eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Tesis Doctoral/Doktorego Tesia

Adapting Hybrid Monte Carlo methods for solving complex problems in Life and Materials sciences

Autor/Egilea:
Mario Fernández Pendás

Directora/Zuzendaria:
Prof. Elena Akhmatskaya

2018

Doctoral Thesis

Adapting Hybrid Monte Carlo methods for solving complex problems in Life and Materials sciences

Author:
Mario Fernández Pendás

Advisor:
Prof. Elena Akhmatskaya

2018

This research was carried out at the Basque Center for Applied Mathematics (BCAM) within the Group Modelling and Simulation in Life and Materials Sciences. This research was supported by MINECO under Grants BES-2014-06864, MTM2013-46553-C3-1-P and MTM2016-76329-R (AEI/FEDER, EU) and also by the Basque Government through the BERC 2018-2021 program and by the Spanish Ministry of Economy and Competitiveness MINECO: BCAM Severo Ochoa accreditation SEV-2013-0323. This work has been possible thanks to the support of the computing infrastructure of the i2BASQUE academic network, the technical and human support provided by IZO-SGI SGIker of UPV/EHU and European funding (ERDF and ESF), the in-house BCAM-MSLMS group's cluster Monako, and the BCAM's cluster Hipatia.

"*Pasioa da hemen exigitzea zilegi den gutxieneko hori.*"
Berri Txarrak

Abstract

Efficient sampling is the key to success of molecular simulation of complex physical systems. Still, a unique recipe for achieving this goal is unavailable. Hybrid Monte Carlo (HMC) is a promising sampling tool offering a smart, free of discretization errors, propagation in phase space, rigorous temperature control, and flexibility. However, its inability to provide dynamical information and its weakness in simulations of reasonably large systems do not allow HMC to become a sampler of choice in molecular simulation of complex systems. In this thesis, we show that performance of HMC can be dramatically improved by introducing in the method the splitting numerical integrators and importance sampling.

We propose a novel splitting integration scheme called Adaptive Integration Approach or AIA, which leads to very promising improvements in accuracy and sampling in HMC simulations. Given a simulation problem and a time step, AIA automatically chooses the optimal scheme out of the family of two-stage splitting integrators. A system-specific integrator identified by our approach is optimal in the sense that it provides the best conservation of energy for harmonic forces.

The role of importance sampling on the performance of HMC is studied through the modified Hamiltonian Monte Carlo (MHMC) methods, sampling with respect to a modified or shadow Hamiltonian. The particular attention is paid to Generalized Shadow Hybrid Monte Carlo (GSHMC), introduced by Akhmatskaya and Reich in 2008. To improve the performance of MHMC in general and GSHMC in particular, we develop and test the new multi-stage splitting integrators, specially formulated for sampling with respect to modified Hamiltonians. The novel adaptive two-stage integration approach or MAIA, specifically derived for MHMC is presented. We also discuss in detail the adaptation of GSHMC to the NPT ensemble and provide the thorough analysis of its performance. Moreover, for the first time, we formulate GSHMC in the grand canonical ensemble. A general framework, useful for an extension of other Hybrid Monte Carlo methods to the grand canonical ensemble, is also provided.

The software development is another fundamental part of the present work. The algorithms presented in this thesis are implemented in MultiHMC-GROMACS, an in-house version of the popular software package GROMACS. We explain the details of such implementation and give useful recommendations and hints for implementation of the new algorithms in other software packages.

In summary, in this thesis, we propose the new numerical algorithms that are capable of improving the accuracy and sampling efficiency of molecular simulations with Hybrid Monte Carlo methods. We show that equipping the Hybrid Monte Carlo algorithm with extra features makes it even a "smarter" sampler and, no doubts, a strong competitor to the well-established molecular simulation techniques such as molecular dynamics (MD) and Monte Carlo. The up to 60 times increase in sampling efficiency of GSHMC over MD, due to the new algorithms in simulations of selected systems, supports such a belief.

Summary

The Hybrid Monte Carlo (HMC) method is a promising sampling tool offering a smart, free of discretization errors, propagation in phase space, rigorous temperature control, and flexibility. The HMC method appeared in the late eighties in the context of lattice field theories (Duane et al., 1987). A few years later, the HMC algorithm was extended to molecular simulations (Heermann, Nielaba, and Rovere, 1990) and then to condensed-matter systems (Mehlig, Heermann, and Forrest, 1992). The HMC method aims at combining the advantages of the molecular dynamics (MD) and Monte Carlo (MC) methods. MD allows for approximating the physical dynamics of the system while MC helps to explore the phase space more globally. In fact, HMC is a Metropolis-Hastings algorithm in which proposals are constructed using the NVE Hamiltonian flow of the system. The goal of HMC is to perform an efficient sampling in the canonical ensemble which ultimately allows for an accurate estimation of ensemble averages.

We consider Hamiltonians as

$$H(\mathbf{q},\mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + U(\mathbf{q}) \equiv A + B, \tag{1}$$

where M is the diagonal mass matrix, and $\mathbf{q} \in \mathbb{R}^{3D}$, $\mathbf{p} \in \mathbb{R}^{3D}$ are the positions and momenta, respectively (D is a system's dimension). We denote by $A = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}$ the kinetic energy, and by $B = U(\mathbf{q})$ the potential energy.

We are interested in sampling the variable $\mathbf{q} \in \mathbb{R}^{3D}$ that is distributed according to the probability $\pi(\mathbf{q})$. The target probability density function (p.d.f.) is written as

$$\pi(\mathbf{q}) \propto \exp\left(-\beta U(\mathbf{q})\right).$$

The HMC method combines an MD global move with Monte Carlo sampling in the following way. For each Monte Carlo iteration: (i) the momenta are resampled from the Maxwell-Boltzmann distribution $\rho_P(\mathbf{p})$; (ii) a proposed new state $(\mathbf{q}', \mathbf{p}')$ is generated by integrating the equations of motion with an integrator $\Psi_{\Delta t,L}$; (iii) the preservation of the desired canonical distribution $\pi(\mathbf{q},\mathbf{p})$ is ensured by a Metropolis test. Its acceptance probability can be calculated as:

$$P_A\left((\mathbf{q},\mathbf{p}) \to \Psi_{\Delta t,L}(\mathbf{q},\mathbf{p})\right) = \min\left\{1, \exp\left(-\beta\Delta H\right)\right\},$$

where

$$\Delta H = H\left(\Psi_{\Delta t,L}(\mathbf{q},\mathbf{p})\right) - H(\mathbf{q},\mathbf{p})$$

is the energy error associated to the integration scheme. A joint p.d.f. $\pi(\mathbf{q},\mathbf{p})$ is defined as

$$\pi(\mathbf{q},\mathbf{p}) = \pi(\mathbf{q})\rho_P(\mathbf{p}) \propto \exp\left(-\beta H(\mathbf{q},\mathbf{p})\right). \tag{2}$$

Therefore, HMC can be viewed as a method that samples points in phase space by means of a Markov Chain in which stochastic and dynamical transitions alternate.

The complete resampling in (i) can be replaced with the partial momentum update as proposed in (Horowitz, 1991). The current momenta are mixed with an independent and identically distributed (i.i.d.) Gaussian noise $\mathbf{u} \sim \mathcal{N}(0, \beta^{-1}M)$ to obtain

$$\begin{aligned} \mathbf{p}^* &= \quad \cos\varphi\ \mathbf{p} + \sin\varphi\ \mathbf{u} \\ \mathbf{u}^* &= -\sin\varphi\ \mathbf{p} + \cos\varphi\ \mathbf{u}, \end{aligned} \tag{3}$$

where $\varphi \in (0, \pi/2]$ controls the amount of noise introduced. The angle φ also introduces extra control over the sampling efficiency of the method and may lead to the superior performance over HMC. The idea was formalized in the Generalized Hybrid Monte Carlo (GHMC) method (Kennedy and Pendleton, 2001).

Meaning to be an improvement of both Monte Carlo and molecular dynamics, Hybrid Monte Carlo turned out to inherit two unfortunate drawbacks. Like Monte Carlo, it does not generate dynamic information, and its performance degrades with an increase of either the system size or the time step. Therefore, the goal of the thesis is to introduce the new algorithms for HMC which can potentially minimize these limitations. In order to enhance the performance of the HMC method, two main tools are considered: the splitting numerical integrators and the importance sampling technique.

The efficiency and even the feasibility of molecular dynamics simulations depend crucially on the choice of a numerical integrator. As to the role of integrators in enhancing the performance of Hybrid Monte Carlo, it has been a subject of active research in recent years (McLachlan, 1995; Blanes, Casas, and Sanz-Serna, 2014; Chao et al., 2015; Campos and Sanz-Serna, 2017; Bou-Rabee and Sanz-Serna, 2017a). The velocity Verlet algorithm is currently the method of choice; its algorithmic simplicity and optimal stability properties make it very difficult to beat. Splitting integrators offer the possibility of improving on Verlet, at least in some circumstances. Those integrators evaluate the forces more than once per step and, due to their simple kick-drift structure, may be implemented easily by modifying existing implementations of the Verlet scheme.

The Hamilton equations of motion, with the notations in (1), can be written as

$$\frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} = \nabla_{\mathbf{p}} A(\mathbf{q}, \mathbf{p}) = M^{-1}\mathbf{p}, \qquad \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = -\nabla_{\mathbf{q}} B(\mathbf{q}, \mathbf{p}) = -\nabla_{\mathbf{q}} U(\mathbf{q}).$$

These equations can be integrated in closed form and their solution flows at a time t are respectively given by

$$(\mathbf{q}(t), \mathbf{p}(t)) = \phi_t^A(\mathbf{q}(0), \mathbf{p}(0)), \qquad \mathbf{q}(t) = \mathbf{q}(0) + t\ M^{-1}\mathbf{p}(0), \qquad \mathbf{p}(t) = \mathbf{p}(0), \tag{4}$$

and

$$(\mathbf{q}(t), \mathbf{p}(t)) = \phi_t^B(\mathbf{q}(0), \mathbf{p}(0)), \qquad \mathbf{q}(t) = \mathbf{q}(0), \qquad \mathbf{p}(t) = \mathbf{p}(0) - t\ \nabla_{\mathbf{q}} U(\mathbf{q}(0)). \tag{5}$$

Here ϕ_t^A and ϕ_t^B denote the exact solution flows of the partial systems, i.e., the maps that associate the exact solution value $(\mathbf{q}(t), \mathbf{p}(t))$ with each initial condition $(\mathbf{q}(0), \mathbf{p}(0))$. Sometimes (4) might also be called a *drift* in the position and (5) a momentum *kick*.

