Cent. Eur. J. Math. • 11(4) • 2013 • 787-799
DOI: 10.2478/s11533-012-0164-x

VERSITA

Central European Journal of **Mathematics**

Combining stochastic and deterministic approaches within high efficiency molecular simulations

Research Article

Bruno Escribano[1*], Elena Akhmatskaya[1†], Jon I. Mujika[2]

1 Basque Center for Applied Mathematics, Alameda de Mazarredo 14, 48009 Bilbao, Spain

2 Kimika Fakultatea, Euskal Herriko Unibertsitatea (UPV/EHU) and Donostia International Physics Center, PK 1072, 20080 Donostia, Spain

Received 21 June 2012; accepted 13 September 2012

Abstract: Generalized Shadow Hybrid Monte Carlo (GSHMC) is a method for molecular simulations that rigorously alternates Monte Carlo sampling from a canonical ensemble with integration of trajectories using Molecular Dynamics (MD). While conventional hybrid Monte Carlo methods completely re-sample particle's velocities between MD trajectories, our method suggests a partial velocity update procedure which keeps a part of the dynamic information throughout the simulation. We use shadow (modified) Hamiltonians, the asymptotic expansions in powers of the discretization parameter corresponding to timestep, which are conserved by symplectic integrators to higher accuracy than true Hamiltonians. We present the implementation of this method into the highly efficient MD code GROMACS and demonstrate its performance and accuracy on computationally expensive systems like proteins in comparison with the molecular dynamics techniques already available in GROMACS. We take advantage of the state-of-the-art algorithms adopted in the code, leading to an optimal implementation of the method. Our implementation introduces virtually no overhead and can accurately recreate complex biological processes, including rare event dynamics, saving much computational time compared with the conventional simulation methods.

MSC: 82B80, 65P10

Keywords: Hybrid Monte Carlo • Shadow Hamiltonian • Molecular dynamics

1. Introduction

Molecular simulations are commonly approached by two basic simulations methods: Molecular Dynamics (MD), which numerically integrates Newton's equations of motion to deterministically predict the time evolution of a molecular system,

* *E-mail: bescribano@bcamath.org*
† *E-mail: akhmatskaya@bcamath.org*

and Monte Carlo (MC) simulations, which generate a random walk in the phase space of the system. A third approach is the hybrid Monte Carlo method (HMC) [7] which alternates MC proposal steps with short MD trajectories followed by a Metropolis acceptance test, where each trajectory is accepted or rejected with a Boltzmann probability $e^{-\Delta H/k_B T}$, where ΔH is the energy change in the tested trajectory. MD and MC are highly complementary methods (deterministic vs. stochastic), so the HMC method can take advantage of the combined properties of both.

MD trajectories are integrated using symplectic numerical methods, which conserve energy and provide deterministic dynamical data. The MC proposal steps offer a faster exploration of the phase space and maintain constant temperature by sampling from a Boltzmann distribution. However, in order to preserve the dynamical information between MD trajectories, the HMC method needs to be generalized so that particle velocities are only partially updated at the MC step, instead of being completely reset. This led to the generalized hybrid Monte Carlo method (GHMC) [13, 16] which includes a partial velocity update procedure.

Another limitation of HMC, and by extension of GHMC, is that the acceptance rate of the MD step decreases with increasing size of molecular systems due to discretization errors introduced by the numerical integrator [15]. This implies that HMC methods can be highly inefficient for large bio-molecular systems. This limitation is remedied by using importance sampling with respect to shadow Hamiltonians, as was introduced by Izaguirre et al. [14]. Shadow Hamiltonians are modified energies derived by asymptotic expansions in powers of the discretization step-size. Their advantage is that symplectic integrators commonly used in MD, like the Störmer–Verlet method [10], conserve shadow Hamiltonians to higher accuracy than true Hamiltonians [23]. The introduction of shadow Hamiltonians into the GHMC method led to the generalized shadow hybrid Monte Carlo (GSHMC) method [4], which overcomes the limitations of HMC methods while offering most of the advantages of both MD and MC methods.

In [25], GSHMC is applied to coarse-grained simulations of biological systems like phospholipid bilayers in cell membranes, showing an improvement in sampling efficiency over regular MD methods. In that case, the authors use a workaround implementation, in which shell scripting alternates short MD trajectories and Monte Carlo sampling. In order to obtain all of its potential advantages, it is necessary to implement GSHMC within a highly efficient MD code. In this work we present the implementation of GSHMC into the well-known MD software GROMACS [12] and demonstrate its performance and accuracy on computationally expensive systems like proteins.

GROMACS is a software package that is mainly used to perform MD simulations, but it also includes many tools that analyze dynamical and non-equilibrium thermodynamical properties of the simulated system. Our modification introduces virtually no overhead; it does not restrict any other functionalities; and it can accurately recreate complex biological processes, including rare event dynamics, improving computational efficiency when compared with conventional simulation methods. By being implemented into the highly parallelized code GROMACS, the GSHMC method takes advantages of the state-of-the-art algorithms adopted in the code. This leads to the optimal implementation of the method, and improves its sampling efficiency by combining it with other enhanced sampling methods supported in the GROMACS package.

As a testing system we have chosen serum Transferrin (sTf), a member of the Transferrin family of enzymes which are present in vertebrates and some insects. The physiological role of sTf is of major importance, as it controls the level of free Fe^{+3} in our bloodstream by capturing and transporting it to cells [1, 17]. In addition, this protein is able to transport other cations, such as aluminum. A number of simulations of Transferrin have been conducted before. In particular, we focus on the work of Rinaldo and Field [22] to set the initial state of our system. In our simulations we will represent the N-lobe of human serum transferrin (the C-lobe has not been crystallized yet), which is composed of 5027 atoms, and which was already used in our previous simulations without GSHMC [21].

