Conjugate gradient heat bath for ill-conditioned actions

Michele Ceriotti,* Giovanni Bussi, and Michele Parrinello
Computational Science, Department of Chemistry and Applied Biosciences, ETH Zurich, USI Campus, Via Giuseppe Buffi 13, CH-6900 Lugano, Switzerland
(Received 29 March 2007; published 29 August 2007)

We present a method for performing sampling from a Boltzmann distribution of an ill-conditioned quadratic action. This method is based on heat-bath thermalization along a set of conjugate directions, generated via a conjugate-gradient procedure. The resulting scheme outperforms local updates for matrices with very high condition number, since it avoids the slowing down of modes with lower eigenvalue, and has some advantages over the global heat-bath approach, compared to which it is more stable and allows for more freedom in devising case-specific optimizations.

DOI: 10.1103/PhysRevE.76.026707 PACS number(s): 02.70.Tt, 02.50.Ng

A common problem in many branches of statistical physics is the sampling of distributions of the type $p \propto \exp(-\frac{1}{2}\mathbf{x}\underline{\underline{\mathbf{A}}}\mathbf{x})$, where $\underline{\underline{\mathbf{A}}}$ is a positive definite $N \times N$ matrix and the random variable $\mathbf{x}$ an N-dimensional vector. Areas in which such sampling is needed are for instance QCD [1–3] and a recently developed linear scaling electronic structure method [4,5]. In principle sampling p is straightforward, if diagonalizing $\underline{\underline{\mathbf{A}}}$ is an option. However, in many cases, N is so large that circumventing the $O(N^3)$ diagonalization step becomes mandatory. Different approaches have been proposed. In the so-called global heat-bath method one writes $\underline{\underline{\mathbf{A}}}=\underline{\underline{\mathbf{M}}}^T\underline{\underline{\mathbf{M}}}$, and obtains a series of statistically independent vectors by solving the linear system $\underline{\underline{\mathbf{M}}}\mathbf{x}=\mathbf{R}$, where $\mathbf{R}$ is a vector whose components are distributed according to a Gaussian with zero mean and unit variance $\langle R^2\rangle=1$. The advantage of this method is that the algorithmic complexity of the problem can be reduced by using an iterative solver for the linear system. In order to expedite sampling a Metropolis-like criterion has been suggested that leads to correct sampling without having to bring the iterative process to full convergence [6,7]. Unfortunately, when the ratio between the largest and smallest eigenvalues is large (ill-conditioned matrices), the acceptance of this scheme drops to zero unless full convergency is achieved. An alternative approach is the local heat-bath algorithm, in which at every step one single component of the state vector $\mathbf{x}$ is thermalized in turn, keeping the others fixed. It has been pointed out elsewhere [8,9] that there is a close analogy between this second method and the Gauss-Seidel minimization technique. This approach is relatively inexpensive, but becomes very inefficient when the condition number of $\underline{\underline{\mathbf{A}}}$ is large, and even more inefficient when the observable of interest depends strongly on the eigenvectors corresponding to smaller eigenvalues.

In this paper we propose a heat-bath algorithm in which moves are performed along mutually conjugated directions. This choice is based on the analogy between various heat-bath methods (see, e.g., Ref. [8]) and directional minimization techniques. We show both analytically and numerically that the choice of conjugate directions allows all the degrees of freedom to become decorrelated on the same time scale, independent of their associated eigenvalue. We also discuss the cases in which the improved efficiency outbalances the additional computational cost. Our method can be interpreted as the subdivision of the global heat-bath matrix inversion process into N intermediate steps, all of which guarantee an exact sampling of the probability distribution.

In Sec. I we introduce a simple formalism to treat heat-bath moves along general directions, discuss the properties of a sweep through a set of conjugate directions, and describe a couple of algorithms to obtain such a set with reasonable effort. In Sec. II we present some numerical tests on a model action and compare the efficiency of conjugate directions heat bath with local moves for a model observable. In Sec. III we compare our method with global heat bath, and in Sec. IV we present our conclusions.

I. COLLECTIVE MODES HEAT BATH

Given a probability distribution

$$P(\mathbf{x}) \propto \exp\left[-\left(\frac{1}{2}\mathbf{x}\underline{\underline{\mathbf{A}}}\mathbf{x}-\mathbf{b}\cdot\mathbf{x}\right)\right], \tag{1}$$

a generic heat-bath algorithm can be described as a stochastic process in which the vector $\mathbf{x}(t+1)$ is related to the vector at the previous step $\mathbf{x}(t)$ by

$$\mathbf{x}(t+1)=\mathbf{x}(t)+\tau\mathbf{d}, \tag{2}$$

where $\mathbf{d}$ is a direction in the $\mathbf{x}$ space and

$$\tau=-\frac{\mathbf{d}(\underline{\underline{\mathbf{A}}}\mathbf{x}-\mathbf{b})}{\mathbf{d}\underline{\underline{\mathbf{A}}}\mathbf{d}}+(\mathbf{d}\underline{\underline{\mathbf{A}}}\mathbf{d})^{-1/2}R, \tag{3}$$

where R is a Gaussian random number with zero mean and unitary spread $\langle R^2\rangle=1$. The application of this algorithm does not require inversion of the matrix $\underline{\underline{\mathbf{A}}}$. The sequence of directions $\mathbf{d}$ is rather arbitrary, and could be a random sequence or a predefined deterministic sequence. Strictly speaking, detailed balance is satisfied only if the directions are randomly chosen at each step. Nevertheless it has been shown in Ref. [10] that correct sampling can be achieved if every Monte Carlo move leaves the equilibrium distribution

*michele.ceriotti@phys.chem.ethz.ch

unchanged. In Appendix A we show that this is the case, provided that direction $\mathbf{d}$ is chosen independently from position $\mathbf{x}$. Nevertheless, different choices of directions can lead to different sampling efficiency. Our final choice will be to select for $\mathbf{d}$ a sequence of conjugate directions (Sec. I A). However, we shall first analyze the choice of random, uncorrelated directions, and a sequential sweep along a set of orthogonal directions.