Given a time step Δt, a velocity Verlet step corresponds to a transformation in phase space $(\mathbf{q}(t + \Delta t), \mathbf{p}(t + \Delta t)) = \psi_{\Delta t}(\mathbf{q}(t), \mathbf{p}(t))$ that can be written as

$$\psi_{\Delta t} = \phi^B_{\Delta t/2} \circ \phi^A_{\Delta t} \circ \phi^B_{\Delta t/2}.$$

In this thesis, for HMC methods, we study in detail mainly two-stage splitting integrators, which are the splitting schemes that perform two force evaluations per time step:

$$\psi_{\Delta t} = \phi^B_{b\Delta t} \circ \phi^A_{\Delta t/2} \circ \phi^B_{(1-2b)\Delta t} \circ \phi^A_{\Delta t/2} \circ \phi^B_{b\Delta t}, \tag{6}$$

where $b \in (0, 1/4]$ is a parameter of $\psi_{\Delta t}$. The two-stage integrators in (6) form a one-parameter family (Blanes, Casas, and Sanz-Serna, 2014). The value of a parameter for a two-stage integrator that results in a method leading to the smallest energy error was first identified by McLachlan, 1995 (ME integrator). On the other hand, Blanes, Casas, and Sanz-Serna, 2014 suggested choosing a parameter value so that a balance between good conservation of energy for reasonable values of Δt and accuracy for small Δt is achieved (BCSS integrator). The parameter values of (McLachlan, 1995; Blanes, Casas, and Sanz-Serna, 2014) do not vary with the problem being considered or with the value of Δt attempted by the user.

On the contrary, the method we propose here for two-stage integrators, and which we call Adaptive Integration Approach (AIA), automatically adjusts the parameter value for each problem and each choice of the time step Δt. Using the information on the highest frequencies of the harmonic interactions present in the system, AIA offers a system-specific integrator which guarantees the best energy conservation for harmonic forces achievable by an integrator from the family of two-stage splitting schemes, including Verlet, for any chosen Δt. While improvements in energy conservation do not necessarily imply dramatic changes in sampling, they improve acceptance rates in Hybrid Monte Carlo methods. The experiments performed in the present study also show that in both Hybrid Monte Carlo and molecular dynamics AIA leads to improvements of sampling as measured by the metrics considered. The improved sampling may arise as a consequence of either high acceptance rates (HMC) and enhanced accuracy (MD, HMC) with a given time step, or due to the possibility of using longer time steps (MD, HMC). On stability grounds, for any given problem, there is a maximum possible value of Δt; beyond this maximum all integrators in the family are unstable. When the time step chosen by the user is near the maximum value, AIA picks up an integrator that is (equivalent to) the standard Verlet scheme. As Δt decreases, AIA changes the integrator to ensure optimal conservation of energy; for Δt close to 0, AIA chooses McLachlan's scheme. In other words, the AIA approach successfully realizes the fail-safe strategy when the integrators are concerned. The AIA scheme can be implemented, without introducing computational overheads in simulations, in any software package which includes MD and/or HMC. In this study, we implement the AIA method in MultiHMC-GROMACS, a modified version of the popular GROMACS code (Berendsen, van der Spoel, and van Drunen, 1995; Hess et al., 2008), and test the new algorithm in HMC and MD simulations of unconstrained and constrained dynamics. The tests demonstrate the superiority of the novel scheme over Verlet, BCSS and some other advanced integration schemes, previously proposed in the literature. For a wide range of time steps and MD trajectory lengths, AIA outperforms other tested integrators in accuracy and sampling efficiency. The analysis of integrated autocorrelation functions (IACF) and folding evolution in the constrained benchmark demonstrates, for selected sizes of time steps, that AIA possesses up to 5 times better sampling performance than the other tested schemes.

We study the role of importance sampling on the performance of HMC through the modified Hamiltonian Monte Carlo (MHMC) methods. Such algorithms introduce the importance sampling in original HMC by sampling with respect to a modified or shadow Hamiltonian. Instead of sampling from the target canonical distribution (2), MHMC methods sample from

an auxiliary importance canonical density

$$\tilde{\pi}(\mathbf{q},\mathbf{p}) \propto \exp\left(-\beta \tilde{H}^{[k]}(\mathbf{q},\mathbf{p})\right). \tag{7}$$

Here $\tilde{H}^{[k]}$ denotes a truncated modified Hamiltonian. In this thesis, special attention is paid to the Generalized Shadow Hybrid Monte Carlo (GSHMC) method formulated by Akhmatskaya and Reich, 2008, and belonging to the MHMC class of samplers. The purpose of GSHMC was to enable sampling of large complex systems while retaining dynamical information. This is achieved by employing the modified energy for sampling and by partially updating momentum (cf. (3)). A modified Metropolis test is also introduced after the partial momentum update to preserve the desired modified density $\tilde{\pi}$.

As HMC and GHMC, the GSHMC method was first formulated in the canonical (NVT) ensemble. In this study, we discuss in detail the adaptation of GSHMC to the isobaric-isothermal (NPT) ensemble and propose the thorough analysis of its performance. The GSHMC method is adapted to the NPT ensemble using an Andersen barostat (Andersen, 1980) and we call the resulting algorithm NPT-GSHMC. It is implemented in the MultiHMC-GROMACS software package. The implementation is tested against the NPT-MD and NVT-GSHMC methods. NPT-GSHMC shows the same level of accuracy as demonstrated by NPT-MD and NVT-GSHMC in the calculation of the thermodynamic properties of the chosen benchmarks. The NPT-GSHMC method also proves to achieve a comparable sampling efficiency to NVT-GSHMC, as was expected from the theoretical formulation. The introduction of a barostat does not limit the benefits over MD that were previously obtained by the use of NVT-GSHMC. The method does not introduce any noticeable computational load. Thus, all advantages offered by the Generalized Shadow Hybrid Monte Carlo method, such as rigorous temperature control, sampling efficiency, are available in NPT-GSHMC and implemented in MultiHMC-GROMACS for simulation of real-life experiments at constant pressure and constant temperature without a loss of computational efficiency.

Furthermore, for the first time, we formulate GSHMC in the grand canonical (μVT) ensemble, and we call it GC-GSHMC. A general framework, useful for an extension of other Hybrid Monte Carlo methods to the grand canonical ensemble, is provided. Thus, the HMC and GHMC algorithms are also extended for the first time to the grand canonical ensemble. The validity of the three new methods has been proved in simulations of Lennard-Jones fluids at different conditions. All those methods reproduce well the predicted data (Nicolas et al., 1979; Johnson, Zollweg, and Gubbins, 1993). Also, the new algorithms sample up to 16 times better than the state-of-the-art MC algorithm by Yao, Greenkorn, and Chao, 1982. Among three new methods, GC-GSHMC shows the best accuracy and sampling efficiency. However, the proposed algorithms are only valid for homogeneous systems. Our future goal is to extend them to simple inhomogeneous systems and implement and test with rigid water models for the potential use in simulation of proteins in water.

To further improve the performance of importance sampling methods, we introduce new multi-stage integrators specifically derived for MHMC. The proposed two- and three-stage integration methods provide better conservation of modified Hamiltonians than does the Verlet integrator, commonly used in MHMC. Each of the derived methods is characterized by its coefficients, which are obtained from the minimization of the (expected with respect to a modified density (7)) error in modified Hamiltonians introduced by numerical integration. The new methods are tested and compared with Verlet and also with the sophisticated splitting integrators previously suggested for sampling with HMC.

For two-stage modified integrators, we also propose an adaptive integration approach which ultimately leads to enhancing the accuracy and sampling efficiency of MHMC methods. Given a simulation system and a user-chosen time step, the Modified Adaptive Integration Approach (MAIA) identifies by using information on the highest frequencies of the harmonic interactions present in the system the two-stage numerical integrator which, when used in the Hamiltonian dynamics step of an MHMC method, provides the best conservation of the modified Hamiltonian and thus the highest acceptance of the proposed trajectories. An enhanced variant of MAIA tailored to Generalized Shadow Hybrid Monte Carlo (GSHMC) methods is the extended MAIA (e-MAIA). It additionally supplies a value of the parameter φ that, for the problem under consideration, keeps the momentum acceptance at a user-desired level. The MAIA algorithm is implemented, with no computational overhead during simulations, in MultiHMC-GROMACS. The effect of the use of MAIA on the sampling efficiency of GSHMC is demonstrated in simulations with constrained atomistic and unconstrained coarse-grained benchmarks and compared with the performance of other suitable integration schemes, including velocity Verlet integrator. The tests reveal that the replacement in GSHMC of any fixed two-stage integrator with e-MAIA leads systematically to improvements in sampling efficiency of up to an order of magnitude. The performance comparison of GSHMC, HMC, and MD combined with their best choices of numerical integrators (e-MAIA, AIA, AIA, respectively) confirms the efficiency and robustness of the GSHMC-MAIA combination, whose advantages are especially noticeable when using the longest possible simulation time steps. For such cases, GSHMC, while maintaining good accuracy in simulation, provides a sampling efficiency (as measured with IACF) up to 30 times higher than the efficiency that may be achieved with MD.

The software development is another fundamental part of this work. The algorithms presented in this thesis are implemented in MultiHMC-GROMACS, the in-house version of the popular software package GROMACS. We explain the details of such implementation and give useful recommendations and hints for implementation of the new algorithms in other software packages. In addition, we supply the implementation details of some well-established methodologies which do not appear in the released version of GROMACS. The current structure of MultiHMC-GROMACS provides the flexibility for introducing different Hybrid Monte Carlo algorithms. Switching from one methodology to another is regulated by the values of input parameters. The MultiHMC-GROMACS code also offers a general framework for introducing new integrators and algorithms that can be expressed in a Trotter formulation (De Raedt and De Raedt, 1983). Two-, three- and four-stage integrators in original and modified formulations, and the adaptive integration schemes for HMC, MD and GSHMC (AIA, MAIA, e-MAIA) are implemented in MultiHMC-GROMACS. The two-stage integrators are also combined with the v-rescale (Bussi, Donadio, and Parrinello, 2007), Nosé-Hoover (Nosé, 1984b; Hoover, 1985), and MTTK (Martyna et al., 1996) thermostats and barostats. Since the multi-step algorithms can be easily expressed in the Trotter form, the current structure allows for a smooth implementation of this kind of methodologies.

In summary, in this thesis, we propose new numerical algorithms that are capable of improving the accuracy and sampling efficiency of molecular simulation using Hybrid Monte Carlo methods. We show that equipping the Hybrid Monte Carlo algorithm with extra features makes it a "smarter" sampler and a strong competitor to the well established molecular dynamics and Monte Carlo.

Resumen

El método Hybrid Monte Carlo (HMC) es una herramienta de sampleo que ofrece una propagación iteligente libre de errores de discretización en el espacio de fases, un control riguroso de la temperatura y flexibilidad. HMC apareció a finales de los ochenta en el contexto de las teorías de campos reticulados (Duane et al., 1987). Unos años después, HMC fue extendido a simulaciones moleculares (Heermann, Nielaba, y Rovere, 1990) y después a sistemas de materia condensada (Mehlig, Heermann, y Forrest, 1992). El método HMC pretende combinar las ventajas de la dinámica molecular (DM) y de los métodos de Monte Carlo (MC). DM permite aproximar la dinámica del sistema mientras que MC ayuda a explorar el espacio de fases de forma más global. De hecho, HMC es un algoritmo de tipo Metropolis-Hastings en el cual las propuestas se construyen utilizando el flujo hamiltoniano en el colectivo NVE del sistema. El objetivo de HMC es producir un sampleo eficiente en el colectivo canónico que, en última instancia, permita una aproximación precisa de medias de observables en el colectivo.