The paper is organized as follows: in Section 2 we provide the algorithmic summary for the GSHMC method and describe how it has been implemented in GROMACS; in Section 3 we present results from our simulations with the serum Transferrin protein comparing them with similar simulations performed with classical MD methods; in Section 4 we review the advantages obtained by the implementation of GSHMC within an already optimized MD software.

2. Methods

2.1. GSHMC algorithmic summary

The original GSHMC method [4], was intended to be an extension of the hybrid Monte Carlo method, in which short MD trajectories serve as an intelligent move in the MC procedure. In our case however, since we aim to make use of the high efficiency MD algorithms implemented in GROMACS, we propose an alternative approach to the method in which MC sampling is periodically introduced during a long MD trajectory. In this way, we are only running one simulation, which only requires to be loaded and initialized once, but the full trajectory is divided into shorter trajectories and MC sampling is performed between them.

The resulting GSHMC method is composed of two main alternating steps: a short molecular dynamics trajectory, and a partial velocity update including Monte Carlo sampling. Both steps must pass a Metropolis acceptance test in which they will be accepted or rejected according to the energy change during the last step. See Figure 1 for a flowchart of the GSHMC method.

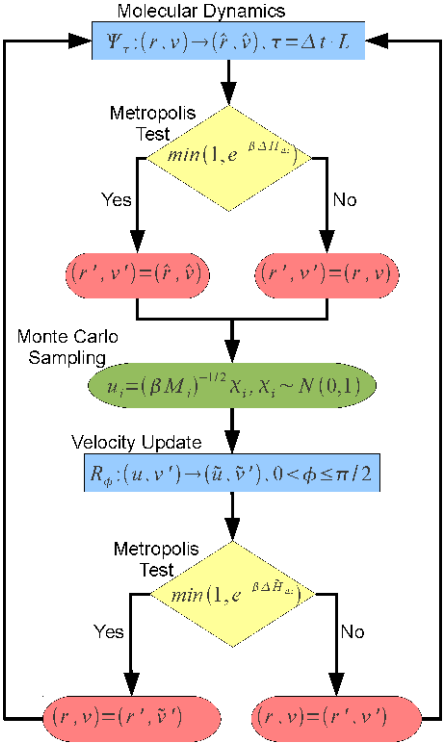

Figure 1.

In the case of a rejection of the molecular dynamics proposal, we need to introduce a momentum flip $\mathcal{F}$: $(r,v)\mapsto(r,-v)$ to validate the standard detailed balance condition [18],

$$A(\Gamma'|\Gamma)\rho(\Gamma)=A(\Gamma|\Gamma')\rho(\Gamma'),$$

which in turn verifies the stationarity of a probability density function $\rho(\Gamma)$ under a given Markov chain, i.e.,

$$\rho(\Gamma')=\int A(\Gamma'|\Gamma)\rho(\Gamma)\,d\Gamma,$$

where the state space of a Markov chain, $\Omega \subset \mathbb{R}^N$, consists of states $\Gamma \in \Omega$, and its transition probability kernel is $A(\Gamma'|\Gamma)$, and Γ' denotes a proposal state.

In practice, however, a momentum flip can be overcome as discussed in [2–4], as long as acceptance rates are maintained above 75–80% [3]. We have tested both implementations, with and without momentum flip. Our numerical results indicate that the implementation with momentum flip displays a favorable behavior in terms of sampling efficiency, whereas avoiding the momentum flip has advantages in reproducing dynamical properties of the system. For the discussion and practical issues we refer readers to [2, 3]. Both approaches are implemented in the code.

Molecular dynamics trajectory

This step consists of the integration of a short MD trajectory using the appropriate symplectic integrator, e.g., Verlet, followed by a Metropolis acceptance test.

1. *Molecular dynamics*: Given an accepted state $\Gamma_i = (r, v, t)$, where r is a collective vector of atomic positions, v is a collective vector of atomic velocities and t is time, we apply a time reversible and volume preserving method $\Psi_{\Delta t}$ over L steps of step-size Δt to the current state. The proposal state is defined by

$$\widehat{\Gamma}_i = (\widehat{r}, \widehat{v}, t + L\Delta t), \qquad \text{with} \quad \widehat{r} = \Psi_\tau(r)$$

for $\tau = L\Delta t$ and a given integer $L \geq 1$.

2. *Metropolis test*: The next accepted state Γ_{i+1} is found through a modified Metropolis accept/reject criterion

$$(r', v', t + L\Delta t) = \Gamma_{i+1} = \begin{cases} \widehat{\Gamma}_i & \text{with probability} \quad \min\left(1, e^{-\beta \Delta H_{\Delta t}}\right), \\ \mathcal{F}\Gamma_i & \text{otherwise,} \end{cases}$$

using shadow Hamiltonians $H_{\Delta t}$ instead of true Hamiltonians:

$$\Delta H_{\Delta t} = H_{\Delta t}(\widehat{r}, \widehat{v}) - H_{\Delta t}(r, v).$$

In the case of rejection we apply a momentum flip $\mathcal{F}\colon (r, v) \mapsto (r, -v)$, unless we have decided to avoid it, following [2, 3].

Partial velocity update

This part of GSHMC consists of a proposal step and a Metropolis accept/reject criterion adapted for this step.

1. *Monte Carlo sampling*: The velocity vector v' is mixed with a $3N$-dimensional noise vector u. The partial velocity refreshment R_ϕ is given by

$$\begin{pmatrix} \widetilde{u} \\ \widetilde{v}' \end{pmatrix} = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} u \\ v' \end{pmatrix}, \tag{1}$$

where $0 < \phi \leq \pi/2$ is a given angle and $u \sim \mathrm{N}[0, (\beta M)^{-1/2}]$ is the $3N$-dimensional normally distributed vector with zero mean and covariance matrix $(\beta M)^{-1/2}$, M is the diagonal mass matrix of the molecular system, $\beta = 1/k_B T$ is the inverse temperature. We denote the proposed state vector by $\widetilde{\Gamma}_{i+1} = (r', \widetilde{v}', t + L\Delta t)$.