For the sake of simplicity, we take $\mathbf{b}=0$ and we choose the basis into which $\underline{\underline{\mathbf{A}}}$ is diagonal, $A_{ij}=a_i\delta_{ij}$. Since these properties are subsequently never used, no loss of generality is implied. To compare the efficiency of the different choices of directions we shall consider the autocorrelation matrix for the components along the eigenmodes $\langle x_i(0)x_j(t)\rangle$. A quantitative measure of the speed of decorrelation of $\langle x_i(0)x_j(t)\rangle$ can be obtained from its slope at the origin. Since in Monte Carlo one progresses in discrete steps, this quantity is given by

$$\langle x_i(0)x_j(1)\rangle = \sqrt{\langle x_i(0)^2\rangle\langle x_j(0)^2\rangle}[\delta_{ij} - \Delta_{ij}(\mathbf{d})]. \tag{4}$$

In Eq. (4) we have introduced the normalized slope tensor $\underline{\underline{\mathbf{\Delta}}}$, which can be expressed as a function of the eigenvalues of $\underline{\underline{\mathbf{A}}}$ and of the components of $\mathbf{d}$, using Eqs. (2) and (3):

$$\Delta_{ij}(\mathbf{d}) = \frac{a_i h_i h_j}{\sum_k a_k h_k^2} \frac{\langle x_i(0)^2\rangle}{\sqrt{\langle x_i(0)^2\rangle\langle x_j(0)^2\rangle}} = \frac{\sqrt{a_i a_j} d_i d_j}{\sum_k a_k d_k^2}. \tag{5}$$

Therefore, depending on the choice of direction $\mathbf{d}$, the different components of the vector $\mathbf{x}$ decorrelate at different speeds. However, since $\mathrm{Tr}\,\underline{\underline{\mathbf{\Delta}}}=1$, the sum of these normalized speeds does not depend on the direction chosen. The same quantity $\underline{\underline{\mathbf{\Delta}}}$ also enters a recursion relation for the autocorrelation functions at a generic Monte Carlo step t,

$$\langle x_i(0)x_j(t+1)\rangle = \langle x_i(0)x_j(t)\rangle - \sum_k \left[\langle x_i(0)x_k(t)\rangle \sqrt{\frac{a_k}{a_j}}\Delta_{kj}(\mathbf{d})\right]. \tag{6}$$

Use of this equation requires that one appropriately averages over the direction $\mathbf{d}$, as we shall discuss in the following.

We will begin our analysis from the simpler case, in which the direction $\mathbf{d}$ is chosen at every step to be equal to a stochastic vector $\mathbf{R}$, whose components are distributed as Gaussian random numbers with zero mean and standard deviation one. The normalized slope tensor (5) in this case results from an average over the possible directions,

$$\langle \Delta_{ij}(\mathbf{d}=\mathbf{R})\rangle = a_i\delta_{ij}\left\langle \frac{R_i^2}{\sum_k a_k R_k^2}\right\rangle \overset{N\to\infty}{\approx} \frac{a_i\delta_{ij}}{\mathrm{Tr}\,\underline{\underline{\mathbf{A}}}}. \tag{7}$$

The limit expression holds for the size N of the matrix going to infinity (see Appendix B), under the hypothesis that the largest eigenvalue of $\underline{\underline{\mathbf{A}}}$ does not grow with N and that $\mathrm{Tr}\,\underline{\underline{\mathbf{A}}}$ is $O(N)$, hypotheses which are relevant to many physical problems. Since in this case the direction chosen at every step is independent of all the previous choices, the same average enters Eq. (6) at any time, so that proceeding by induction one can easily obtain the entire autocorrelation function,

$$\langle x_i(0)x_j(t)\rangle = \delta_{ij}\langle x_i(0)^2\rangle[1-\langle\Delta_{ij}\rangle]^t, \tag{8}$$

where $\langle\Delta_{ij}\rangle$ is the quantity obtained in Eq. (7). From Eq. (8) we can calculate the autocorrelation time for mode i,

$$\tau_i = \frac{\sum_{t=0}^{\infty}\langle x_i(0)x_i(t)\rangle}{\langle x_i^2\rangle} = \left[a_i\left\langle\frac{R_i^2}{\sum_k a_k R_k^2}\right\rangle\right]^{-1} \overset{N\to\infty}{\approx} \frac{\mathrm{Tr}\,\underline{\underline{\mathbf{A}}}}{a_i}.$$

In the case of large N, the decorrelation speed of the components along normal modes is directly proportional to the corresponding eigenvalue, so that in ill-conditioned cases a critical slowing down for the softer normal modes will be present.

Let us now consider moves along a predefined set of orthogonal directions $\{\mathbf{u}^{(m)}\}_{m=0\ldots N-1}$. This is done to mimic the case in which one performs a sweep along Cartesian directions. In our reference frame, where $\underline{\underline{\mathbf{A}}}$ is taken to be diagonal, this would be trivial, hence the choice of an arbitrarily oriented set of orthogonal directions. As in standard local heat bath, the outcome will depend on the orientation of the $\{\mathbf{u}^{(m)}\}$ relative to the eigenvectors of $\underline{\underline{\mathbf{A}}}$. Averaging over all the possible choices of initial direction, we find the slope at $t=0$,

$$\langle\Delta_{ij}\rangle = \frac{1}{N}\sum_m \Delta_{ij}(\mathbf{u}^{(m)}) = \frac{1}{N}\sum_m \frac{\sqrt{a_i a_j}u_i^{(m)}u_j^{(m)}}{\sum_k a_k u_k^{(m)2}}. \tag{9}$$

Obviously, it is not possible to reduce this result to an expression which does not depend on the particular set of orthogonal directions. However, the following inequality holds

$$\frac{a_i\delta_{ij}}{N a_{\max}} \le \langle\Delta_{ij}\rangle \le \frac{a_i\delta_{ij}}{N a_{\min}}. \tag{10}$$

Equation (10) does not put rigid constraints on the value of $\langle\Delta_{ij}\rangle$, but demonstrates that also in this case $\underline{\underline{\mathbf{\Delta}}}$ is diagonal and suggests that in real life the convergence will be faster for the higher eigenvalues, and that the spread in the relaxation speed for different modes is larger when the condition number $\kappa=a_{max}/a_{\min}$ is higher.