Consideramos hamiltonianos como

$$H(\mathbf{q},\mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + U(\mathbf{q}) \equiv A + B, \tag{1}$$

donde M es la matriz diagonal de masas y $\mathbf{q} \in \mathbb{R}^{3D}$, $\mathbf{p} \in \mathbb{R}^{3D}$ son las posiciones y momentos, respectivamente (D es la dimensión del sistema). Denotamos con $A = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}$ a la energía cinética y con $B = U(\mathbf{q})$ a la energía potencial.

Nos interesa samplear la variable $\mathbf{q} \in \mathbb{R}^{3D}$ que está distribuida de acuerdo con la probabilidad $\pi(\mathbf{q})$. La función de densidad de probabilidad (f.d.p.) objetivo se expresa como

$$\pi(\mathbf{q}) \propto \exp\left(-\beta U(\mathbf{q})\right).$$

El método HMC combina un movimiento global de DM con el sampleo de Monte Carlo del siguiente modo. Para cada iteración de Monte Carlo: (i) los momentos se resamplean siguiendo una distribución de Maxwell-Boltzmann $\rho_P(\mathbf{p})$; (ii) se genera un estado propuesto $(\mathbf{q}', \mathbf{p}')$ integrando las ecuaciones de movimiento con un integrador $\Psi_{\Delta t, L}$; (iii) se asegura la preservación de la distribución canónica deseada $\pi(\mathbf{q},\mathbf{p})$ mediante un test de Metropolis. Su probabilidad de aceptación se puede calcular como:

$$P_A\left((\mathbf{q},\mathbf{p}) \to \Psi_{\Delta t,L}(\mathbf{q},\mathbf{p})\right) = \min\left\{1, \exp\left(-\beta\Delta H\right)\right\},$$

donde

$$\Delta H = H\left(\Psi_{\Delta t,L}(\mathbf{q},\mathbf{p})\right) - H(\mathbf{q},\mathbf{p})$$

es el error de la energía asociado al integrador. Una f.d.p. conjunta $\pi(\mathbf{q},\mathbf{p})$ se define como

$$\pi(\mathbf{q},\mathbf{p}) = \pi(\mathbf{q})\rho_P(\mathbf{p}) \propto \exp\left(-\beta H(\mathbf{q},\mathbf{p})\right). \tag{2}$$

Por lo tanto, HMC puede ser entendido como un método que samplea puntos en el espacio de fases mediante una cadena de Markov en la cual transiciones estocásticas y dinámicas se

alternan.

El resampleo completo en (i) puede ser reemplazado por una actualización parcial del momento como fue propuesto en (Horowitz, 1991). Los momentos actuales se mezclan con un ruido gaussiano independiente e idénticamente distribuido $\mathbf{u} \sim \mathcal{N}(0, \beta^{-1}M)$ para obtener

$$\begin{aligned} \mathbf{p}^* &= \quad \cos\varphi\ \mathbf{p} + \sin\varphi\ \mathbf{u} \\ \mathbf{u}^* &= -\sin\varphi\ \mathbf{p} + \cos\varphi\ \mathbf{u}, \end{aligned} \tag{3}$$

donde $\varphi \in (0, \pi/2]$ controla la cantidad de ruido introducida. El ángulo φ también introduce un control extra en la eficiencia del sampleo del método y puede llevar a un rendimiento superior al de HMC. Esta idea fue formalizada en el método Generalized Hybrid Monte Carlo (GHMC) (Kennedy y Pendleton, 2001).

Pretendiendo ser una mejora de los métodos de Monte Carlo y de la dinámica molecular, Hybrid Monte Carlo resultó heredar dos desventajas desafortunadas. Como Monte Carlo, no genera información de la dinámica y su rendimiento se degrada al aumentar o el tamaño del sistema o el del paso. Por lo tanto, el objetivo de la tesis es la introducción de nuevos algoritmos para HMC que potencialmente pueden minimizar estas limitaciones. Para mejorar el rendimiento del método HMC, principalmente dos herramientas son consideradas: los integradores numéricos de división y la técnica del sampleo de importancia.

La eficiencia e incluso la viabilidad de las simulaciones de dinámica molecular depende crucialmente de la elección del integrador numérico. El papel de los integradores en mejorar la eficiencia de Hybrid Monte Carlo ha sido objeto de estudio activo en los últimos años (McLachlan, 1995; Blanes, Casas, y Sanz-Serna, 2014; Chao et al., 2015; Camos y Sanz-Serna, 2017; Bou-Rabee y Sanz-Serna, 2017). Actualmente, el algoritmo velocity Verlet es el método elegido por defecto; su simplicidad algorítmica y sus propiedades de estabilidad óptimas lo convierten en un método muy difícil de superar. Los integradores numéricos de división ofrecen la posibilidad de superar a Verlet, al menos en algunas circunstancias. Dichos integradores evalúan las fuerzas más de una vez por paso y, debido a su simple estructura kick-drift, pueden ser implementados fácilmente modificando implementaciones ya existentes del algoritmo de Verlet.

Las ecuaciones de movimiento de Hamilton, con la notación de (1), se pueden escribir como

$$\frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} = \nabla_{\mathbf{p}} A(\mathbf{q}, \mathbf{p}) = M^{-1}\mathbf{p}, \qquad \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = -\nabla_{\mathbf{q}} B(\mathbf{q}, \mathbf{p}) = -\nabla_{\mathbf{q}} U(\mathbf{q}).$$

Estas ecuaciones pueden ser integradas de forma analítica y sus flujos de solución en un tiempo t son respectivamente

$$(\mathbf{q}(t), \mathbf{p}(t)) = \phi_t^A(\mathbf{q}(0), \mathbf{p}(0)), \qquad \mathbf{q}(t) = \mathbf{q}(0) + t\ M^{-1}\mathbf{p}(0), \qquad \mathbf{p}(t) = \mathbf{p}(0), \tag{4}$$

y

$$(\mathbf{q}(t), \mathbf{p}(t)) = \phi_t^B(\mathbf{q}(0), \mathbf{p}(0)), \qquad \mathbf{q}(t) = \mathbf{q}(0), \qquad \mathbf{p}(t) = \mathbf{p}(0) - t\ \nabla_{\mathbf{q}} U(\mathbf{q}(0)). \tag{5}$$

Aquí ϕ_t^A y ϕ_t^B denotan el flujo de solución exacto de los sistemas parciales, i.e., las funciones que asocian el valor de la solución exacta $(\mathbf{q}(t), \mathbf{p}(t))$ con cada condición inicial $(\mathbf{q}(0), \mathbf{p}(0))$. A veces se puede llamar a (4) un *drift* en la posición y a (5) un *kick* en el momento.

Dado un paso Δt, un paso de velocity Verlet corresponde con una transformación en el espacio de fases $(\mathbf{q}(t + \Delta t), \mathbf{p}(t + \Delta t)) = \psi_{\Delta t}(\mathbf{q}(t), \mathbf{p}(t))$ que puede escribirse como

$$\psi_{\Delta t} = \phi^B_{\Delta t/2} \circ \phi^A_{\Delta t} \circ \phi^B_{\Delta t/2}.$$

En esta tesis, para métodos HMC, estudiamos en detalle principalmente integradores de dos etapas, los cuales son integradores de división que evalúan las fuerzas dos veces en cada paso:

$$\psi_{\Delta t} = \phi^B_{b\Delta t} \circ \phi^A_{\Delta t/2} \circ \phi^B_{(1-2b)\Delta t} \circ \phi^A_{\Delta t/2} \circ \phi^B_{b\Delta t}, \tag{6}$$

donde $b \in (0, 1/4]$ es un parámetro de $\psi_{\Delta t}$. Los integradores de dos etapas de (6) forman una familia que depende de un parámetro (Blanes, Casas, y Sanz-Serna, 2014). El valor de dicho parámetro que produce un método con el menor error de energía posible fue identificado por primera vez por McLachlan, 1995 (integrador ME). Por otra parte, Blanes, Casas, y Sanz-Serna, 2014 sugirieron escoger el parámetro b de tal manera que se satisfaciese un balance entre una buena conservación de la energía para valores razonables de Δt y precisión para Δt pequeños (integrador BCSS). Los parámetros de (McLachlan, 1995; Blanes, Casas, y Sanz-Serna, 2014) no varían con el problema que se considera o con el valor de Δt seleccionado por un usuario.

El método que proponemos aquí para integradores de dos etapas, y al que llamamos Adaptive Integration Approach (AIA), ajusta automáticamente el parámetro para cada problema y cada elección del paso Δt. Utilizando información de las frecuencias más altas dentro de las interacciones armónicas presentes en el sistema, AIA ofrece un integrador específico para cada sistema que garantiza la mejor conservación de energía posible para fuerzas armónicas que puede obtenerse con un integrador de dos etapas, incluído Verlet, para cualquier Δt elegido. Pese a que las mejores en la conservación de la energía no implican necesariamente cambios dramáticos en el sampleo, sí que mejoran las tasas de aceptación en los métodos Hybrid Monte Carlo. Los experimentos presentados en este estudio también muestran que tanto en Hybrid Monte Carlo como en dinámica molecular AIA mejora el sampleo para las métricas consideradas. Dicha mejora puede surgir como consecuencia o de las altas tasas de aceptación (HMC) y la mejora en la precisión (DM, HMC) para un paso dado, o debido a la posibilidad de utilizar pasos más largos (DM, HMC). En términos de la estabilidad, para cualquier problema estudiado, hay un valor máximo posible de Δt; más allá de dicho máximo todos los integradores son inestables. Cuando el paso escogido por el usuario es cercano a este máximo, AIA escoge un integrador que es equivalente al estándar Verlet. Cuando Δt disminuye, AIA cambia el integrador para asegurar una conservación de la energía óptima; para Δt cercano a 0, AIA escoge el método de McLachlan. Se puede implementar AIA sin introducir coste computacional adiccional en las simulaciones en cualquier paquete de software que incluya DM y/o HMC. En este estudio, AIA está implementado en MultiHMC-GROMACS, una versión modificada del popular paquete GROMACS (Berendsen, van der Spoel, y van Drunen, 1995; Hess et al., 2008), y comparamos el nuevo algoritmo con HMC y DM en simulaciones de sistemas con y sin *constraints*. Los experimentos demuestran la superioridad del nuevo método sobre Verlet, BCSS y algunos otros integradores presentes en la literatura. Para un amplio rango de pasos y trayectorias de DM, AIA supera al resto de integradores probados en precisión y eficiencia del sampleo. El análisis de funciones de autocorrelación integrada (IACF) y la evolución del pliegue de una proteína demuestran que, para los pasos seleccionados, AIA samplea hasta 5 veces mejor que el resto de métodos probados.