2. *Metropolis test*: The accepted state $\overline{\Gamma}_{i+1}$ is found through a Metropolis accept/reject criterion:

$$(r, v, t + L\Delta t) = \overline{\Gamma}_{i+1} = \begin{cases} \widetilde{\Gamma}_{i+1} & \text{with probability} \quad \min\left(1, e^{-\beta \Delta \widetilde{H}_{\Delta t}}\right), \\ \Gamma_{i+1} & \text{otherwise,} \end{cases} \tag{2}$$

with

$$\Delta \widetilde{H}_{\Delta t} = H_{\Delta t}(r', \widetilde{v}') + \frac{1}{2}\widetilde{u}^T M \widetilde{u} - H_{\Delta t}(r', v') - \frac{1}{2} u^T M u. \tag{3}$$

The angle ϕ in equation (1) is a tunable parameter of the method and defines how much of the dynamics is kept or disregarded during the sampling. With $\phi = 0$ all dynamical information is preserved and there is no gain in sampling. With $\phi = \pi/2$ the velocities are completely regenerated as in a standard HMC method. Setting the angle with $\phi = \sqrt{2\gamma\tau}$ provides a statistically rigorous implementation of stochastic Langevin dynamics [2], where γ is the friction coefficient.

At this point, in the case of rejection, it is possible to repeat the velocity update several times until it is accepted. This is relatively inexpensive compared to the MD step and can very easily raise the acceptance rate. An additional strategy for improving the acceptance rate of the partial velocity update step is to apply a variable change to the velocity vector v as described in [4, 24]. This method consists of proposing a modified velocity vector $\overline{v}$ defined by

$$\overline{v} = v - \frac{\Delta t}{24M}(F(t+\Delta t) - F(t-\Delta t)), \tag{4}$$

where F represents the atomic forces, which need to be calculated one time-step backward and forward in time. We can then apply the partial velocity update in equation (1) to the modified velocities. The variable change needs to be undone before evaluating the shadow Hamiltonian for the Metropolis test. Both strategies aimed to increase the acceptance rate at the partial velocity update step are implemented in our code.

Modified Hamiltonians

Modified or 'shadow' Hamiltonians are asymptotic expansions of the true Hamiltonian in powers of the step-size Δt. They are exactly conserved by symplectic integrators like Störmer–Verlet [10] and are relatively inexpensive to calculate. Efficient algorithms for computing modified energies can be found in [23]. In our implementation we have included 4th and 6th order approximations following the alternative method suggested by Akhmatskaya and Reich [4] for the Stoïmer–Verlet method.

The 4th order shadow Hamiltonian then is given by the following equation:

$$H^4_{\Delta t} = U + \frac{1}{2}\dot{R}[M\dot{R}] + \frac{\Delta t^2}{12}\dot{R}[M\dddot{R}] - \frac{\Delta t^2}{24}\ddot{R}[M\ddot{R}],$$

where U is the potential energy, R is the positions vector and M is the atomic mass matrix. The derivatives of the positions are obtained using the centered difference approximation, which up to 6th order Hamiltonians requires to save the positions for $t = n-3, \ldots, n, \ldots, n+3$, where n is the time step at which we are evaluating the shadow Hamiltonian. The order of approximation of modified Hamiltonians used in the simulation also affects the acceptance rates. Higher approximation orders provide better acceptance rates, but they also require more time to compute.

For arbitrary $p = 2m$, $m \geqslant 2$, the p-order shadow Hamiltonian can be derived with the following expression [4]:

$$H^{[p]}_{\Delta t} = \sum_{i=1}^{p-1}\left\{\sum_{j=0}^{i-1}(-1)^j\left[\frac{d^j}{dt^j}\frac{\partial \mathcal{L}^{[p]}_{\Delta t}}{\partial R^{(i)}}\right]R^{(i-j)}\right\} - \mathcal{L}^{[p]}_{\Delta t},$$

where $\mathcal{L}^{[p]}_{\Delta t}$ is the p-order modified Lagrangian density and $R^{(i)}$ is the i-order derivative of the positions vector calculated by the centered differences method.

Data analysis

The key novel step of the proposed GSHMC method is the importance sampling with respect to a shadow Hamiltonian $H_{\Delta t}$. This improves the acceptance rate of the Metropolis test for MD trajectories; however, sampling from a modified ensemble implies that expected values have to be re-weighted. Let $\{\Gamma_i\}_{i=1}^{I}$ denote a sequence of I accepted states from a GSHMC simulation with $\Gamma_i = (Y_i, t_i)$ and $Y_i = (r_i, v_i)$. Average values of a function $f(Y)$ with respect to the canonical density are computed according to the formula

$$\langle f \rangle = \frac{\sum_{i=1}^{I} w_i f(Y_i)}{\sum_{i=1}^{I} w_i}$$

with weight factors

$$w_i = \exp\left(-\beta\left(H(Y_i) - H_{\Delta t}(Y_i)\right)\right).$$

2.2. Implementation into GROMACS

Implemented algorithm

We have implemented the GSHMC method into the GROMACS molecular dynamics software [12]. GROMACS is an open-source package available under the GNU General Public License. It is written in the C programming language, highly optimized for maximum computational efficiency and fully parallelized using the MPI protocol. GROMACS is intended for molecular dynamics simulations and energy minimization. Its high efficiency makes it one of the most popular codes for the simulation of large molecular systems like proteins. We have included our development in version 4.5.4 and are planning to port it to the latest development branch which will benefit from GPU parallelization. We currently offer the development version of our code upon request to the corresponding author.

