}\dots V_{k-m+1}J_0 \tag{A21}$$

and we find that

$$B_{k+1}\mathbf{y}_k=J_{k+1}J_{k+1}^T\mathbf{y}_k=\alpha_k(\tilde{\mathbf{h}}_k-\beta_k\mathbf{h}_k)+\beta_k\mathbf{s}_k. \tag{A22}$$

with $\tilde{\mathbf{h}}_k=\tilde{B}_k\mathbf{y}_k$ and $\beta_k=\tilde{\mathbf{h}}_k^T\mathbf{y}_k/\mathbf{h}_k^T\mathbf{y}_k$. Consequently, the secant condition is satisfied only when $\tilde{\mathbf{h}}_k=\mathbf{h}_k=J_kJ_k^T\mathbf{y}_k$, which is generally not the case. We now redefine V_k as

$$V_k=\left[I-\frac{1}{\tilde{\mathbf{h}}_k^T\mathbf{y}_k}\left(\tilde{\mathbf{h}}_k-\frac{\mathbf{s}_k}{\tilde{\alpha}_k}\right)\mathbf{y}_k^T\right] \tag{A23}$$

with $\tilde{\alpha}_k^2=\mathbf{s}_k^T\mathbf{y}_k/\tilde{\mathbf{h}}_k^T\mathbf{y}_k$. Substituting this into Eq. (A20) gives

$$J_{k+1}J_{k+1}^T\mathbf{y}_k=V_k\tilde{B}_kV_k^T\mathbf{y}_k=\tilde{\alpha}_kV_k\tilde{B}_k\mathbf{y}_k=\tilde{\alpha}_kV_k\tilde{\mathbf{h}}_k=\mathbf{s}_k \tag{A24}$$

and the secant condition is again satisfied. We note that only the $\mathbf{h}_k$ for $k\geq m$ are affected by this redefinition of V_k.

4. Recursive scheme for the limited-memory update

The update scheme can be casted into **Algorithm 1**.

$$\mathbf{d}=\mathbf{d}(\mathbf{x}_{K+1}); \tag{A25}$$

$$\left\{\begin{array}{l} \textbf{for} \quad i=K,\dots,\max(0,K-m+1) \\ \mathbf{v}_i=\mathbf{h}_i-\mathbf{s}_i/\alpha_i; \\ \lambda_i=\mathbf{v}_i^T\mathbf{d}; \\ \mathbf{d}=\mathbf{d}-(\lambda_i/\mathbf{h}_i^T\mathbf{y}_i)\mathbf{y}_i; \\ \text{end} \end{array}\right. \tag{A26}$$

$$\mathbf{d}=J_0J_0^T\mathbf{d}; \tag{A27}$$

$$\begin{cases} \textbf{for} \;\; i = \max(0, K-m+1), \ldots, K \\ \beta_i = \mathbf{y}_i^T \mathbf{d}; \\ \mathbf{d} = \mathbf{d} - (\beta_i / \mathbf{h}_i^T \mathbf{y}_i) \mathbf{v}_i; \\ \textbf{end} \end{cases} \tag{A28}$$

$$\textbf{stop with result } \mathbf{d} = J(\mathbf{x}_{K+1}) J(\mathbf{x}_{K+1})^T \mathbf{d}. \tag{A29}$$

It is clear that for $K=k$, the procedure in Algorithm 1 provides the drift term in Eq. (6) for $\mathbf{d}=\mathbf{d}(\mathbf{x}_{k+1})=-\nabla\Phi(\mathbf{x}_{k+1})\Delta t$ in Eq. (A25). The noise term can be calculated using the second part of Algorithm 1, starting with Eq. (A27) and $\mathbf{d} = \sqrt{2k_B T J_0} \Delta W_t$. For $k<m$, the vector $\mathbf{h}_k = J_k J_k^T \mathbf{y}_k$ can also be obtained using Algorithm 1 by setting $\mathbf{d}=\mathbf{y}_k$ and $K=k-1$. Consequently, we obtain α_k from

$$\alpha_k = \alpha_k(\mathbf{h}_k) = \sqrt{\frac{\mathbf{s}_k^T \mathbf{y}_k}{\mathbf{h}_k^T \mathbf{y}_k}} \tag{A30}$$

and store this new value α_k in a vector α. For $k \geq m$, $\tilde{\mathbf{h}}_k = \tilde{B}_k \mathbf{y}_k$ can be obtained from Algorithm 1 starting with $\mathbf{d} = \mathbf{y}_k$ with the recursive index running between $k-1$ to $k-m+1$. We store $\alpha_k = \alpha_k(\tilde{\mathbf{h}}_k)$ and $\mathbf{h}_k = \tilde{\mathbf{h}}_k = \mathbf{d}$. This scheme requires only permanent storage of vector triplets $\{\mathbf{s}_k, \mathbf{y}_k, \mathbf{h}_k\}$ (each of length n) for each iteration step k. In agreement with general practice the small additional effort for storing and calculating the vector α of length m is not considered in the analysis [24].