Estudiamos el papel que juega el sampleo de importancia en la eficiencia de HMC a través de los métodos modified Hamiltonian Monte Carlo (MHMC). Estos algoritmos introducen el sampleo de importancia en el HMC original sampleando con respecto a un hamiltoniano modificado o shadow. En lugar de samplear con respecto a la distribución canónica objetivo (2), los métodos MHMC samplean con respecto a una densidad canónica de importancia

auxiliar

$$\tilde{\pi}(\mathbf{q},\mathbf{p}) \propto \exp\Big(-\beta\tilde{H}^{[k]}(\mathbf{q},\mathbf{p})\Big). \tag{7}$$

Aquí $\tilde{H}^{[k]}$ denota un hamiltoniano modificado truncado. En esta tesis, prestamos especial atención al método Generalized Shadow Hybrid Monte Carlo (GSHMC) formulado por Akhmatskaya y Reich, 2008, y que pertenece a la clase de los sampleadores MHMC. El propósito de GSHMC fue permitir samplear sistemas grandes y complejos mientras se mantenía la información de la dinámica. Esto se consigue empleando la energía modificada para samplear y actualizando el momento parcialmente (cf. (3)). Un test de Metropolis modificado también se introduce después de la actualización parcial del momento para preservar la densidad modificada $\tilde{\pi}$.

Como HMC y GHMC, GSHMC inicialmente fue formulado en el colectivo canónico (NVT). En este estudio, discutimos en detalle la adaptación de GSHMC al colectivo isobárico-isotérmico (NPT) y proponemos el análisis detallado de su eficiencia. El método GSHMC se adapta al colectivo NPT utilizando el barostato de Andersen (Andersen, 1980) y llamamos al algoritmo resultante NPT-GSHMC. Está implementado en el paquete MultiHMC-GROMACS. La implementación se compara con los resultados de los métodos NPT-DM y NVT-GSHMC. NPT-GSHMC demuestra el mismo nivel de precisión que NPT-DM y NVT-GSHMC en el cálculo de las propiedades termodinámicas de los sistemas escogidos. El método NPT-GSHMC también muestra una eficiencia de sampleo comparable a la de NVT-GSHMC, como se puede esperar de su formulación teórica. La introducción de un barostato no limita los beneficios sobre DM que previamente fueron observados al utilizar NVT-GSHMC. El método no introduce coste computacional adiccional. Por tanto, todas las ventajas ofrecidas por el método Generalized Shadow Hybrid Monte Carlo, tales como el riguroso control de la temperatura o la eficiencia del sampleo, están disponibles también en NPT-GSHMC e implementadas en MultiHMC-GROMACS para poder simular experimentos de la vida real a temperatura y presión constantes.

Más aún, por primera vez formulamos GSHMC en el colectivo macrocanónico (μVT), y lo llamamos GC-GSHMC. Un marco general útil para extender otros métodos tipo Hybrid Monte Carlo al colectivo macrocanónico es propuesto. Así, HMC y GHMC también son extendidos por primera vez al colectivo macrocanónico. La validez de los tres nuevos métodos ha sido probada en simulaciones de fluidos tipo Lennard-Jones en distintas condiciones. Dichos métodos reproducen bien los datos predecidos previamente (Nicolas et al., 1979; Johnson, Zollweg, y Gubbins, 1993). Además, los nuevos algoritmos samplean hasta 16 veces mejor que un algoritmo MC de la literatura (Yao, Greenkorn, y Chao, 1982). Entre los tres nuevos métodos, GC-GSHMC muestra las mejores precisión y eficiencia de sampleo. Sin embargo, los algoritmos propuestos son solo válidos para sistemas homogéneos. Nuestro futuro objetivo es extenderlos a sencillos sistemas no homogéneos y testearlos con modelos rígidos de agua para su uso potencial en simulaciones de proteínas en agua.

Para mejorar aún más la eficiencia de los métodos de sampleo de importancia, introducimos integradores de varias etapas especialmente derivados para MHMC. Los integradores propuestos de dos y tres etapas proporcionan mejor conservación de los hamiltonianos modificados que Verlet, comúnmente usado en MHMC. Cada uno de los métodos derivados viene caracterizado por sus coeficientes, los cuales son obtenidos de la minimización del error (esperado con respecto a una densidad modificada (7)) en los hamiltonianos modificados introducido por los integradores numéricos. Los nuevos métodos son comparados con Verlet y con otros integradores sofisticados de varias etapas sugeridos para samplear con HMC.

Para los integradores modificados de dos etapas también proponemos un método adaptativo que mejora la precisión y la eficiencia del sampleo de los métodos MHMC. Dado un sistema y un paso elegido por un usuario, el Modified Adaptive Integration Approach (MAIA) identifica, utilizando información de las frecuencias más altas de las interacciones armónicas presentes en el sistema, el integrador de dos etapas que produce la mejor conservación del hamiltoniano modificado y, por tanto, las mayores tasas de aceptación de las trayectorias propuestas. Una versión mejorada de MAIA para el método Generalized Shadow Hybrid Monte Carlo (GSHMC) es el extended MAIA (e-MAIA). Adicionalmente proporciona un valor del parámetro φ que, para el problema considerado, mantiene la aceptación del momento a un nivel escogido por el usuario. MAIA está implementado, sin añadir costes computacionales en las simulaciones, en MultiHMC-GROMACS. El efecto del uso de MAIA en la eficiencia de sampleo de GSHMC está demostrado en simulaciones con sistemas con y sin *constraints* y comparado con la eficiencia de otros integradores disponibles, incluido velocity Verlet. Los experimentos revelan que reemplazar en GSHMC cualquier integrador de dos etapas por e-MAIA mejora sistemáticamente la eficiencia del sampleo. La comparación de GSHMC, HMC y DM combinada con sus mejores integradores (e-MAIA, AIA, AIA, respectivamente) confirma la eficiencia y robustez de GSHMC-MAIA, cuyas ventajas son más apreciables cuando se usan pasos grandes. Para este caso, GSHMC, mientras mantiene una buena precisión, proporciona una eficiencia de sampleo (medida con IACF) hasta 30 veces mejor que la proporcionada por DM.

El desarrollo de software es otra parte fundamental de este trabajo. Los algoritmos presentados en esta tesis están implementados en MultiHMC-GROMACS, el paquete de nuestro grupo basado en el popular GROMACS. Explicamos los detalles de dichas implementaciones y damos recomendaciones útiles para implementar los nuevos algoritmos en cualquier otro paquete. Además, proporcionamos los detalles de la implementación de algunos métodos clásicos que no aparecen en la versión original de GROMACS. La actual estructura de MultiHMC-GROMACS proporciona flexibilidad para introducir diferentes algoritmos tipo Hybrid Monte Carlo. Cambiar de un método a otro viene regulado por los parámetros dados como valores de entrada. Además MultiHMC-GROMACS ofrece un marco general para introducir nuevos integradores y algoritmos que puedan ser expresados en la formulación de Trotter (De Raedt y De Raedt, 1983). Los integradores de dos, tres y cuatro etapas en sus formulaciones original y modificada, así como los métodos adaptativos para HMC, DM y GSHMC (AIA, MAIA, e-MAIA) están implementados en MultiHMC-GROMACS. Los integradores de dos etapas además, pueden ser combinados con los termostatos y barostatos v-rescale (Bussi, Donadio, y Parrinello, 2007), Nosé-Hoover (Nosé, 1984b; Hoover, 1985), y MTTK (Martyna et al., 1996). Puesto que los métodos multi-paso pueden ser expresados fácilmente en una formulación de Trotter, la actual estructura del código permite su implementación directa.

En resumen, en esta tesis proponemos nuevos algoritmos numéricos que son capaces de mejorar la precisión y la eficiencia del sampleo de simulaciones moleculares utilizando métodos tipo Hybrid Monte Carlo methods. Demostramos que equipar al algoritmo Hybrid Monte Carlo con características adiccionales lo convierten en un sampleador "inteligente" y un serio competidor de los métodos clásicos, dinámica molecular y Monte Carlo, para simulaciones moleculares.

Acknowledgements

I am grateful to all the people that have shared these years with me. I want to thank them for the technical, scientific and personal support. I hope I have not forgotten anybody. In case I did it, my apologies.

I would like to start thanking my advisor Elena Akhmatskaya. I am truly grateful for her unconditional support, her accurate advice and her infinite patience. I believe this period has not been only about learning different things in science, which she mainly taught me, but also about growing as a person. Without her example, I would not be here.

I also want to thank Jesús María Sanz-Serna, the PI of the project where my PhD grant was held. It has been a real pleasure and a true honor to work with him. I am also very grateful for all the advice.

I want to thank Luis Vega who accepted to be my tutor in the University.

During three months of the PhD, I was visiting the group of António M. Baptista in the ITQB in Oeiras (Portugal). Some of the outcomes of that visit appear in the part of Chapter 5 devoted to the grand canonical ensemble. I was trained as a mathematician. Thus I started the PhD with the lack of knowledge of physical statistics and thermodynamics. I am very grateful for all the lessons I received in Portugal and for the personal experience. It is always good to have a great host.

Now that a stage is coming to an end, I do not want to forget the two people that guided me in my first days around science: Luis Miguel Pardo and Elias Fernández-Combarro. The lost kid needed some advice. Many thanks.

I believe that one of the most valuable aspects in science is working in a group, sharing ideas, discussing and learning from others. I had very good luck with the group where I worked these years. Especially with my scientific brothers. My older sister Tijana Radivojević has always been willing to give me advice, to help with anything and, more importantly, to let me know when I was wrong. That could be a definition of friendship. It has been a great pleasure to work with you, and I hope this is not a full stop. About my brother Simone Rusconi, I like telling people that at some point in our lives we shared an office, a flat and an advisor. Many thanks for not getting fed up with me and for the friendship.

Working in a place like BCAM is a great pleasure. The center is small (or it used to be), it is full of young people and the interactions are always very easy. This helped us to make some strong friendships with people from many different parts of the world. We got the experimental proof that countries mean very little when individuals are involved. I will never forget the assemblies in the kitchen and the evenings in Somera. It has been a huge pleasure to meet Umberto Biccari, Biagio Cassano, Carmen-Ana Domínguez, Daniel Eceizabarrena, Gorka Kobeaga, Dae-Jin Lee, Ariel Lozano, Josu Najera, Thomas Ourmières-Bonafos, Jorge Pérez, Fabio Pizzichillo, Antsa Ratsimanetrimanana, Stefano Scrobogna and Goran Stipcich. Eskerrik asko guztioi! Of course, I do not want to go to the next paragraph without a special mention to Julia Kroos. Your strength is an example.

The staff in BCAM is another very strong point. We are always certain that somebody will come and fix our problems. Many thanks to Miguel Benítez, Itziar Carro, Irantzu Elespe, Lorea Gómez, Ainara González, and especially to both Enekos, Anasagasti and Pérez.

I want to thank those who became my family in Bilbo outside of BCAM. I have always been lucky to live three hours away from home. But I have been as lucky to find a place in Bilbo where I could feel safe. It has been a big pleasure to meet my friends from Ferrol, Marcos Díaz and Eva Zamora. For all the valuable time we have spent together and the time we will spend, grazas irmáns. During these years, I have lived in different flats with many different people. If I have to choose, the best line-up I can pick is the one we made with Felipe Chaves and Sebas Banzas. Many thanks for your true friendship.

To all my friends from Asturies: united we stand, divided we fall, together we are what we can't be alone. The special mention is for my brothers from different mothers: Sergio Cobo and Fafa. Ye un honor.