In the case where directions $\{\mathbf{u}^{(m)}\}$ are swept sequentially we have not been able to derive a closed expression for $\langle x_i(0)x_j(t)\rangle$ because of the dependence of $\mathbf{d}(t)$ on the previous history. If, on the other hand, a random direction is drawn from $\{\mathbf{u}^{(m)}\}$ at every step, $\langle x_i(0)x_j(t)\rangle$ is given by expression (8) where $\langle\Delta_{ij}\rangle$ has the value in Eq. (9).

A. Moves along conjugate directions

It is clear from Eq. (8) that a random choice of the directions $\mathbf{d}$ leads to fast decorrelation of the components relative to the eigenvectors with high eigenvalues. On the other hand, the components relative to the eigenvectors with low eigenvalues will decorrelate more slowly. Similar behavior is ex-

pected for the local heat-bath method, unless particular relations hold between the eigenvectors and the Cartesian axes. If the operator $\underline{\underline{\mathbf{A}}}$ is ill conditioned, the practical consequence is that the slow modes will be accurately sampled only after a very large number of steps. As we have already discussed, the sum of the decorrelation slopes of the different components does not depend on the choice of the directions $\mathbf{d}$. However, with a proper choice of the directions $\mathbf{d}$ this sum could be spread in a uniform way among the different modes. A similar problem arises in minimization algorithms based on directional search, and is often solved choosing a sequence of conjugated directions [11]. In the same spirit, we can compute the decorrelation speed of the different modes when the $\mathbf{d}$'s are chosen to be conjugated directions. Let us consider a set of conjugated directions $\{\mathbf{h}^{(i)}\}$, such that $\mathbf{h}^{(i)}\underline{\underline{\mathbf{A}}}\mathbf{h}^{(j)}=\delta_{ij}$. The set $\{\mathbf{h}^{(i)}\}$ can be generated with various algorithms, such as a Gram-Schmidt orthogonalization that uses the positive definite $\underline{\underline{\mathbf{A}}}$ matrix as a metric, or a conjugate gradient procedure, as described in Sec. I B.

Using the fact that $\Sigma_k h_i^{(k)} h_j^{(k)} = a_i^{-1}\delta_{ij}$, the slope at $t=0$ is

$$\langle \Delta_{ij} \rangle = \frac{1}{N}\sum_m \frac{\sqrt{a_i a_j} h_i^{(m)} h_j^{(m)}}{\mathbf{h}^{(m)}\underline{\underline{\mathbf{A}}}\mathbf{h}^{(m)}} = \frac{1}{N}\sqrt{\frac{a_j}{a_i}}\sum_m \frac{a_i h_i^{(m)} h_j^{(m)}}{\mathbf{h}^{(m)}\underline{\underline{\mathbf{A}}}\mathbf{h}^{(m)}} = \frac{\delta_{ij}}{N}.$$

With this choice, the decorrelation slopes of the different modes are independent of the eigenvalue. If one chooses one conjugate direction at random at each step it is straightforward to show that overall the autocorrelation function decays exponentially as

$$\langle x_i(0)x_j(t)\rangle = \delta_{ij}\langle x_i(0)^2\rangle\left[1-\frac{1}{N}\right]^t.$$

This derivation shows that if matrix $\underline{\underline{\mathbf{A}}}$ is ill conditioned and one wishes to decorrelate the slow modes, then the choice of performing the heat bath using a sequence of conjugated directions can improve the sampling quality dramatically. Of course, the slow modes are accelerated and the fast modes are decelerated. However, it is clear that a completely independent vector $\mathbf{x}$ is obtained only when all the modes are decorrelated. A heat bath on conjugate directions allows all the modes to be decorrelated with the same efficiency, irrespective of their stiffness. Even better efficiency can be obtained by sequentially sweeping a set of conjugated directions. At first sight it would appear that the dependence of $\mathbf{h}(t)$ on $\mathbf{h}(t-1)$ would make it very difficult if not impossible to obtain the autocorrelation function in a closed form. However, conjugate directions have a redeeming feature. If we expand the position vector on the nonorthogonal basis $\{\mathbf{h}^{(m)}\}$, $\mathbf{x}=\Sigma_i \alpha^i \mathbf{h}^{(i)}$, and we evaluate the correlation matrix between the contravariant components α^i, we find that $\langle \alpha^i \alpha^j\rangle = \delta_{ij}$. This property can be easily demonstrated taking into account that the ensemble average $\langle x_i x_j\rangle = A_{ij}^{-1}$, and that conjugacy implies $\mathbf{h}^{(i)}\underline{\underline{\mathbf{A}}}\mathbf{h}^{(j)}=\delta_{ij}$. Thus, effectively, every time we perform a heat-bath move along direction $\mathbf{h}^{(i)}$ the component α^i is randomized, without affecting the others. After a complete sweep across the set of directions a completely independent state is obtained.

A more formal proof is provided in Appendix C, where it is also demonstrated that the autocorrelation function is

$$\langle x_i(0)x_i(t)\rangle = \langle x_i(0)^2\rangle \times \begin{cases} \left[1-\frac{t}{N}\right], & t<N, \\ 0, & t\geq N. \end{cases} \tag{11}$$

Therefore the corresponding autocorrelation time is $\tau_i=(N+1)/2$. A remarkable feature of Eq. (11) is that the autocorrelation function is linear, and that after N moves a completely independent vector is obtained. This property holds also for the global heat-bath method. In Sec. III we shall discuss the relation between our approach and global heat-bath sampling.