Upon analyzing the computational load, operations Eqs. (A26) and (A28) add up to $3mn$ and $2mn$ multiplications, respectively. An additional n operations are needed for Eq. (A27), if we assume J_0 is a diagonal (positive definite) matrix, giving rise to $5mn+n$ operations. Recursive calculation of $\mathbf{h}_k$ requires a maximum of $5mn+n$ operations (for $k=m-1$), and slightly less for other k. The total is a maximum of $10mn+2n$ multiplications per step for the drift term only. For the noise term only the second part of the algorithm is required. Assuming again a diagonal J_0, we find that n multiplications are required for $\sqrt{2k_B T J_0} \Delta W_t$ and $2mn$ multiplications for Eq. (A28). This brings us to a total of $2mn+n$ multiplications for the noise term, and a total of $12nm+3n$ for the complete cycle at time step k.

[1] C. D. Chau, G. J. A. Sevink, and J. G. E. M. Fraaije, J. Chem. Phys. **128**, 244110 (2008).

[2] B. Dünweg, in *Accelerated Algorithms 2, Computer Simulations of Surfaces and Interfaces*, Proceedings of the NATO Advanced Study Institute, edited by B. Dünweg, D. P. Landau, and A. Milchev (Kluwer Academic Publishers, Dordrecht, Boston, London, 2003).

[3] U. H. E. Hansmann, Chem. Phys. Lett. **281**, 140 (1997).

[4] Y. Sugita and Y. Okamoto, Chem. Phys. Lett. **314**, 141 (1999).

[5] E. Marinari and G. Parisi, EPL **19**, 451 (1992).

[6] P. Liu, B. Kim, R. A. Friesner, and B. J. Berne, Proc. Natl. Acad. Sci. U.S.A. **102**, 13749 (2005).

[7] N. Nakajima, J. Higo, A. Kidera, and H. Nakamura, Chem. Phys. Lett. **278**, 297 (1997).

[8] F. Wang and D. P. Landau, Phys. Rev. Lett. **86**, 2050 (2001).

[9] A. Laio and M. Parrinello, Proc. Natl. Acad. Sci. U.S.A. **99**, 12562 (2002).

[10] A. F. Voter, J. Chem. Phys. **106**, 4665 (1997).

[11] B. Space, H. Rabitz, and A. Askar, J. Chem. Phys. **99**, 9070 (1993).

[12] M. E. Tuckerman, B. J. Berne, and A. Rossi, J. Chem. Phys. **94**, 1465 (1991).

[13] G. Zhang and T. Schlick, J. Chem. Phys. **101**, 4995 (1994).

[14] G. G. Batrouni, G. R. Katz, A. S. Kronfeld, G. P. Lepage, B. Svetitsky, and K. G. Wilson, Phys. Rev. D **32**, 2736 (1985).

[15] F. J. Alexander, B. M. Boghosian, R. C. Brower, and S. R. Kimura, Phys. Rev. E **64**, 066704 (2001).

[16] B. Dünweg, private communication.

[17] J. E. Dennis Jr. and Robert B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1983); reprinted as *Classics Appl. Math. 16* (SIAM, Philadelphia, PA, 1996).

[18] D. Goldfarb, Math. Comput. **30**, 796 (1976).

[19] K. W. Brodlie, A. R. Gourlay, and J. Greenstadt, J. Inst. Math. Appl. **11**, 73 (1973).

[20] D. C. Liu and J. Nocedal, Math. Program. **45**, 503 (1989).

[21] P. J. M. Laarhoven and E. H. L. Aarts, *Simulated Annealing, Theory and Applications* (D. Reidel Publishing Company, Dordrecht, 1987).

[22] D. J. Wales and J. P. K. Doye, J. Phys. Chem. A **101**, 5111 (1997).

[23] R. Fletcher and M. J. D. Powell, Comput. J. **6**, 163 (1963).

[24] J. Nocedal and S. J. Wright, *Numerical Optimisation*, Springer Series in Operation Research (Springer, New York, 1999).

[25] M. T. Chu, R. E. Funderlic, and G. H. Golub, SIAM J. Matrix Anal. Appl. **20**, 428 (1998).

[26] R. B. Schnabel, Math. Program. **15**, 247 (1978).

[27] W. C. Davidon, Math. Program. **9**, 1 (1975).

[28] C. R. Rao and S. K. Mitra, *Generalized Inverse of Matrices and its Applications* (Wiley, New York, 1971).

[29] M. B. Reed, Int. J. Comput. Math. **86**, 606 (2009).

[30] J. L. McCauley, Physica A **382**, 445 (2007).

[31] J. L. McCauley, MPRA Paper No. 2128; http://mpra.ub.uni-muenchen.de/2128/, (2007).

[32] P. Eastman and S. Doniach, Proteins **30**, 215 (1998).

[33] B. Mishra and T. Schlick, J. Chem. Phys. **105**, 299 (1996).

[34] L. N. Mazzoni and L. Casetti, Phys. Rev. Lett. **97**, 218104 (2006).

[35] A. Buckley and A. Lenir, Math. Program. **27**, 155 (1983).

[36] D. F. Shanno and K.-H. Phua, J. Optim. Theory Appl. **25**, 507 (1978).

[37] M. Hütter and H. C. Öttinger, J. Chem. Soc., Faraday Trans. **94**, 1403 (1998).