For the time integration of Newton's equations of motion, GROMACS counts with several symplectic algorithms like the leap-frog and the velocity Verlet method. Our implementation of GSHMC is included within the latter algorithm, inside the main time-step loop of the function do_md(), see Figure 2, which is called at the end of the function mdrunner(), after the initial atomic positions and all simulation parameters have been loaded from the tpr file. Additional external routines need to be implemented for saving/restoring states, calculating shadow Hamiltonians, evaluating Metropolis tests and performing the partial velocity updates. The relative encapsulation of this method suggests that the same behavior can be achieved by executing a shell script that alternates calls to the GROMACS executables (grompp and mdrun) with calls to a GSHMC algorithm implemented externally. However, such an approach is very inefficient due to the time spent in file reading and writing, which would have to be repeated before and after every short MD trajectory. On the other hand, implementing the algorithm within the MD software ensures that the relevant data structures are always loaded in memory and the new method does not introduce any additional I/O operations.

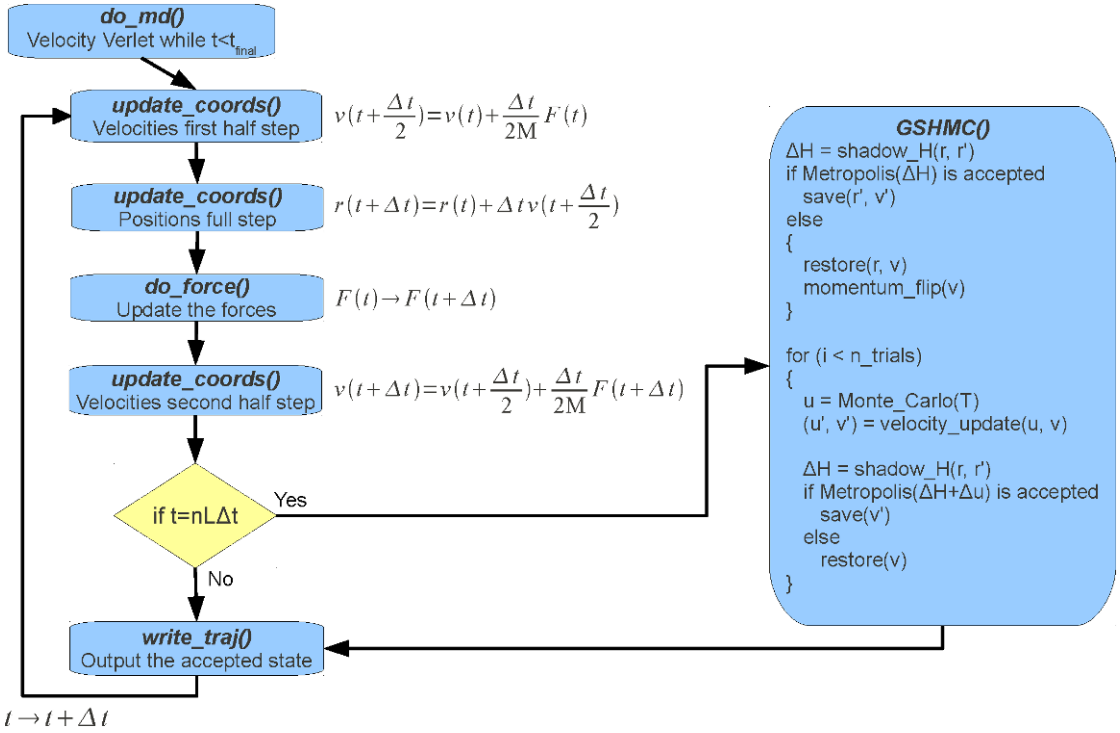

Figure 2.

The execution of a simulation with the GSHMC method is similar to standard molecular dynamics except that every L time-steps, after completing the Verlet integration but before writing the current state to the output files, we invoke the GSHMC algorithm. The algorithm begins by calculating the modified Hamiltonians as explained in the previous section, both for the current state and for the saved state. Since we are using the centered differences method to calculate derivatives, we require the positions vector for several time-steps before and after the current state for calculating the shadow Hamiltonian. We also require the potential energies for the current and for the saved state. We then use the calculated Hamiltonians to perform a Metropolis test to decide if we accept the current state or restore the state saved L time-steps before. If the test is rejected, we restore the state of the system that we saved at the beginning of the last L time-steps and, if momentum flip is used, we negate the velocities. If the test is accepted, we keep the current state and move on to the partial velocity update step. In this step we first generate new random velocities by sampling from a Maxwell–Boltzmann distribution at a fixed temperature T. We then perform the partial velocity update described in equation (1). If we are using the change of variables described in (4), we require the forces one time-step before and after the current state. Finally, we do a second Metropolis test as indicated in equations (2) and (3) to decide if we want to keep the newly updated velocities or restore the previous ones. This part of the algorithm can be repeated several times until we find a suitable set of velocities. Once we have an accepted state, we can save the values for the current coordinates, velocities and energies and move on to writing to the output files.

Parameters

Several new parameters need to be added to GROMACS in order to use and tune GSHMC with optimal efficiency, see Table 1. Such parameters can be loaded at run time from an additional input file, but it is preferable to introduce them through the input MD parameters file (mdp). Here we followed the standard programming guidelines described by the GROMACS developing team [26]. This procedure implies adding new variables to the inputrec data structure, which will store the new parameters and make them available throughout the runtime execution. Further modifications for reading from file, loading in memory and checking the values are straightforward and have been implemented according to the standard procedure.

Table 1. Parameters used by the GSHMC method.