B. Conjugate-gradient approach to generate conjugate directions

In the last section we have shown how a heat-bath algorithm based on conjugate directions can dramatically improve the sampling of the slow modes for an ill-conditioned action. An efficient strategy to generate these directions is the application of the conjugate gradient procedure [11]. For the sake of completeness and to introduce a consistent notation we give here an outline of the conjugate-gradient (CG) algorithm. One starts from a random configuration and search direction, $\mathbf{h}^{(0)}=\mathbf{g}^{(0)}=\mathbf{R}$, so that the directions obtained and the sample vector $\mathbf{x}$ are independent as required. Then, a series of directions $\mathbf{h}^{(m)}$ and residuals $\mathbf{g}^{(m)}$ are generated using the recurrence relations

$$\mathbf{g}^{(i+1)} = \mathbf{g}^{(i)} - \lambda_i \underline{\underline{\mathbf{A}}}\cdot\mathbf{h}^{(i)}, \quad \mathbf{h}^{(i+1)} = \mathbf{g}^{(i+1)} + \gamma_i \cdot \mathbf{h}^{(i)},$$

$$\lambda_i = \frac{\mathbf{g}^{(i)}\cdot\mathbf{g}^{(i)}}{\mathbf{h}^{(i)}\underline{\underline{\mathbf{A}}}\mathbf{h}^{(i)}}, \quad \gamma_i = \frac{\mathbf{g}^{(i+1)}\cdot\mathbf{g}^{(i+1)}}{\mathbf{g}^{(i)}\cdot\mathbf{g}^{(i)}}.$$

This procedure generates at every step a new direction $\mathbf{h}^{(i)}$, conjugated to all the previous ones, and it can be used to perform a directional heat-bath move on $\mathbf{x}$. It should be stressed that there is no need to store all the $\mathbf{h}^{(i)}$ if the heat-bath moves are performed concurrently with the CG minimization. The "force" $\underline{\underline{\mathbf{A}}}\mathbf{h}^{(i)}$ can be reused for performing the heat-bath update [cf. Eq. (2)]. At a certain point the CG procedure will be over, with the residual $\mathbf{g}$ dropping to zero. The sequential sweep algorithm described into the previous section can be implemented starting again from the same $\mathbf{g}^{(0)}$.

In contrast to the global heat-bath method, numerical stability is not a major issue, since the accuracy of the sampling does not depend on the search directions being exactly conjugated. The only effect of imperfect conjugation would be to slightly reduce the decorrelation efficiency. There is, however, a drawback to this approach. In order to be ergodic, the set of directions must span the whole space. The problem arises when there are degenerate eigenvalues, as CG converges to zero in a number p of iterations equal to the number of distinct eigenvalues. If we keep reusing the same set

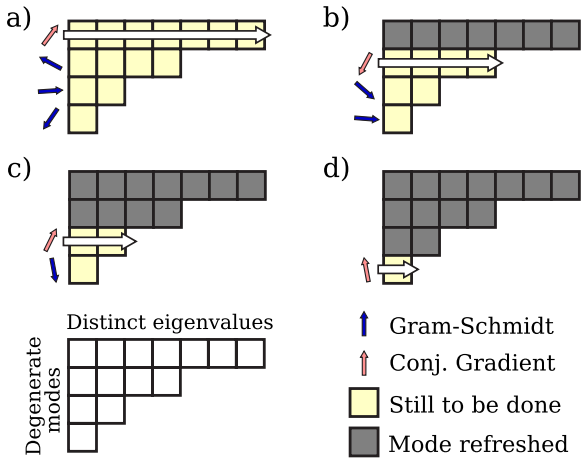

FIG. 1. (Color online) Scheme of the block algorithm described in Sec. I B; squares represent eigenvectors of the action matrix, which need to be refreshed in order to obtain a statistically independent sample point; modes on the same column correspond to the same, degenerate eigenvalue. At every step, one of the vectors of a set with the same size as the biggest degenerate subspace is used in a conjugate gradient minimization, while the remaining ones are made orthogonal to the search directions that are generated in the process. When the first vector approaches zero, one can start back on the second one [(b), and the process can be continued (c) and (d)] until the refresh is complete.

of $p<N$ directions, only a part of the subspaces corresponding to degenerate eigenvalues will be explored, and the sampling will not be ergodic.

We have considered two possible ways of recovering ergodicity. The simplest consists in drawing a different random point $\mathbf{g}^{(0)}=\mathbf{R}$ every time we reset the CG search. This causes a deviation from the linear behavior of the autocorrelation functions for $t\approx N$. Nondegenerate eigenvalues will initially converge with $-1/p$ instead of $-1/N$ slope, but degenerate ones will converge more slowly, and with exponential trend, as we are sampling random directions within every degenerate subspace.

In order to improve the efficiency, we mix CG with Gram-Schmidt orthogonalization of a small set of vectors, ideally of the same size d of the largest degeneracy present. As discussed earlier, here Gram-Schmidt orthogonalization has to be performed using the metric of $\underline{\underline{\mathbf{A}}}$, which amounts to imposing conjugacy. The procedure is illustrated in Fig. 1. We start from d random vectors, $\{\mathbf{v}^{(j)}\}_{j=0..d-1}$. We set $\mathbf{h}^{(0)}=\mathbf{g}^{(0)}=\mathbf{v}^{(0)}$ and begin a CG minimization. At each step we obtain a search direction $\mathbf{h}^{(i)}$, and make each of the other $d-1$ vectors conjugate to $\mathbf{h}^{(i)}$ with a Gram-Schmidt procedure. This does not require any matrix-vector product other than the one necessary for the heat-bath step. After p iterations the conjugate gradient will have converged and $\mathbf{g}$ will be close to zero. We can start again from the second vector in the pool, which meanwhile has become $\bar{\mathbf{v}}^{(1)}$, and is conjugate to all the directions visited so far. Thus, we set $\mathbf{h}^{(0)}=\mathbf{g}^{(0)}=\bar{\mathbf{v}}^{(1)}$ and start again the CG procedure, orthogonalizing the $d-2$ remaining vectors to $\mathbf{h}^{(i)}$, and so on and so forth. After N steps the procedure will be converged. At the successive sweep, one can generate again a set of random initial $\{\mathbf{v}^{(j)}\}$. This can make the method more stable, at the cost of some loss in performance. Some savings can be made if one stores the conjugated $\bar{\mathbf{v}}^{(i)}$, and uses them in the subsequent sweeps, avoiding the need to repeat the GS orthogonalizations (see Fig. 1). In practice, where more than one complete sweep is affordable, it is easy to devise adaptive variations of this scheme, in which the pool of vectors $\{\mathbf{v}^{(j)}\}$ is enlarged whenever the CG minimization converges in less than N steps, so that in a few sweeps the optimal size to guarantee ergodicity is attained.