Due to the amount of time I have spent in the office with the headphones on, I believe it is fair to thank also Ian MacKaye, Henry Rollins and Jello Biafra for screaming at a wall when I did not know how to solve a problem.

Los domingos tras un fin de semana en casa en Pravia siempre son de confusión de sentimientos. Tristeza y alegría. Tristeza por la distancia, pero alegría por la suerte que tengo. Suena a tópico, pero suerte de tener la mejor familia posible. Gratitud eterna a mis padres y a mi hermano por su apoyo incondicional. Sin ellos nada sería posible.

Azken lerroak, Irati, zuretzat dira. Ez dut hitzik nire esker ona adierazteko. Soilik jakin gure bidea elkarrekin egitea nire indarren iturria dela. Mila esker!

Mario
Bilbo, 2018

Contents

List of Algorithms

List of Figures

List of Tables

List of Abbreviations

ACF	**A**uto **C**orrelation **F**unction
AIA	**A**daptive **I**ntegration **A**pproach
BCSS	**B**lanes **C**asas **S**anz-**S**erna
e-MAIA	**E**xtended **M**odified **A**daptive **I**ntegration **A**pproach
EOS	**E**quation **O**f **S**tate
ESS	**E**ffective **S**ample **S**ize
GC	**G**rand **C**anonical
GHMC	**G**eneralized **H**ybrid **M**onte **C**arlo
GSHMC	**G**eneralized **S**hadow **H**ybrid **M**onte **C**arlo
HMC	**H**ybrid **M**onte **C**arlo
IACF	**I**ntegrated **A**uto **C**orrelation **F**unction
L2MC	**Second** order **L**angevin **M**onte **C**arlo
LD	**L**angevin **D**ynamics
MAIA	**M**odified **A**daptive **I**ntegration **A**pproach
MC	**M**onte **C**arlo
MCMC	**M**arkov **C**hain **M**onte **C**arlo
MD	**M**olecular **D**ynamics
MDMC	**M**olecular **D**ynamics **M**onte **C**arlo
ME	**M**inimum **E**rror
MHMC	**M**odified **H**ybrid **M**onte **C**arlo
MTS	**M**ultiple-**T**ime **S**tep
PMMC	**P**artial **M**omentum **M**onte **C**arlo
PMU	**P**artial **M**omentum **U**pdate
RG	**R**adius of **G**yration
RMSD	**R**oot **M**ean **S**quared **D**eviation
VV	**V**elocity **V**erlet

List of Symbols

q	position
v	velocity
M	mass matrix
p	momentum
D	size of the system
F	force
K	kinetic energy
U	potential energy
H	Hamiltonian
$\mathcal{L}$	Lagrangian
N	total number of samples
V	volume of the system
α	acceptance probability
β	thermodynamic beta
ω	angular frequency

A los mios güelos.

Chapter 1

Introduction to the Molecular Simulation of Complex Systems

1.1 Motivation

Molecular modeling relies on the methods, theoretical and computational, capable of modeling or mimicking the behavior of molecules. Such methods are used in different fields, for example, drug design, computational biology, nanoscience or materials science to study molecular systems ranging from small chemical systems to large biological molecules and material assemblies. Computers are required to perform molecular simulations, which can provide an *atomistic* level of description of molecular systems. Quantitative/qualitative information about the *macroscopic* behavior of macromolecules can be obtained from simulation of a system at an atomistic level. Due to the increase of the power of supercomputers, molecular simulation is becoming the technique of choice to describe systems of ever-increasing complexity, to discover new phenomena, or understand their structure, dynamics or thermodynamics. Molecular simulation constitutes a crossroads of mathematics, biology, chemistry, physics, and computer science.

The growth of molecular simulations popularity over the last decades would not have been possible without the rapid progress in computers technology. The improvement of CPUs made feasible simulations of large complex systems. Then, parallelization paradigms, such as OpenMP or MPI (Jost et al., 2003), as well as computers specialized in molecular simulation, e.g., MDGRAPE (Susukita et al., 2003) or Anton (Shaw et al., 2008), came to assist in the distribution of the most computationally or memory demanding parts of the simulations among processors, and thus opened new horizons for advanced simulations. In the last years, the introduction of GPUs helped to increase the speed of the calculations in molecular simulation further.

The revolution in hardware development requires new sophisticated algorithms utilizing the computer power. On the other hand, the complexity of molecular simulation dealing with real physical systems cannot be solved only using powerful computers. The efficient numerical algorithms are in high-demand. The development of simulation algorithms aiming to improve the sampling efficiency, while not suffering any loss of accuracy, is fundamental in the progress of molecular simulation abilities. The purpose of this thesis is to explore and develop new efficient numerical algorithms for molecular simulations.

In classical molecular simulation, a molecule is described as a series of charged points or atoms linked by springs or bonds. Thus, for each atom in every molecule, one needs to know the positions q, the momentum p, the charge θ and the bond information (which atoms, bond angles, etc.). Besides, the *potential energy* U helps to predict behavior and structure of systems since it describes the interaction energies of all atoms and molecules in the system.

However, it has to be remarked that U is always defined as an approximation. The closer to real physics it is, the more time is required to compute it. More precise interactions increase the accuracy of predictions of molecular simulations, but they also make the calculations computationally more demanding.

The main steps involved in a molecular simulation are (i) building a realistic atomistic model of the system to be studied; (ii) specifying simulation conditions (temperature, pressure, volume, etc.); (iii) simulation of the behavior of the system over time; (iv) analysis of the microscopic data obtained from a simulation and relating them to macroscopic properties. The conversion of microscopic information to macroscopic observables requires *statistical mechanics*. Average values are defined as *ensemble averages*, where an *ensemble* is a collection of all possible systems which have different *microscopic states* but posses an identical macroscopic state. The *ergodic hypothesis* states that, for any observable, the ensemble average and the time average coincide (Hill, 1960).

A key to success in molecular simulations is the proper *sampling* over accessible simulation times, since insufficient sampling is the greatest source of inaccuracy. Two big challenges can be outlined in quantitative simulations: complex structures of simulated systems and motions over different time scales. As an extreme example, we can mention the case of having fast bond vibrations and slow folding processes in a single simulation.

Two basic simulation approaches for sampling in molecular simulations are Monte Carlo and Molecular Dynamics.

- The *Monte Carlo* (MC) method is a stochastic approach. It generates a random walk of the system using a proposal density and provides a method for rejecting/accepting proposed moves (the *Metropolis test*). Discontinuous trajectories are produced and time has no clear meaning. Simulations do not offer kinetic information.

- The *Molecular Dynamics* (MD) method is a deterministic approach. The smooth trajectories are obtained by integrating *Hamilton's equations of motion* numerically. Time has a clear (physical) interpretation, and the simulations provide thermodynamic and kinetic properties.

1.2 Statistical ensembles

The concept of an ensemble was initially introduced by Gibbs at the beginning of the 20th Century. An ensemble is a collection of systems described by the same set of microscopic interactions and sharing a common set of macroscopic properties (e.g., the same total energy, volume, etc.). Each system evolves under the microscopic laws of motion from a different initial condition so that at any point in time, every system has a unique microscopic state. Once an ensemble is defined, macroscopic observables are calculated by performing averages over the systems in the ensemble. Ensembles can be defined for a wide variety of thermodynamic situations by choice of variables to be fixed during simulation. Their formulation usually comes from physical situations. The ensembles most commonly used in biomolecular simulations are the microcanonical (NVE), the canonical (NVT) and the isobaric-isothermal (NPT). In this section, we summarize these ensembles as well as the grand canonical ensemble (μVT), which, while being less popular due to its non-trivial implementation in MC and MD, will also be discussed and used later in this dissertation.

1.2.1 Microcanonical ensemble (NVE)

The microcanonical ensemble is composed of a collection of systems isolated from their surroundings. Each system in the ensemble is characterized by fixed values of the particle number N, volume V and total energy E. Thus, this ensemble is also called the NVE ensemble, as each of these three quantities is constant in the ensemble. Molecular dynamics naturally performs in the NVE ensemble.

The main disadvantage of the microcanonical ensemble is that a constant total energy is not a typical condition for laboratory experiments. Therefore, it is important to develop also ensembles that have different sets of thermodynamic control variables in order to reflect more common experimental setups.

1.2.2 Canonical ensemble (NVT)

The canonical ensemble is the statistical ensemble that represents the possible states of a system in thermal equilibrium with a heat bath at a fixed temperature T. The system can exchange energy with the heat bath so that the states of the system will differ in total energy. In the canonical or NVT ensemble, the number of particles N, the volume V and the temperature T are conserved.

In NVT, the total energy is exchanged with a *thermostat*, which controls the temperature. Monte Carlo serves as a thermostat itself, preserving temperature through the Metropolis test. In molecular dynamics, no temperature is present in the Hamilton's equations. A variety of thermostats are available to add and remove energy from the boundaries of an MD simulation. In the classical work by Andersen, 1980 an algorithm was suggested for simulations at a constant temperature for the first time. In this approach, the particles change their velocities by stochastic collisions which are chosen to reproduce the canonical ensemble. More thermostats appeared later such as Nosé-Hoover (Nosé, 1984a; Nosé, 1984b; Hoover, 1985), Berendsen (Berendsen et al., 1984), the velocity rescaling thermostat, or v-rescale, (Bussi, Donadio, and Parrinello, 2007).

1.2.3 Isobaric-isothermal ensemble (NPT)

In the isobaric-isothermal or NPT ensemble, the number of particles N, the pressure P and the temperature T are fixed. The NPT ensemble corresponds most closely to laboratory conditions. The volume of the system is allowed to fluctuate to maintain the pressure constant. Thus, an isobaric system can be viewed as coupled to an isotropic *piston* that compresses or expands the system uniformly in response to instantaneous pressure fluctuations such that the average internal pressure is maintained equal to an externally applied pressure.

In the NPT ensemble, in addition to a thermostat, a *barostat* is needed. The barostats are the algorithms that mimic the effect of the piston. Two main kinds of barostats are worth of mentioning here: those, which introduce an extended variable for the equations of motion (extended ensemble coupling) and those that use an external bath to perform the coupling (weak coupling). In the classical work by Andersen (Andersen, 1980), the first barostat for MD simulations was also proposed. The idea is that a simulated system is coupled to a fictitious "pressure bath" which mimics the action of an imaginary external piston linked to a constant reference pressure. The Parrinello-Rahman barostat (Parrinello and Rahman, 1981), the Nosé-Hoover barostat (Nosé, 1984a; Nosé, 1984b; Hoover, 1985) and the MTTK barostat (Martyna, Tobias, and Klein, 1994; Martyna et al., 1996; Tuckerman et al., 2006) are based on

the Andersen barostat. The most popular barostat that uses the external bathing approach is the Berendsen barostat (Berendsen et al., 1984).

1.2.4 Grand canonical ensemble (μVT)

The grand canonical ensemble is the statistical ensemble that represents the possible states of a system of particles maintained in thermodynamic equilibrium (thermal and chemical) with a reservoir. In the grand canonical or μVT ensemble, the chemical potential μ, the volume V and the temperature T are conserved. The system can exchange energy and particles with a reservoir, so that various possible states of the system can differ in both their total energy and a total number of particles. The system's volume, shape, and other external coordinates are kept the same in all possible states of the system. The grand canonical ensemble is appropriate for describing an open system such as, liquid-vapor equilibrium, capillary condensation, or molecular electronics and batteries, in which a device is assumed to be coupled with an electron source.