Parameter	Value	Description
GSHMC	yes/no	Choose whether to run GSHMC or regular MD
flip	yes/no	Choose whether to perform a velocity reversal after a rejection
GS2HMC	yes/no	Include a variable change in the velocity update to increase the acceptance rate
ϕ	0 to $\pi/2$	Angle for the partial velocity update
n_{trials}	≥ 1	Number of velocity update trials to be performed
L	~ 1000	Length of the MD trajectory between Monte Carlo tests
T	> 0	Temperature used for the generation of the random velocities in the momentum updates
p	4 or 6	Order of approximation for the shadow Hamiltonians

The sampling efficiency of the method varies greatly with the choice of GSHMC parameters. It is important to keep the acceptance rate of both the MD trajectories and the velocity updates as high as possible. If many trajectories are rejected then we are wasting much computational time. If many velocity updates are rejected then we will not gain much from the Monte Carlo sampling.

The parameter ϕ has a strong influence on the velocity update step. Larger values of ϕ are beneficial for GSHMC sampling. However, the acceptance rate drops quickly as ϕ is increased. In general, we need $\sim$ 85% of accepted proposals to obtain sampling advantages and to keep the system thermalized. The acceptance rate can be further improved by using the GS2HMC method, in which the velocity update is performed with a change of variables like the one proposed in equation (4). Another option is to perform several update trials until one of them is accepted. Since the velocity update step is relatively inexpensive in the computational sense, we can very easily raise the acceptance rate in this way without introducing much computational overhead.

Other tunable parameters are the step-size Δt and the number of time-steps L in an MD trajectory. Δt should be chosen so that the desired accuracy of energy approximation using shadow Hamiltonians is achieved. Longer trajectories and longer time-steps in general would decrease the acceptance rate but at the same time can improve sampling compared to short trajectories and small time-steps, if the acceptance rate is kept reasonably high.

Simulation setup

In general GSHMC is consistent with the algorithms implemented in GROMACS. There are, however, some limitations that need to be considered. First place, all MD trajectories are integrated in the NVE ensemble, so there is no temperature coupling. The temperature will be maintained constant by the Monte Carlo sampling. Second, the only available integration scheme in the current implementation of GSHMC is velocity Verlet. Because of the way GROMACS is structured, combining GSHMC with other integrators implemented in GROMACS would require further modifications to the source code. Finally, in order to ensure the high accuracy of the modified Hamiltonians, the LINCS algorithm for applying constraints must be used with a 6th or 8th order of approximation and with at least two iterations [11].

Parallelization

We have employed the parallelization of MD trajectory integration that is already implemented in standard GROMACS. This includes the possibility of using either domain decomposition or particle decomposition [12]. In any case, the computational expense of including GSHMC (mainly shadow Hamiltonians) is small enough compared to MD integration (mainly force evaluations) as all extra calculations can be performed on the master node without any noticeable decrease in performance. Only a few modifications must be included to be able to run GSHMC simulations in parallel. In first place, it is necessary to gather the information about atomic positions from all processors before calculating the shadow Hamiltonian, which represents a significant amount of communication but is only required every L time-steps. Additionally, after each Metropolis test we have to broadcast the result to all processors. Finally, after each accepted proposal step, we have to scatter the new state to all processors.

3. Results and discussion

We have tested the new implementation of GSHMC in GROMACS on a variety of systems including atomic, coarse-grained and meso-scale modeling. Due to its fast sampling rate, the GSHMC algorithm is specially useful when simulating large macromolecules like proteins. Here we present benchmark results taken from the ongoing research project studying metal binding and release by the protein serum Transferrin (sTf), a member of the Transferrin family of enzymes that are present in vertebrates and some insects. The physiological role of sTf is of major importance, as it controls the level of free Fe^{+3} in our bloodstream by capturing and transporting it to cells [1, 17, 21]. In addition, this protein is able to transport other cations, such as Al^{+3}.

In our simulations we only represented the N-lobe of human sTf (the C-lobe has not been crystallized yet), which is composed of 5027 atoms. The CHARMM27 all-atom force field [20] was employed to build the topology of the protein. The steepest descent (SD) method was employed to minimize the energy of the system. Periodic boundary conditions were applied in all directions using a rhombic dodecahedron cell, with a minimal distance between the protein and the wall of the cell set to 10 Å. The system was solvated with 13218 water molecules using the TIP3P water model [9]. We then run a 100 ps long equilibration imposing weak constrains on the atomic positions of the protein. GSHMC simulations achieve constant temperature by sampling from a Maxwell–Boltzmann distribution at 300 K and do not require any additional thermostat. Other standard MD simulations shown here were performed under canonical thermodynamic ensemble (NVT), where the temperature of the protein and the rest of the system were independently coupled to 300 K using velocity rescaling with a stochastic term (V-rescale) algorithm [5]. For both simulation approaches, long-range Coulombic electrostatics were calculated using the smooth particle mesh Ewald (PME) method [6, 8] with a cut-off of 12 Å and a spacing of the Fourier grid of 1.2 Å. For the van der Walls non-bonded interactions, a Lennard–Jones potential with a cut-off radius of 12 Å and a switch function with a radius of 9 Å were employed. Bond stretching and bond bending were constrained during the simulation using Linear Constraint Solver (LINCS) [11], allowing an integration time step of 2 fs. Total simulation times for production runs were between 20 and 60 ns.

For the GSHMC method some additional parameters are necessary to optimize the performance, see Table 1. In all GSHMC simulations discussed in the next section we used velocity reversal after rejection, no variable change in the velocity update, $\phi = 1.5$, $n_{\text{trials}} = 5$, $L = 1000$, and 6th order shadow Hamiltonians. With this set of parameters we have obtained an acceptance rate of at least 99% for the MD step and 90% for the momentum update. Due to the computational cost of these simulations, we did not repeat the long runs without momentum flip.

3.1. Protein folding simulations

When Transferrin is bound to a metal ion such as Al^{+3}, it adopts a closed or folded configuration. When this metal ion is released, the Transferrin protein will unfold to an open configuration. In our simulations we try to reproduce this unfolding process both with standard MD and with GSHMC simulations. We can then compare the resulting spatial configurations to test the accuracy of the GSHMC method and the employed computational time to evaluate the performance.