II. BENCHMARKS AND COMPARISON WITH LOCAL HEAT BATH

In the previous section we have discussed a collective modes heat-bath method that could outperform standard local heat-bath techniques when the Hamiltonian has a very large condition number and sampling along the slower eigenmodes is required. In this section we illustrate the efficiency of our algorithm using numerical experiments on a simple model for $\underline{\underline{\mathbf{A}}}$,

$$\underline{\underline{\mathbf{A}}}=\underline{\underline{\mathbf{1}}}+\begin{pmatrix} -2b & b & 0 & \cdots & 0 & b \\ b & -2b & b & 0 & \cdots & 0 \\ 0 & b & -2b & b & \ddots & \vdots \\ \vdots & 0 & b & -2b & \ddots & 0 \\ 0 & \vdots & \ddots & \ddots & \ddots & b \\ b & 0 & \cdots & 0 & b & -2b \end{pmatrix}. \quad (12)$$

This matrix corresponds to the dynamical matrix of a linear chain of spring-connected masses, with periodic boundary conditions and an additional diagonal term to make the acoustic mode nonzero. b can be chosen so as to obtain the desired condition number. Eigenmodes and eigenvalues for such a matrix are easily obtained,

$$a_k=1+2b\left(1-\cos\frac{2k\pi}{N}\right),$$

$$u_l^{(k)}=\sqrt{\frac{1+\delta_{0k}+\delta_{N/2,k}}{N}}\times\begin{cases}\cos\dfrac{2kl\pi}{N}, & k\le N/2,\\ \sin\dfrac{2kl\pi}{N}, & k>N/2,\end{cases}$$

and projection of a state on the eigenvectors is quickly done via fast-Fourier transform. In Fig. 2 we compare the autocorrelation functions obtained with different algorithms for a matrix of the form (12). Figure 2 also highlights the ergodicity problems connected with the naive use of the conjugate gradient algorithm to generate the search directions, and shows how both the suggestions of Sec. I B can help in solving this problem. In general, a conjugate directions search speeds up decorrelation for the slower modes, but is less efficient than local heat bath for the modes with a high eigenvalue. This is a direct consequence of the fact that $\mathrm{Tr}\,\underline{\underline{\mathbf{A}}}=1$. An additional advantage of our method is the linear rate of decorrelation, which allows complete decorrelation just like the direct inversion of $\underline{\underline{\mathbf{M}}}$, whereas moves along the Car-

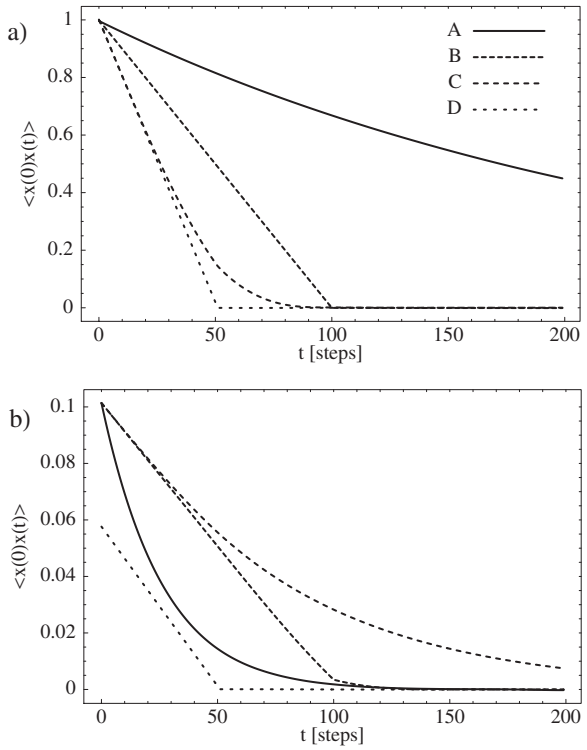

FIG. 2. Autocorrelation functions for (a) the projection along the mode $a_0=1$; (b) the projection along the mode $a_4\approx 9.8$ for a matrix of the form (12) with $N=100$ and condition number $\kappa=10^3$. Line A corresponds to local heat-bath moves (one step stands for a complete sweep of the N coordinates), lines B to D to conjugate direction moves: B is the hybrid conjugate gradient–Gram-Schmidt block algorithm; C corresponds to CG sweeps, with the search direction randomized at the beginning of every sweep; curve D corresponds to CG sweeps starting from the same initial vector. Conjugate direction moves decorrelate faster than local heat bath for the slow mode, but are less efficient for modes with higher eigenvalue. For degenerate eigenmodes, the method used for curve D is not ergodic (and thus gives incorrect values for $\langle x_i^2\rangle$), and random restarts (curve C) are much less efficient than the hybrid (curve B) algorithm.

tesian axes lead to approximatively exponential autocorrelation functions.

We stress again that the relative efficiency of the two methods depends strongly on the observable being calculated and on the actual spectrum of the Hamiltonian of the system. As a more realistic benchmark we will consider the evaluation of the trace of the inverse matrix, i.e.,

$$\Omega = \mathrm{Tr}(\underline{\underline{\mathbf{A}}}^{-1}) = \langle \mathbf{x}^2 \rangle. \tag{13}$$

This observable is strongly dependent on the slow modes.

In Fig. 3 we plot the ratios of the autocorrelation times $\tau[\Omega]$ as obtained with local heat-bath moves and with the block conjugate gradient version of our algorithm, as a function of changing condition number and system size.