1.3 Simulation techniques

As it was mentioned in Section 1.1, there are two main sampling approaches which can be applied in molecular simulation, stochastic Monte Carlo (MC) and deterministic Molecular Dynamics (MD). The two methods are summarized below. The combination of both MC and MD leads to the *Hybrid Monte Carlo* (HMC) algorithm, which is also briefly explained here.

1.3.1 Monte Carlo (MC)

The first use of random methods to solve a physical problem dates back to the 30s when Enrico Fermi employed such an approach to study the properties of neutrons. Later, the method was formulated and named as Monte Carlo[1] (MC) by Nicholas Metropolis and Stanislaw Ulam (Metropolis and Ulam, 1949). Though it was intensively used in the late 40s in connection with the Manhattan Project, the first publication of an application of the method appeared in 1953 (Metropolis et al., 1953). Several types of Monte Carlo algorithms have been proposed in the literature (Kroese et al., 2014). In molecular simulation, the most common algorithm is *Metropolis-Hastings*, named after Wilfred K. Hastings (Hastings, 1970). It is a Markov Chain Monte Carlo (MCMC) method for obtaining a sequence of random samples from a probability distribution for which direct sampling is difficult. The method is designed to generate samples that make a large contribution to the distribution of interest. At each iteration, a new state is sampled from a proposal distribution, which depends on the current state, and either accepted or rejected according to the probability of the new sample relative to the current one.

In practice, MC neglects velocities and looks for minima on the potential energy surface by randomly probing configuration space. The sampling procedure is the following: (i) generate a random move; (ii) evaluate the potential energy U; (iii) accept/reject with Boltzmann probability

$$\min\left(1, \exp\left(-\beta \Delta U\right)\right), \tag{1.1}$$

where ΔU is the change in energy U and β is the inverse of the thermodynamic temperature $k_B T$ (k_B is the Boltzmann constant and T is the temperature). It has to be remarked that

[1] The name Monte Carlo came after conversations between Metropolis and Ulam in which the latter used to mention that his uncle "just had to go to Monte Carlo" to gamble (Metropolis, 1987).

the potential energy U only depends on the positions, which is consistent with the fact that dynamics are neglected in MC simulations. After a simulation, one can take averages in the NVT ensemble, since the temperature is prescribed and fixed via $\beta = 1/k_B T$. However, one advantage of the MC method is that it can be readily adapted to sampling in any ensemble, such as NPT (Wood, 1968) or μVT (Norman and Filinov, 1969).

In an MC simulation, the system behaves as a Markov process, meaning that the current state depends only on the previous one. The system is assumed to be ergodic. Therefore, any state can be reached from any other state, and time and ensemble averages are equivalent. However, the main drawback of the method in the context of molecular simulations is that, while being an exact method, MC does not offer a time evolution of the system. Moreover, since the moves are generated randomly, a usual procedure for avoiding big changes in the potential energy U (that would lead to a rejection) is to move only a few atoms at a time.

1.3.2 Molecular Dynamics (MD)

Molecular dynamics (MD) simulation is a computer approach to statistical mechanics used to estimate equilibrium and dynamic properties of complex systems by numerically solving Newton's equations. The molecular dynamics method was also originally developed within the field of theoretical physics in the late 50s. There are three seminal works which are considered as the foundations of MD: (Fermi, Pasta, and Ulam, 1955)[2]; (Alder and Wainwright, 1959); and (Rahman, 1964). Inspired by the success of the Monte Carlo simulations (Metropolis et al., 1953), Fermi, Pasta, Ulam and Tsingou first suggested MD in the mid-50s. It was then formulated independently by Alder and Wainwright in the late 50s and Rahman in the 60s. In 1957, Alder and Wainwright used an IBM 704 computer to simulate elastic collisions between hard spheres. In 1964, Rahman published landmark simulations of liquid argon that used a Lennard-Jones potential. Calculations of system properties, such as the coefficient of self-diffusion, compared well with experimental data. A few years later, Verlet performed simulations for the same system as Rahman using for the first time the famous integrator named after him (Verlet, 1967).

The formal structure of MD is the Hamiltonian mechanics, where the forces are *conservative*:

$$F(\mathbf{q}) = -\nabla_{\mathbf{q}} U(\mathbf{q}).$$

A large class of many-particle systems can be described by a classical separable *Hamiltonian* of the form

$$H(\mathbf{q}, \mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + U(\mathbf{q}), \tag{1.2}$$

in which we denote the *kinetic energy* as

$$K(\mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}.$$

Here M is the diagonal mass matrix, and $\mathbf{q} \in \mathbb{R}^{3D}$, $\mathbf{p} \in \mathbb{R}^{3D}$ are the positions and momenta, respectively. D is a system's dimension.

[2] Although Mary Tsingou did not appear as an author of the original paper, she was mentioned in an acknowledgement for her work in programming the MANIAC simulations. Mary Tsingou's contributions to the Fermi-Pasta-Ulam-Tsingou problem were largely ignored by the community until Dauxois, 2008 published additional information regarding the development and called for the problem to be renamed to grant proper attribution.

The *Hamilton's equations of motion* that drive the dynamics of the system read as

$$\frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} = \frac{\partial H}{\partial \mathbf{p}} = M^{-1}\mathbf{p}, \qquad \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = -\frac{\partial H}{\partial \mathbf{q}} = F(\mathbf{q}), \tag{1.3}$$

which are in agreement with Newton's laws of motion. Hamilton's equations conserve the total Hamiltonian H. The equations of motion (1.3) are solved using *numerical integrators* since analytical (closed-form) solutions are only known for very simple systems. The numerical integration generates a sequence of positions and momenta pairs $(\mathbf{q}^i, \mathbf{p}^i)$ for integers i that represent discrete times $t = i\Delta t$ at intervals or time steps Δt. Numerical integrators are only required to be *time-reversible* and *symplectic*.

The most computationally demanding part of an MD simulation is the calculation of the forces. A usual procedure is to increase the time step Δt to perform fewer integration steps and then calculate the forces less frequently. There are several techniques for increasing the time steps. For instance, it is possible to solve the equations of motion under the *constraints* so that the rigid bonds and/or bond angles do not change during a simulation (Ryckaert, Ciccotti, and Berendsen, 1977). The motion associated with the remaining degrees of freedom is slower, and thus it helps to use bigger time steps. Different multiple-time step (MTS) techniques and coarse-graining approaches having the same purpose of decreasing a frequency of the force evaluations are available in the literature (Tuckerman, Berne, and Martyna, 1992; Marrink et al., 2007).

As stated before, the NVE or microcanonical ensemble (see Section 1.2.1) consists of all microscopic states on the constant energy hypersurface $H(\mathbf{q}, \mathbf{p}) = E$. Since the equations of motion (1.3) preserve the Hamiltonian, a trajectory computed with such equations generates microscopic configurations in the microcanonical ensemble. Thus, by construction, the temperature T is not maintained. The main drawbacks of MD are that (i) it is not free of discretization errors; (ii) it does not allow for large moves between consecutive configurations (the numerical time step has to be small to ensure energy conservation); and (iii) it does not provide rigorous temperature control in a simulation.

1.3.3 Hybrid Monte Carlo (HMC)

In contrast to molecular dynamics, Monte Carlo methods generate canonical distribution and do not introduce discretization errors. However, such methods can only attempt to move a few particles at a time in order to maintain a reasonable average acceptance rate. Thus, another difference between MC and molecular dynamics is in the ability of the latter to generate moves of the entire system. Nevertheless, such moves are deterministic and fundamentally limited by the time step, which has to be sufficiently small to ensure energy conservation. Clearly, that Monte Carlo and molecular dynamics are surprisingly complementary: where one method fails another succeeds. The Hybrid Monte Carlo (HMC) algorithm came as an attempt to combine advantages of the two methods.

The Hybrid Monte Carlo method was originally formulated by Duane et al., 1987 to study lattice models of quantum field theory. In the recent years, HMC also became popular in computational statistics, known under the name of Hamiltonian Monte Carlo (Neal, 2011).

The HMC algorithm consists of short MD trajectories (in the NVE ensemble) which are accepted/rejected according to the Metropolis test with a probability

$$\min\left(1, \exp\left(-\beta \Delta H\right)\right), \tag{1.4}$$

where ΔH is not zero due to the error introduced by the numerical integrator. The acceptance rule (1.4) differs from (1.1) in the presence of the kinetic energy. The Metropolis test in HMC is followed by a full reset of momenta in order to start a new MD trajectory. Thus, dynamical properties of a simulated system cannot be properly reproduced using HMC. Originally, HMC samples in the NVT ensemble, but it can be extended to other statistical ensembles (Faller and de Pablo, 2002).

The HMC algorithm can be summarized as a combination of Hamiltonian dynamics (MD) and a Metropolis-Hastings acceptance rule (MC). It offers the possibility of generating proposals that, while being far away from the current state of the Markov chain, may be accepted with high probability. Thus, it avoids random walk behavior, and it reduces the correlation between samples. HMC can be understood either as an efficient MC with a "smart" collective move or as a thermodynamically consistent MD with corrupted dynamics.

1.4 Objectives of the thesis

Meaning to be an improvement of both Monte Carlo and molecular dynamics, Hybrid Monte Carlo turned out to inherit two unfortunate drawbacks. It does not generate dynamic information, and its performance degrades with an increase of either the system size or the time step. The goal of this thesis is to introduce the new algorithms for HMC which can potentially minimize these limitations.

In the following chapters of this dissertation, in order to enhance the performance of the HMC method, two main tools will be considered: the splitting numerical integrators and the importance sampling technique.

Splitting integrators are more sophisticated integration schemes than the commonly used in molecular simulation Verlet/leapfrog integrator. The development of such integrators may lead to very promising improvements in accuracy and sampling in MD and HMC as has been demonstrated in (McLachlan, 1995; Blanes, Casas, and Sanz-Serna, 2014). In this thesis, we develop various novel splitting integrators and analyze their accuracy and effect on sampling performance of HMC in comparison with the methods available in the literature.

In this work, the role of importance sampling on the performance of HMC is studied through the modified Hamiltonian Monte Carlo (MHMC) methods. Such algorithms introduce the importance sampling in original HMC by sampling with respect to a modified or shadow Hamiltonian. Special attention is paid to the Generalized Shadow Hybrid Monte Carlo (GSHMC) method formulated by Akhmatskaya and Reich, 2008. As Hybrid Monte Carlo, the GSHMC method was first formulated in the NVT ensemble, though, the hints about its extension to the NPT ensemble were also provided in the original paper (Akhmatskaya and Reich, 2008). In this study, we discuss in detail the adaptation of GSHMC to the NPT ensemble and propose the thorough analysis of its performance. Moreover, for the first time, we formulate GSHMC in the grand canonical or μVT ensemble. A general framework, useful for an extension of other Hybrid Monte Carlo methods to the grand canonical ensemble, is also provided.