We start with the closed configuration which is obtained experimentally from X-ray diffraction analysis [19]. We then remove the metal ion and allow the protein to unfold until it reaches a stable or equilibrium configuration. In Figure 3 (a) we represent this unfolding process by plotting the distance measured between the center of mass of the two sub-domains that form the protein. We can see that both MD and GSHMC simulations can reproduce the unfolding properly. However, in the GSHMC simulation the process starts after 2–3 ns of simulation, while regular MD requires more than 25 ns of simulation before the unfolding begins. In Figure 3 (b) we see the result of a similar experiment but in this case we plot the distances between two of the residues that bind the metal to the protein. Once again we see that the GSHMC simulation starts unfolding almost immediately and it reaches the same final configuration as the MD simulation but much sooner. GSHMC simulations shown in Figure 3 are shorter than MD simulations because there was no need to continue them once the equilibrium state had been found. In general, GSHMC can obtain similar results to MD but saving at least half of the simulation time.

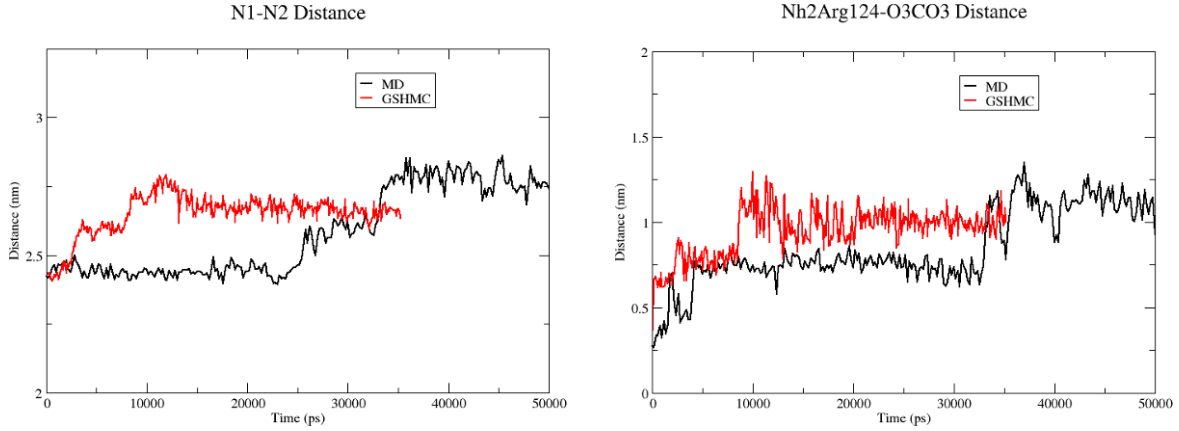

Figure 3.

In Figure 4 we plot the normalized autocorrelation function (ACF) for the distances shown in Figure 3. It is clear that in both cases the simulations performed with GSHMC reach the equilibrium value more rapidly than those performed with MD. For a more rigorous measure of sampling efficiency we calculate the integrated autocorrelation function [16] for a time series Ω_k of K samples, where Ω is an observable,

$$A_\Omega = \sum_{l=1}^{K'} \text{ACF}(\tau_l);$$

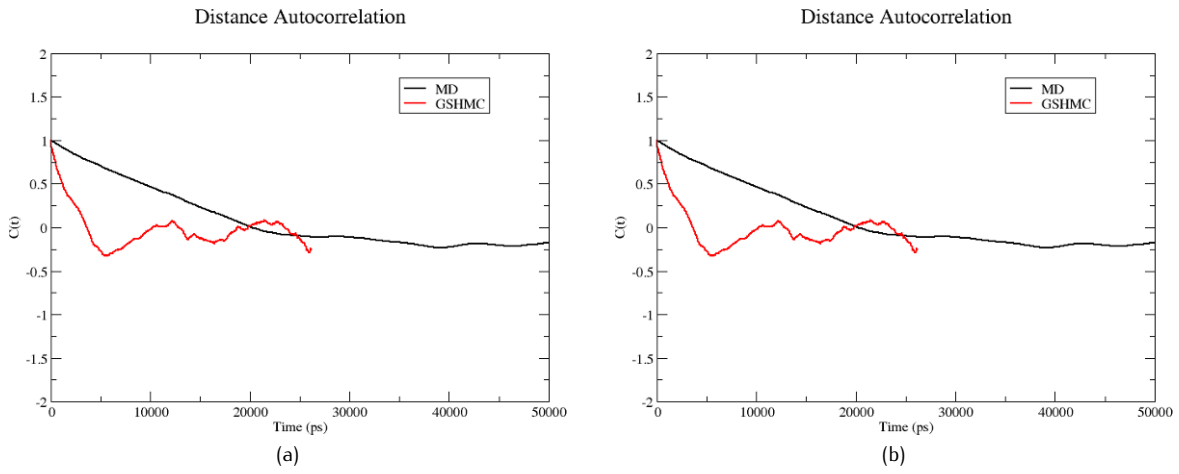

Figure 4.

here $\mathrm{ACF}(\tau_l)$, $l = 0, \ldots, K' < K$, is the value of the autocorrelation function for the time series Ω_k. To compare the sampling efficiency of both simulation methods we compute the ratio $A_\Omega^{\mathrm{MD}}/A_\Omega^{\mathrm{GSHMC}}$.

We consider separately two time ranges in each simulation: the unfolding phase, where the measured distances are increasing, and the equilibrated phase, where the protein has finished unfolding and the distances remain stable. In the unfolding phase we obtain a ratio $A_{\mathrm{Unfold}}^{\mathrm{MD}}/A_{\mathrm{Unfold}}^{\mathrm{GSHMC}}$ of 7.03 for the case of Figure 4 (a) and 4.06 for Figure 4 (b). In the equilibrated phase the ratios $A_{\mathrm{Equil}}^{\mathrm{MD}}/A_{\mathrm{Equil}}^{\mathrm{GSHMC}}$ are 1.50 and 1.45 respectively. The higher ratios for the unfolding phase mean that GSHMC is specially efficient for sampling conformational changes, but even in the equilibrated phase GSHMC still samples ~ 1.5 times more efficiently than MD.