III. COMPARISON WITH GLOBAL HEAT BATH

It remains for us to discuss how our method fares in comparison with global heat bath. The latter requires that matrix

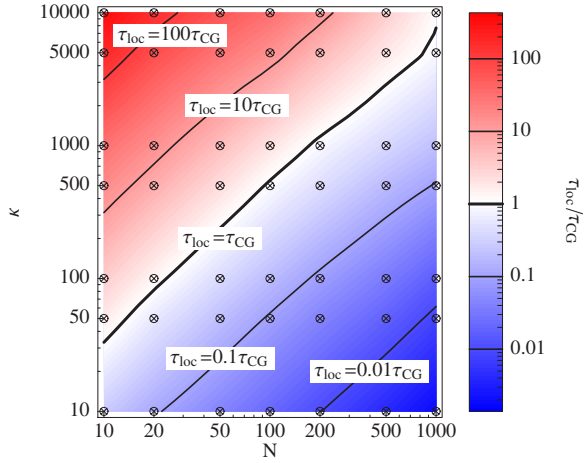

FIG. 3. (Color online) Comparison of the efficiency of local heat bath versus conjugate-gradient moves. The graph represents τ_{CG}/τ_{loc}, the ratio of the autocorrelation times for the observable Ω (13); τ_{loc} corresponds to the value obtained from standard local heat-bath moves (one unit of Monte Carlo time corresponds to a whole coordinates sweep), while τ_{CG} corresponds to the value obtained with moves along conjugate directions, as obtained from our block algorithm with random restarts. The data plotted results from a linear interpolation of some simulations (labeled by ⊗) performed for an action of the form (12), with varying size N and condition number κ.

$\underline{\underline{\mathbf{A}}}$ be decomposable in the form $\underline{\underline{\mathbf{A}}}=\underline{\underline{\mathbf{M}}}^T\underline{\underline{\mathbf{M}}}$. This is the case in many fields [4], but in principle if it were necessary to decompose $\underline{\underline{\mathbf{A}}}$ this would add extra cost. Here we make our comparison assuming that $\underline{\underline{\mathbf{M}}}$ is already available. In such a case, the two algorithms are on paper equally efficient in producing statistically independent samples. The global heat bath might offer some numerical advantages when the spectrum of $\underline{\underline{\mathbf{M}}}$ is highly degenerate, since the number of CG iterations needed to solve the $\underline{\underline{\mathbf{M}}}x=\mathbf{R}$ linear system is $p<N$, as discussed earlier. Whenever a good preconditioner for the linear system is available, other inversion algorithms such as the stabilized biconjugate gradient [12] or the generalized conjugate residual may allow to solve the linear system with a sufficient accuracy more efficiently than using CG. In this paper we make the comparison with conjugate gradient because of the close analogy with our scheme and because our method is aimed at problems where ill conditioning cannot be otherwise relieved.

In this respect, our method displays significant advantages. First, it is more stable, because every move preserves the probability distribution, and the conjugate gradient procedure (which is known to be quite delicate in problems with large condition number) is only used to generate search directions. Instabilities in the procedure, which would cause incorrect sampling in the global heat bath, affect only the efficiency, and not the accuracy. Moreover, dividing the N steps of an iterative inversion process into separate heat-bath moves greatly improves the flexibility of the sampling scheme. To give some examples, if one needs to perform an average on a slowly varying $\underline{\underline{\mathbf{A}}}$, it is possible to perform only a partial sweep with fixed action, then continue with the new $\underline{\underline{\mathbf{A}}}$, assuming that eigenmodes will change slowly. It is also

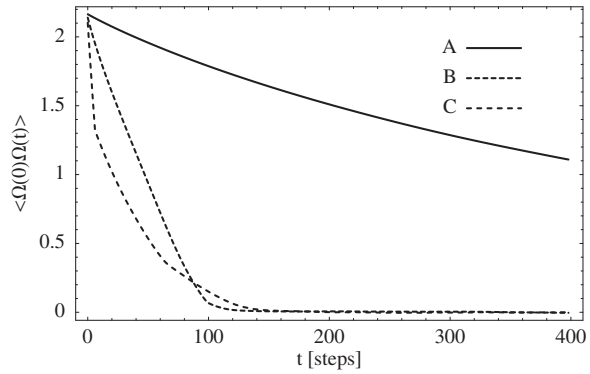

FIG. 4. Autocorrelation function for the observable (13) for an action of the form (12), with size $N=100$ and condition number $\kappa=5\times10^3$. Line *A* corresponds to local heat bath, line *B* to the "hybrid" versions of our CG algorithm, with a pool of two vectors with random restarts, while curve *C* is obtained including the tricks described in Sec. III with $m=5$.

straightforward to tailor the choice of directions in order to optimize the convergence speed for the observable or interest. Adler's overrelaxation [13] can be included naturally, and can help in further optimizing the autocorrelation time. As an example of possible fine tunings, let us recall the observable Ω introduced in the previous section [Eq. (13)]. This observable depends strongly on the softer eigenvector of $\underline{\underline{\mathbf{A}}}$. We have then modified our algorithm in the following way: we perform block conjugate gradient sweeps, with random resets, and we monitor the curvature along the direction being thermalized, $\mathbf{h}\underline{\underline{\mathbf{A}}}\mathbf{h}/\mathbf{h}\cdot\mathbf{h}$. We save the direction of minimum curvature encountered along the sweep, $\mathbf{h}_{\min}$; during the following sweep, every m moves along the CG directions, one move is performed along $\mathbf{h}_{\min}$. As is evident from Fig. 4, this trick considerably reduces the autocorrelation time for Ω. Even smarter combinations of moves can be devised, and the one we suggest is just an example of how the additional flexibility gained through subdividing the inversion process in N exact sampling moves can be exploited. In Table I we report some numerical estimates of the error in the evaluation or Ω, which can serve as a reference to compare our method to other approaches.