The software development is another fundamental part of the present work. The algorithms presented in this thesis are implemented in MultiHMC-GROMACS, an in-house version of the popular software package GROMACS. We explain the details of such implementation and give useful recommendations and hints for implementation of the new algorithms in other software packages.

In summary, in this thesis, we propose new numerical algorithms that are capable of improving the accuracy and sampling efficiency of molecular simulation using Hybrid Monte Carlo methods. We show that equipping the Hybrid Monte Carlo algorithm with extra features makes it a "smarter" sampler and a strong competitor to the well established molecular dynamics and Monte Carlo.

The structure of the present document is as follows. Chapter 2 presents Hybrid Monte Carlo in detail and provides the useful references to the literature. In Chapter 3, following a summary of state-of-the-art numerical integrators, the adaptive two-stage integration algorithm is derived and illustrated with numerical experiments. In Chapter 4, the importance sampling Hybrid Monte Carlo algorithms are introduced. A particular emphasis is put on the Generalized Shadow Hybrid Monte Carlo method. In Chapter 5, we extend GSHMC to simulations in two statistical ensembles, NPT and μVT, introducing the new sampling methods, NVT-GSHMC and GC-GSHMC. HMC and GHMC in the μVT ensemble are also presented. The new algorithms are tested and compared in accuracy and performance. In Chapter 6 the integrators specially formulated for sampling with respect to modified Hamiltonians are derived and tested. The novel adaptive two-stage integration approach specifically derived for modified Hamiltonian Monte Carlo is presented. In Chapter 7, the implementation of new algorithms in the MultiHMC-GROMACS package, as well as the structure and novel features of the software package, are discussed. The conclusions and some ideas for future work are summarized in Chapter 8.

Chapter 2

Hybrid Monte Carlo Methods

2.1 Formulation of Hybrid Monte Carlo

The Hybrid Monte Carlo (HMC) method appeared in the late eighties in the context of lattice field theories. The original paper in which the method was first formulated is (Duane et al., 1987). A few years later, the HMC algorithm was extended to molecular simulations (Heermann, Nielaba, and Rovere, 1990) and then to condensed-matter systems (Mehlig, Heermann, and Forrest, 1992). The HMC method aims at combining the advantages of the molecular dynamics (MD) and Monte Carlo (MC) methods. MD allows for approximating the physical dynamics of the system while MC helps to explore the phase space more globally. In fact, HMC is a Metropolis-Hastings algorithm in which proposals are constructed using the NVE Hamiltonian flow of the system.

The goal of HMC is to perform an efficient sampling in the canonical ensemble which ultimately allows for an accurate estimation of ensemble averages.

Considering Hamiltonians $H(\mathbf{q},\mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + U(\mathbf{q})$ as in (1.2) and given a system of size D, the canonical ensemble average of an observable $\Omega(\mathbf{q},\mathbf{p})$ is

$$\langle \Omega(\mathbf{q},\mathbf{p})\rangle_{NVT} = \frac{1}{Q}\frac{1}{D!h^{3D}}\int \Omega(\mathbf{q},\mathbf{p})e^{-\beta H(\mathbf{q},\mathbf{p})}\mathrm{d}\mathbf{q}\ \mathrm{d}\mathbf{p},$$

where h is Planck constant, β is the inverse of the thermodynamic temperature $k_B T$, and Q is the canonical partition function denoted by (cf. (Hill, 1960))

$$Q = \frac{1}{D!h^{3D}}\int e^{-\beta H(\mathbf{q},\mathbf{p})}\mathrm{d}\mathbf{q}\ \mathrm{d}\mathbf{p}.$$

One can compute the Gaussian integral over the momenta and obtain

$$Q = \frac{1}{D!\Lambda^{3D}}\mathcal{Z}, \tag{2.1}$$

where Λ is the thermal de Broglie wavelength, which is roughly the distance that molecules can approach before quantum effects become significant, and the *configurational integral* $\mathcal{Z}$ is defined as

$$\mathcal{Z} = \int e^{-\beta U(\mathbf{q})}\mathrm{d}\mathbf{q}.$$

We are interested in sampling the variable $\mathbf{q} \in \mathbb{R}^{3D}$ that is distributed according to the probability $\pi(\mathbf{q})$. The target probability density function (p.d.f.) is written as

$$\pi(\mathbf{q}) = \frac{1}{\mathcal{Z}}e^{-\beta U(\mathbf{q})}, \tag{2.2}$$

where $U(\mathbf{q})$ denotes the potential energy.

Assuming that one wants to study an observable $\Omega(\mathbf{q})$, which is a function of the positions only, its ensemble average is given by

$$\langle \Omega(\mathbf{q}) \rangle = \frac{1}{\mathcal{Z}} \int \Omega(\mathbf{q}) e^{-\beta U(\mathbf{q})} \mathrm{d}\mathbf{q}.$$

Given a sequence of N configurations $\mathbf{q}^n$ distributed according to (2.2), the law of large numbers implies that the ensemble average of the observable $\Omega(\mathbf{q})$ can be computed as

$$\langle \Omega(\mathbf{q}) \rangle = \lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} \Omega(\mathbf{q}^n).$$

In a Monte Carlo calculation the configurations are generated as a Markov chain defined by a conditional transition probability density $\rho_T(\mathbf{q}^n \to \mathbf{q}^{n+1})$. Provided the Markov chain is irreducible and aperiodic (i.e., ergodic if it is a finite state chain), the *detailed balance* condition

$$\pi(\mathbf{q}^n)\rho_T(\mathbf{q}^n \to \mathbf{q}^{n+1}) = \pi(\mathbf{q}^{n+1})\rho_T(\mathbf{q}^{n+1} \to \mathbf{q}^n)$$

ensures that the Markov chain converges to the unique stationary probability distribution (2.2). The detailed balance is a sufficient but not necessary condition, whereas the stationarity is.

In practice, one step $\mathbf{q}^n \to \mathbf{q}^{n+1}$ in the Markov chain of configurations is realized by proposing $\mathbf{q}^{n+1}$ according to a proposal probability density $\rho_P(\mathbf{q}^n \to \mathbf{q}^{n+1})$ and accepting it with the probability

$$P_A(\mathbf{q}^n \to \mathbf{q}^{n+1}) = \min\left\{1, \frac{\pi(\mathbf{q}^{n+1})\rho_P(\mathbf{q}^{n+1} \to \mathbf{q}^n)}{\pi(\mathbf{q}^n)\rho_P(\mathbf{q}^n \to \mathbf{q}^{n+1})}\right\}, \tag{2.3}$$

which is the *Metropolis-Hastings* function (Metropolis et al., 1953; Hastings, 1970). Thus, the conditional probability densities $\rho_T(\mathbf{q}^n \to \mathbf{q}^{n+1})$ are given by

$$\rho_T(\mathbf{q}^n \to \mathbf{q}^{n+1}) = \rho_P(\mathbf{q}^n \to \mathbf{q}^{n+1}) P_A(\mathbf{q}^n \to \mathbf{q}^{n+1}).$$

In conventional MC simulations only local moves are performed, i.e., single particle updates. This may lead to slow exploration of *phase space*. However, updating more than one particle at a time typically results in a very low acceptance rate, which implies large relaxation times and high autocorrelations. Replacing a local move with a global one, as is performed in MD, may in principle improve sampling provided the high acceptance rate. A global move can be described as follows. Given a time step Δt and a number of steps L, integrate the equations of motion of the system (1.3) through phase space for a time $t = \Delta t \cdot L$ using a chosen *discretization scheme* or *integrator*

$$\begin{aligned} \Psi_{\Delta t, L} \colon \mathbb{R}^{6D} &\to \mathbb{R}^{6D}. \\ (\mathbf{q}, \mathbf{p}) &\mapsto (\mathbf{q}', \mathbf{p}') \end{aligned} \tag{2.4}$$

The system is moved deterministically through phase space, and then the conditional probability of proposing a new set of coordinates $\mathbf{q}'$ starting at $\mathbf{q}$ is given by

$$\rho_P(\mathbf{q} \to \mathbf{q}') = \rho_P(\mathbf{p}), \tag{2.5}$$

i.e., the proposal probability depends only on the momenta. The initial momenta for each new global move are drawn from the Maxwell-Boltzmann distribution:

$$\rho_P(\mathbf{p}) \propto \exp\left(-\beta \sum_{i=1}^{D} \frac{p_i^2}{2m_i}\right), \tag{2.6}$$

where m_i are the masses and D is the size of the system.

The HMC method combines an MD global move with Monte Carlo sampling in the following way. For each Monte Carlo iteration: (i) the momenta are resampled from (2.6); (ii) a proposed new state $(\mathbf{q}', \mathbf{p}')$ is generated by integrating the equations of motion with an integrator $\Psi_{\Delta t,L}$ (2.4); (iii) the preservation of the desired distribution $\pi(\mathbf{q}, \mathbf{p})$ is ensured by a Metropolis test. Its acceptance probability can be calculated by combining (2.2), (2.3) and (2.6):

$$P_A\left((\mathbf{q}, \mathbf{p}) \to \Psi_{\Delta t,L}(\mathbf{q}, \mathbf{p})\right) = \min\left\{1, \exp\left(-\beta \Delta H\right)\right\}, \tag{2.7}$$

where

$$\Delta H = H\left(\Psi_{\Delta t,L}(\mathbf{q}, \mathbf{p})\right) - H(\mathbf{q}, \mathbf{p}) \tag{2.8}$$

is the *energy error* associated to the integration scheme. For the sake of simplicity we will denote the acceptance probability in (2.7) as α. Clearly, a joint p.d.f. $\pi(\mathbf{q}, \mathbf{p})$ is defined as

$$\pi(\mathbf{q}, \mathbf{p}) = \pi(\mathbf{q})\rho_P(\mathbf{p}) = \frac{1}{Q} \exp\left(-\beta H(\mathbf{q}, \mathbf{p})\right). \tag{2.9}$$

Therefore, HMC can be viewed as a method that samples points in phase space by means of a Markov Chain in which stochastic and dynamical transitions alternate.