3.2. Computational performance and accuracy

To assess the accuracy of GSHMC compared to MD and its capability for calculating thermodynamical properties we compared several averaged magnitudes for the equilibrated configurations of the simulations shown in Figure 3. In all cases the values are averaged over the last 10 ns of simulation, after the protein has finished unfolding and the system can be considered equilibrated.

We can see in Table 2 that the average values obtained by GSHMC are in agreement with those obtained with standard MD. In the case of the MD simulation the temperature was coupled to 300 K using velocity rescaling with a stochastic term (V-rescale) algorithm [5], while in the GSHMC case temperature was kept constant by sampling from a Boltzmann distribution at a fixed temperature $T = 300\,\mathrm{K}$. The total potential energy E_{pot} is slightly lower in the GSHMC simulation, but within the approximation and statistical error.

Table 2. Equilibrium averages ± standard deviations of observables.

Simulation	T (K)	E_{pot} (kJ/mol)	E_{kin} (kJ/mol)	E_{LJ} (kJ/mol)	E_{Coul} (kJ/mol)
MD	299.98 ± 0.01	-605488 ± 12	111692 ± 23	4836.96 ± 1.7	66472.0 ± 43
GSHMC	299.87 ± 0.02	-617350 ± 11	111649 ± 11	4845.98 ± 2.0	66017.8 ± 45

To benchmark the computational performance of GSHMC we ran two identical simulations using both GSHMC and MD methods for 24 hours on the same server using one node with 8 processors for each method. The GSHMC performance reached 14.67 GFlops producing 0.807 ns/day, while the MD performance was 14.95 GFlops producing 0.825 ns/day. The resulting overhead introduced by GSHMC is less than 2%, which is negligible compared with the improvement in

sampling efficiency provided by the method. The total simulated time for GSHMC was 793 ps, while MD completed 816 ps. The small difference resides in the time spent performing velocity updates and in rejected MD trajectories, which ideally should be kept to a minimum (< 5%) by choosing a proper set of GSHMC parameters.

We have tested parallel scaling by measuring computing performance from 1 to 8 processors for both MD and GSHMC methods. In Figure 5 we can see that both methods scale almost linearly up to 8 processors, showing that GSHMC does not affect the parallelization efficiency of GROMACS.

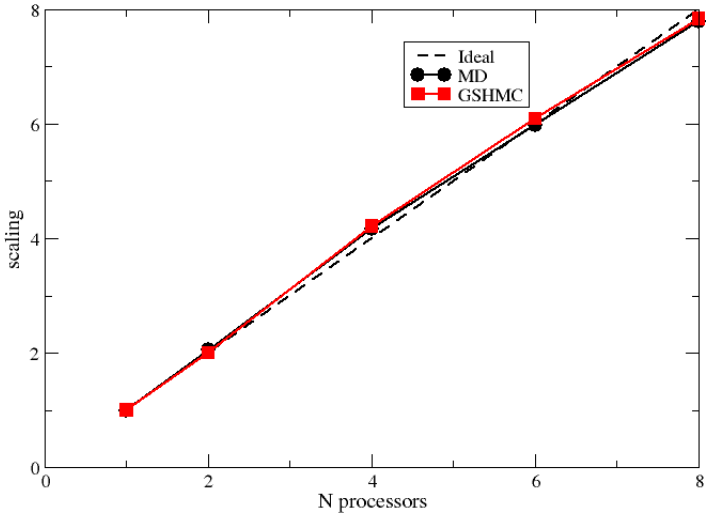

Figure 5.

4. Conclusions

We have successfully implemented the Generalized Shadow Hybrid Monte Carlo method within the molecular dynamics package GROMACS. The implementation implies only moderate modifications to the original source code that do not interfere with the standard operation and performance of MD simulations. To the best of our knowledge, this is the first time that any hybrid Monte Carlo method has been implemented in the GROMACS package.

Our implementation can accurately reproduce biological processes such as protein folding at a much smaller computational cost than conventional MD. We have tested GSHMC by simulating the conformational change of the protein serum Transferrin. We have shown that the new method accurately recreates the dynamics of such a complex bio-molecular system, and finds the equilibrium state in complete agreement with NVT MD simulations, but saves much computational time.

Sampling with modified Hamiltonians and using a partial velocity refreshment step have improved sampling efficiency at the expense of losing some dynamical information. This does not pose a problem for the numerous applications aimed at finding equilibrium or minimal energy configurations, such as binding energy calculations, phase transitions, etc. Choosing the parameters of GSHMC with care will provide accurate dynamical information, as shown in [2, 3]. In general the GSHMC method offers a better compromise between accuracy and performance than traditional MD. With the additional use of shadow Hamiltonians instead of total energies during the Metropolis tests, we have raised the acceptance rate for MD trajectories to almost 100%, even in the case of macromolecular systems, which is a great advantage over the standard hybrid Monte Carlo methods.

The computational performance for GSHMC simulations is comparable to that of the unmodified version of GROMACS. The new method introduces almost no overhead and is fully compatible with the parallelization schemes available in GROMACS such as domain decomposition and particle decomposition using MPI. We included our implementation in

the 4.5.4 version and are planning to port it to the latest version, as we expect the advantages offered by GSHMC will be of general interest for the GROMACS user community.

Acknowledgements

The authors would like to thank for the financial support from MTM2011-24766 funded by MICINN. The SGI/IZO-SGIker UPV/EHU is acknowledged for computational resources. We also thank our colleague from the UPV/EHU in Donostia, Prof. Xabier López for valuable discussions.