IV. CONCLUSIONS

We have presented an algorithm for performing collective modes heat bath along conjugate directions for a quadratic action, which allows the components of the sampling vector along all modes to be decorrelated in N steps, with a linear decay to zero. This method is more computationally demanding than local updates, but becomes competitive for ill-conditioned actions, when one needs to compute observables which depend on modes with low eigenvalues, or when the spectrum of the action matrix has only a few high eigenvalue modes which would slow down Cartesian moves. In fact, this method has an efficiency comparable with that of direct inversion of the matrix, but presents various advantages, such as improved stability, as the numerical issues connected with conjugate gradient method do not affect the accuracy of the sampling, and the possibility of exploiting some additional flexibility to improve the sampling on a case-by-case basis. Lastly, global heat bath requires the knowledge of the square root of the action $\underline{\underline{\mathbf{A}}}$, so our scheme should be considered whenever the square root is difficult to compute or its use is inefficient with respect to the original action.

The geometrical simplicity of this approach, with its close analogy with minimization methods, also suggests that it might be extended to the sampling of anharmonic systems.

TABLE I. Percentual errors in the evaluation of $\Omega=\langle\mathbf{x}^2\rangle$ [Eq. (13)], estimated using a blocking analysis, for different sampling methods. *A* corresponds to local heat bath, *B* corresponds to "hybrid" versions of our CG algorithm, with a pool of two vectors with random restarts, while curve *C* is obtained including the tricks described in Sec. III with $m=50$. Different tests are performed with varying matrix size N, number of sampling steps T, and condition number κ. Due to the large autocorrelation time, the values of the error for local heat bath with $N=100$ and $T=10^6$ could not be estimated as reliably as in the other cases, and are only indicative.

N	κ	T	A	B	C
10^3	5×10^4	10^6	4.0	1.5	1.4
10^3	5×10^4	10^7	1.3	0.51	0.45
10^3	5×10^3	10^6	0.78	0.85	0.85
10^3	5×10^3	10^7	0.24	0.28	0.28
100	5×10^4	10^6	~11	1.2	1.1
100	5×10^4	10^7	4.9	0.44	0.34
100	5×10^3	10^6	~3	0.88	0.82
100	5×10^3	10^7	1.1	0.30	0.25

ACKNOWLEDGMENTS

It is a pleasure to acknowledge useful discussions with Fulvio Ricci and Nazario Tantalo, whose suggestions have helped in improving the manuscript.

APPENDIX A

We report here a simple demonstration of the fact that heat-bath moves along a generic direction $\mathbf{d}$ leave an equilibrium probability distribution unchanged. We will use the fact that if $\mathbf{R}$, $\mathbf{R}'$, and $\mathbf{R}''$ are vectors distributed as Gaussians with zero mean and standard deviation one, then $\underline{\underline{\mathbf{B}}}\mathbf{R}+\underline{\underline{\mathbf{C}}}\mathbf{R}'$ is distributed as $\underline{\underline{\mathbf{D}}}\mathbf{R}''$, where $\underline{\underline{\mathbf{D}}}^T\underline{\underline{\mathbf{D}}}=\underline{\underline{\mathbf{B}}}^T\underline{\underline{\mathbf{B}}}+\underline{\underline{\mathbf{C}}}^T\underline{\underline{\mathbf{C}}}$. Since $\mathbf{x}$ is drawn from the equilibrium distribution, i.e., $\mathbf{x}=\underline{\underline{\mathbf{M}}}^{-1}\mathbf{R}$, we can cast Eqs. (2) and (3) into the form

$$x'_j=\sum_m P_{jm}R_m+\sum_m Q_{jm}R'_m,$$

$$P_{jm}=(\underline{\underline{\mathbf{M}}}^{-1})_{jm}-d_j\sum_k M_{km}d_k,\quad Q_{jm}=d_j\delta_{m0},$$

where we have put $\mathbf{b}=0$ into Eq. (3) and normalized the direction so that $\mathbf{d}\underline{\underline{\mathbf{A}}}\mathbf{d}=1$ in order to simplify the notation. We can then compute

$$\sum_m P_{jm}P_{lm} = \underline{\underline{\mathbf{A}}}^{-1}_{jl} - d_j d_l, \quad \sum_m Q_{jm}Q_{lm} = d_j d_l,$$

so that $\underline{\underline{\mathbf{P}}}^T\underline{\underline{\mathbf{P}}} + \underline{\underline{\mathbf{Q}}}^T\underline{\underline{\mathbf{Q}}} = (\underline{\underline{\mathbf{M}}}^{-1})^T\underline{\underline{\mathbf{M}}}^{-1}$, i.e., also $\mathbf{x}'$ may be written as $\underline{\underline{\mathbf{M}}}^{-1}\mathbf{R}$, and is therefore correctly distributed.

APPENDIX B

We shall discuss here briefly the derivation of the asymptotic form of Eq. (7) when the size N of the action matrix tends to infinity. The quantity to be computed is

$$Q_i = \left\langle \frac{R_i^2}{\sum_k a_k R_k^2} \right\rangle \propto \int d\mathbf{x} \frac{x_i^2}{\sum_k x_k^2 a_k} \exp\left[-\frac{1}{2}\sum_k x_k^2 \right].$$

The integral can be transformed as follows:

$$Q_i \propto \int_0^\infty dt \int d\mathbf{x} x_i^2 \exp\left[-\frac{1}{2}\sum_k (1 + a_k t) x_k^2 \right]$$
$$= \int_0^\infty dt \frac{1}{a_i t + 1} \prod_k \frac{1}{\sqrt{a_k t + 1}},$$

and the resulting expression, including the correct normalization, is

$$Q_i = \frac{1}{2}\int_0^\infty dt \frac{1}{a_i t + 1} f(t), \quad f(t) = \prod_k \frac{1}{\sqrt{a_k t + 1}}. \tag{B1}$$

Let us focus on $F = \int_0^\infty f(t)dt$, since all the Q_i's can be computed as $Q_i = a_i \frac{\partial F}{\partial a_i} + \frac{1}{2}F$. We perform the change of variables $Nt \to t$, so that

$$\int_0^\infty f(t)dt = \frac{1}{N}\int_0^\infty \tilde{f}(t)dt, \quad \tilde{f}(t) = \prod_k \frac{1}{\sqrt{\frac{a_k}{N}t + 1}}.$$

Under the physically reasonable assumption that $\mathrm{Tr}\,\underline{\underline{\mathbf{A}}} = O(N)$, and that the maximum eigenvalue does not scale with the system size, we can use $1/N$ as a small parameter. Expanding $\ln\tilde{f}$ one finds

$$\ln\tilde{f}(t) = \sum_k \ln\left(1 + \frac{a_k}{N}t\right) = \sum_{n=1} \frac{t^n}{n+1} \sum_k \left[\frac{a_k}{N}\right]^n$$
$$= \sum_k \frac{a_k}{N}\frac{t}{2} + \sum_{n=1} t^{n+1} O\left(\frac{1}{N^n}\right).$$

All but the leading term become negligible for $N \to \infty$. This suggests separating out from $\tilde{f}(t)$ the term order zero in $1/N$, and writing for F the expression

$$\frac{1}{N}\int_0^\infty \exp\left(-\frac{t}{2}\frac{\mathrm{Tr}\,\underline{\underline{\mathbf{A}}}}{N}\right)\left[1 + \frac{1}{4}\sum_k \left(\frac{a_k}{N}\right)^2 t^2 + O\left(\frac{1}{N^2}\right)t^3 + \cdots\right]dt, \tag{B2}$$

which leads to the asymptotic result $F = \frac{2}{\mathrm{Tr}\,\underline{\underline{\mathbf{A}}}} + O(N^{-2})$. Correspondingly, dropping the higher order terms in $1/N$, we have $Q_i = \frac{1}{\mathrm{Tr}\,\underline{\underline{\mathbf{A}}}} + O(N^{-2})$, which is the desired result.

APPENDIX C

We obtain here the autocorrelation function for the components along the eigenmodes of the action matrix $\underline{\underline{\mathbf{A}}}$, when performing heat-bath sweeps along a set of conjugate directions $\{\mathbf{h}^{(m)}\}_{m=0\ldots N-1}$. In this section, the indices of the directions are defined modulo N, i.e., $\mathbf{h}^{(j+N)} = \mathbf{h}^{(j)}$. In this case, one can write Eq. (6) as

$$\langle x_i(0)x_i(t+1)\rangle = \langle x_i(0)x_i(t)\rangle - \frac{1}{N}\sum_m \sum_k \left(\langle x_i(0)x_k(t)\rangle \sqrt{\frac{a_k}{a_i}} \Delta_{ki}(\mathbf{h}^{(m)}) \right). \tag{C1}$$

Explicit calculations for small values of t suggest for $t < N$ the ansatz

$$\langle x_i(0)x_i(t)\rangle = \langle x_i(0)^2\rangle \left(1 - \frac{t}{N}\right). \tag{C2}$$

Since the first term in Eq. (C1) does not contain the new direction, we can substitute the ansatz without concern. On the other hand, the second term contains reference to $\mathbf{h}^{(m)}$, so that the average that led to Eq. (C2) cannot be performed separately, and one should rather write

$$\frac{1}{N}\sum_m \sum_k \left[\sum_{k'} \langle x_i(0)x_{k'}(t-1)\rangle \times \left(\delta_{k'k} - \sqrt{\frac{a_{k'}}{a_k}}\Delta_{k'k}(\mathbf{h}^{(m-1)})\right)\sqrt{\frac{a_k}{a_i}}\Delta_{ki}(\mathbf{h}^{(m)}) \right], \tag{C3}$$

which is split into

$$\frac{1}{N}\sum_m \sum_k \left(\langle x_i(0)x_k(t-1)\rangle \sqrt{\frac{a_k}{a_i}}\Delta_{ki}(\mathbf{h}^{(m)}) \right), \tag{C4}$$

$$\frac{1}{N}\sum_{mkk'} \left(\langle x_i(0)x_{k'}(t-1)\rangle \sqrt{\frac{a_{k'}}{a_i}}\Delta_{k'k}(\mathbf{h}^{(m-1)})\Delta_{ki}(\mathbf{h}^{(m)}) \right). \tag{C5}$$

The term (C5) goes to zero, since

$$\sum_k \sum_m \Delta_{ik}(\mathbf{h}^{(m-n)})\Delta_{kj}(\mathbf{h}^{(m)}) = \delta_{n,pN}\delta_{ij},$$

while Eq. (C4) can be expanded again, giving rise to the $t-2$ analog and to a term containing $\Delta_{k'k}(\mathbf{h}^{(m-2)})\Delta_{kj}(\mathbf{h}^{(m)})$. One iterates this process recursively until it reaches $\langle x_i(0)^2\rangle$, thus contributing another $-1/N$ to the autocorrelation function. Things are different for $t \geq N$, since terms involving products of the slopes for the same direction will enter the procedure at a certain point in the iteration. Because of these terms, for $t \geq N$ autocorrelation functions will be identically zero.

[1] M. Lüscher, Nucl. Phys. B **418**, 637 (1994).
[2] G. M. de Divitiis, R. Frezzotti, M. Guagnelli, M. Masetti, and R. Petronzio, Nucl. Phys. B **455**, 274 (1995).
[3] P. de Forcrand, Parallel Comput. **25**, 1341 (1999).
[4] F. R. Krajewski and M. Parrinello, Phys. Rev. B **71**, 233105 (2005).
[5] F. R. Krajewski and M. Parrinello, Phys. Rev. B **73**, 041105(R) (2006).
[6] P. de Forcrand, Phys. Rev. E **59**, 3698 (1999).
[7] W. Wilcox, Nucl. Phys. B **106**, 1064 (2002).
[8] J. Goodman and A. D. Sokal, Phys. Rev. D **40**, 2035 (1989).
[9] S. L. Adler, Nucl. Phys. B **9**, 437 (1989).
[10] V. I. Manousiouthakis and M. W. Deem, J. Chem. Phys. **110**, 2753 (1999).
[11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran 77* (Cambridge University Press, Cambridge, 1986).
[12] H. A. Van Der Vorst, SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput. **13**, 631 (1992).
[13] S. L. Adler, Phys. Rev. D **23**, 2901 (1981).