References

[1] Aisen P., Transferrin, the transferrin receptor, and the uptake of iron by cells, In: Metal Ions in Biological Systems, 35, Marcel Dekker, New York, 1998, 585–631
[2] Akhmatskaya E., Bou-Rabee N., Reich S., A comparison of generalized hybrid Monte Carlo methods without momentum flip, J. Comput. Phys., 2009, 228(6), 2256–2265
[3] Akhmatskaya E., Bou-Rabee N., Reich S., Erratum to "A comparison of generalized hybrid Monte Carlo methods with and without momentum flip" [J. Comput. Phys. 228 (2009) 2256–2265], J. Comput. Phys., 2009, 228(19), 7492–7496
[4] Akhmatskaya E., Reich S., GSHMC: An efficient method for molecular simulation, J. Comput. Phys., 2008, 227(10), 4934–4954
[5] Bussi G., Donadio D., Parrinello M., Canonical sampling through velocity rescaling, J. Chem. Phys., 2007, 126(1), #014101
[6] Darden T., York D., Pedersen L., Particle mesh Ewald: An *N*-log(*N*) method for Ewald sums in large systems, J. Chem. Phys., 1993, 98(12), 10089–10092
[7] Duane S., Kennedy A.D., Pendleton B.J., Roweth D., Hybrid Monte Carlo, Phys. Lett. B, 1987, 195, 216–222
[8] Essmann U., Perera L., Berkowitz M.L., Darden T., Lee H., Pedersen L.G., A smooth particle mesh Ewald potential method, J. Chem. Phys., 1995, 103(19), 8577–8593
[9] Jorgensen W.L., Chandrasekhar J., Madura J.D., Impey R.W., Klein M.L., Comparison of simple potential functions for simulating liquid water, J. Chem. Phys., 1983, 79(2), 926–935
[10] Hairer E., Lubich C., Wanner G., Geometric Numerical Integration, Springer Ser. Comput. Math., 31, Springer, Berlin–Heidelberg, 2002
[11] Hess B., Bekker H., Berendsen H.J.C.. Fraaije J.G.E.M., LINCS: A linear constraint solver for molecular simulations, J. Comput. Chem., 1997, 18(12), 1463–1472
[12] Hess B., Kutzner C., van der Spoel D., Lindahl E., GROMACS 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation, J. Chem. Theory Comput., 2008, 4(3), 435–447
[13] Horowitz A.M., A generalized guided Monte Carlo algorithm, Phys. Lett. B, 1991, 268(2), 247–252
[14] Izaguirre J.A., Hampton S.S., Shadow hybrid Monte Carlo: an efficient propagator in phase space of macromolecules, J. Comput. Phys., 2004, 200(2), 581–604
[15] Kennedy A.D., Pendleton B., Acceptances and autocorrelations in hybrid Monte Carlo, Nuclear Phys. B – Proceedings Supplements, 1991, 20, 118–121
[16] Kennedy A.D., Pedlenton B., Cost of the generalised hybrid Monte Carlo algorithm for free field theory, Nuclear Phys. B, 2001, 607(3), 456–510
[17] Klausner R.D., Ashwell G., van Renswoude J., Harford J.B., Bridges K.R., Binding of apotransferrin to K562 cells—explanation of the transferrin cycle, Proc. Natl. Acad. Sci. USA, 1983, 80(8), 2263–2266
[18] Liu J.S., Monte Carlo Strategies in Scientific Computing, Springer Ser. Statist., Springer, New York, 2001

[19] MacGillivray R.T., Moore S.A., Chen J., Anderson B.F., Baker H., Luo Y., Bewley M., Smith C.A., Murphy M.E., Wang Y., Mason A.B., Woodworth R.C., Brayer G.D., Baker E.N., Two high-resolution crystal structures of the recombinant N-lobe of human transferrin reveal a structural change implicated in iron release, Biochemistry, 1998, 37(22), 7919–7928

[20] MacKerell A.D., Bashford D., Bellott E.M., Dunbrack R.L., Evanseck J.D., Field M.J., Fischer S., Gao J., Guo H., Ha S., Joseph-McCarthy D., Kuchnir L., Kuczera K., Lau F.T.K., Mattos C., Michnick S., Ngo T., Nguyen D.T., Prodhom B., Reiher W.E., Roux B., Schlenkrich M., Smith J.C., Stote R., Straub J., Watanabe M., Wiórkiewicz-Kuczera J., Yin D., Karplus M., All-atom empirical potential for molecular modeling and dynamics studies of proteins, The Journal of Physical Chemistry B, 1998, 102(18), 3586–3616

[21] Mujika J.I., Escribano B., Akhmatskaya E., Ugalde J.M., Lopez X., Molecular dynamics simulations of iron- and aluminum-loaded serum transferrin: protonation of Tyr188 is necessary to prompt the metal release, Biochemistry, 2012, 51(35), 7017–7027

[22] Rinaldo D., Field M.J., A computational study of the open and closed forms of the N-lobe human serum transferrin apoprotein, Biophys. J., 2003, 85(6), 3485–3501

[23] Skeel R.D., Hardy D.J., Practical construction of modified Hamiltonians, SIAM J. Comput., 2001, 23(4), 1172–1188

[24] Sweet C.R., Hampton S.S., Skeel R.D., Izaguirre J.A., A separable shadow Hamiltonian hybrid Monte Carlo method, J. Chem. Phys., 2009, 131(17), #174106

[25] Wee C.L., Sansom M.S., Reich S., Akhmatskaya E., Improved sampling for simulations of interfacial membrane proteins: application of generalized shadow hybrid Monte Carlo to a peptide toxin/bilayer system, The Journal of Physical Chemistry B, 2008, 112(18), 5710–5717

[26] GROMACS Programmer's Guide, available at http://www.gromacs.org/Developer_Zone/Programming_Guide/Programmer