The HMC algorithm can be summarized as follows:

Algorithm 1 Hybrid Monte Carlo

Input: Δt: time step
L: number of integration steps
Ψ: discretization scheme
N: number of MC iterations
T: temperature

1: initialize $\mathbf{q}^0$
2: **for** $n = 1, \ldots, N$ **do**
3: $\mathbf{q} = \mathbf{q}^{n-1}$
4: draw momenta $\mathbf{p}$ from Maxwell-Boltzmann distribution (2.6)
5: generate a proposal by integrating Hamiltonian dynamics

$$(\mathbf{q}', \mathbf{p}') = \Psi_{\Delta t,L}(\mathbf{q}, \mathbf{p})$$

6: calculate the acceptance probability

$$\alpha = \min\left\{1, \exp\left(-\beta\left(H(\mathbf{q}', \mathbf{p}') - H(\mathbf{q}, \mathbf{p})\right)\right)\right\}$$

7: Metropolis test
draw $u \sim \mathcal{U}(0, 1)$
if $u < \alpha$
accept: $\mathbf{q}^n = \mathbf{q}'$

else
reject: $\mathbf{q}^n = \mathbf{q}$
end if
discard momenta $\mathbf{p}'$, $\mathbf{p}$
end for

It can be shown that for the HMC algorithm the detailed balance condition is satisfied for a *time-reversible* discretization scheme Ψ

$$\Psi_{-\Delta t} \circ \Psi_{\Delta t} = I \tag{2.10}$$

and *volume preserving*

$$\det \frac{\partial \Psi_{\Delta t}(\mathbf{q}, \mathbf{p})}{\partial(\mathbf{q}, \mathbf{p})} = 1. \tag{2.11}$$

We need to recall the following algebraic identity:

$$\exp\left(-\beta H(\mathbf{q}, \mathbf{p})\right) \min\left\{1, \exp\left(-\beta \Delta H\right)\right\} = \exp\left(-\beta H\left(\Psi_{\Delta t}(\mathbf{q}, \mathbf{p})\right)\right) \min\left\{\exp\left(\beta \Delta H\right), 1\right\}.$$

Then, we can proof detailed balance as follows:

$$\begin{aligned}
\pi(\mathbf{q})\ \rho_T(\mathbf{q} \to \mathbf{q}') &= \pi(\mathbf{q})\ \rho_P(\mathbf{p})\ P_A((\mathbf{q}, \mathbf{p}) \to \Psi_{\Delta t}(\mathbf{q}, \mathbf{p})) \\
&= \pi(\mathbf{q}')\ \rho_P(\mathbf{p}')\ P_A(\Psi_{\Delta t}(\mathbf{q}, \mathbf{p}) \to (\mathbf{q}, \mathbf{p})) \\
&= \pi(\mathbf{q}')\ \rho_P(\mathbf{p}')\ P_A((\mathbf{q}', \mathbf{p}') \to \Psi_{-\Delta t}(\mathbf{q}', \mathbf{p}')) \\
&= \pi(\mathbf{q}')\ \rho_T(\mathbf{q}' \to \mathbf{q}).
\end{aligned}$$

Therefore, provided a chosen integrator is time-reversible and volume preserving, the HMC algorithm generates a Markov chain with the stationary probability distribution $\pi(\mathbf{q}, \mathbf{p})$ in (2.9) (it is easy to notice that (2.2) is just a marginalization of (2.9)). From now on, we will consider *symplectic* and time-reversible integrators. Symplecticity is a more general condition than volume preservation and it will play a fundamental role in the following chapters. A definition will be provided in Chapter 3. An informative review on symplecticity and its importance in dynamical systems can be found in (Sanz-Serna and Calvo, 1994) and several different symplectic discretization schemes are surveyed in (Sanz-Serna, 1992).

Neither $\pi(\mathbf{q})$ nor any ensemble averages depend on the time step chosen. However, the average acceptance probability depends on the expected discretization error $\mathbb{E}(\Delta H)$ (cf. (2.7)) and hence it does depend on the time step. The useful relation between the average acceptance probability $\langle P_A \rangle$ and the expected energy error $\mathbb{E}(\Delta H)$ for sufficiently large systems has been proposed (cf. (Gupta et al., 1990)):

$$\langle P_A \rangle = \operatorname{ercf}\left(\frac{1}{2}\sqrt{\mathbb{E}(\Delta H)}\right), \tag{2.12}$$

where ercf is the complementary error function. The work by Gupta et al., 1990 was an extension of the analysis of Creutz, 1988 for the large volume limit. Useful discussions on how the acceptance probability changes with the size of the system D can be found in the cited (Creutz, 1988) and also in (Gupta, Kilcup, and Sharpe, 1988). As a curiosity, the two papers appeared the same year in the same issue of the journal and the similar results are proved in different manners. There are two important relations: (i) $\Delta t \propto D^{-1/4}$ to maintain a

reasonable acceptance rate; and (ii) the cost of HMC per independent sample from the target distribution is $\mathcal{O}(D^{5/4})$, which stands in contrast with the $\mathcal{O}(D^2)$ cost of Metropolis.

It is remarked by Mehlig, Heermann, and Forrest, 1992 that it may be possible to formulate more general algorithms, which do not obey detailed balance condition but satisfy the relation

$$\int \pi(\mathbf{q})\ \rho_T(\mathbf{q} \to \mathbf{q}')\ \mathrm{d}\mathbf{q} = \pi(\mathbf{q}'),$$

which is a necessary and sufficient condition for the Markov chain to have the stationary probability distribution. In (Fang, Sanz-Serna, and Skeel, 2014) there is a significant weakening of sufficient conditions for stationarity: preservation of volume in phase space is not required and reversibility of a discretization scheme is needed only in the form of a bijection rather than an involution. The volume-preserving property of the integration schemes can be relaxed by including a Jacobian factor in the calculation of the Metropolis acceptance probability as explained in (Leimkuhler and Reich, 2009).

Useful explanations and theoretical results on the convergence can be found in the review by Cancès, Legoll, and Stoltz, 2007.

There are statistical results and some hints about the optimal tuning of the algorithm in (Beskos et al., 2013). The authors identified the value of 0.651 as an optimal acceptance rate for distributions with independent and identically distributed variates and the Verlet integrator. This result was extended to general distributions and symplectic integrators in (Betancourt, Byrne, and Girolami, 2014) with the optimal interval for average acceptance rate being between 0.6 and 0.9.

Several variations of HMC have been proposed. We will discuss them later in Chapter 4. For now, we will focus on the generalization of HMC in the method called Generalized Hybrid Monte Carlo method (Kennedy and Pendleton, 2001). This algorithm will play a central role in the following chapters.

2.2 Generalized Hybrid Monte Carlo (GHMC)

One of the drawbacks of Hybrid Monte Carlos is its inability to, in contrast to molecular dynamics, predict dynamics of a simulated system. This can be partially overcome by generating, after each NVE trajectory of length L, some new momenta which are correlated with the previous ones. The partial momentum update (in contrast to the complete momentum update) was introduced by Horowitz, 1991 within Generalized guided Monte Carlo, a method that relies on a single step of Hamiltonian dynamics. This method is also known as a second-order Langevin Monte Carlo (L2MC). The purpose of this technique was to retain more dynamical information of the simulated system.

In (Kennedy and Pendleton, 2001) this idea was formalized in the Generalized Hybrid Monte Carlo (GHMC) method. GHMC is defined as the concatenation of two steps: Molecular Dynamics Monte Carlo (MDMC) and Partial Momentum Update (PMU).

The GHMC method only differs from HMC in the momentum update step. The MDMC is defined in the same way as in the HMC method. However, whereas in HMC the momenta are completely reset for initiating a new trajectory, in GHMC, the momenta are partially updated. The current momenta are mixed with an independent and identically distributed

(i.i.d.) Gaussian noise $\mathbf{u} \sim \mathcal{N}(0, \beta^{-1}M)$ to obtain

$$\begin{aligned} \mathbf{p}^* &= \cos\varphi\ \mathbf{p} + \sin\varphi\ \mathbf{u} \\ \mathbf{u}^* &= -\sin\varphi\ \mathbf{p} + \cos\varphi\ \mathbf{u}, \end{aligned} \tag{2.13}$$

where $\varphi \in (0, \pi/2]$ controls the amount of noise introduced. The angle φ also introduces extra control over the sampling efficiency of the method and may lead to the superior performance of GHMC over HMC. It updates the momentum between trajectories partially so that consecutive trajectories tend to move in more similar directions.

It has to be remarked that there is no need for a Metropolis test after the orthogonal transformation in (2.13) since it preserves for $\mathbf{p}^*$ and $\mathbf{u}^*$ the distributions of $\mathbf{p}$ and $\mathbf{u}$. However, since momenta are not discarded, the method incorporates a momentum flip

$$\mathcal{F}(\mathbf{q}, \mathbf{p}) = (\mathbf{q}, -\mathbf{p}) \tag{2.14}$$

upon rejection, that ensures that the detailed balance condition is satisfied. The $\mathbf{u}^*$ generated in (2.13) are always discarded.

The GHMC algorithm can be summarized as follows:

Algorithm 2 Generalized Hybrid Monte Carlo

Input: M: mass matrix
Δt: time step
L: number of integration steps
Ψ: discretization scheme
N: number of MC iterations
T: temperature
$\varphi \in (0, \pi/2]$: noise angle

1: initialize $(\mathbf{q}^0, \mathbf{p}^0)$
2: **for** $n = 1, \ldots, N$ **do**
3: $(\mathbf{q}, \mathbf{p}) = (\mathbf{q}^{n-1}, \mathbf{p}^{n-1})$
4: partial momentum update

$$\begin{aligned} \mathbf{p}^* &= \cos\varphi\ \mathbf{p} + \sin\varphi\ \mathbf{u} \\ \mathbf{u}^* &= -\sin\varphi\ \mathbf{p} + \cos\varphi\ \mathbf{u} \end{aligned}$$

where $\mathbf{u} \sim \mathcal{N}(0, \beta^{-1}M)$
5: generate a proposal by integrating Hamiltonian dynamics

$$(\mathbf{q}', \mathbf{p}') = \Psi_{\Delta t, L}(\mathbf{q}, \mathbf{p}^*)$$

6: calculate the acceptance probability

$$\alpha = \min\left\{1, \exp\left(-\beta\left(H(\mathbf{q}', \mathbf{p}') - H(\mathbf{q}, \mathbf{p}^*)\right)\right)\right\}$$

7: Metropolis test
draw $u \sim \mathcal{U}(0, 1)$
if $u < \alpha$
accept: $(\mathbf{q}^n, \mathbf{p}^n) = (\mathbf{q}', \mathbf{p}')$
else
reject and flip momenta: $(\mathbf{q}^n, \mathbf{p}^n) = \mathcal{F}(\mathbf{q}, \mathbf{p}^*)$

end if
8: **end for**

Note that the formulation above differs from the original one (Kennedy and Pendleton, 2001) in the number of momentum flips performed. In the original formulation, the momentum flip (2.14) is applied before partial momentum refreshment and once again upon acceptance, instead of rejection; thus more momentum flips are needed in this case. It is easy to see that the two formulations are equivalent.

Some well-known methods can be considered as special cases of GHMC (therefore, "generalized"):

- **Hybrid Monte Carlo (HMC)**: If $\varphi = \pi/2$ the momenta are completely resampled. Then, the momentum flips may be ignored in this case since $\mathbf{p}^* = \mathbf{u}$ and the previous momenta are entirely discarded.
- **Langevin Dynamics (LD)**: If all MD proposals are accepted and $\varphi = \sqrt{2\gamma\Delta t} \ll 1$, where $\gamma > 0$ plays the role of the friction coefficient.
- **Molecular Dynamics (MD)**: If $\varphi = 0$ and all trajectories are accepted, meaning that one long trajectory is produced in the NVE ensemble.

The three examples above are summarized in Table 2.1.

Method	φ	L	**Metropolis test**
HMC	$\pi/2$	arbitrary	✓
LD	$\varphi = \sqrt{2\gamma\Delta t} \ll 1$, $\gamma > 0$	1	✗
MD	0	arbitrary	✗

TABLE 2.1: Special cases of GHMC

2.3 HMC applications

As it has been pointed out in Section 2.1, the HMC method was initially formulated for computational physics and molecular simulation (Gupta, Kilcup, and Sharpe, 1988; Gupta et al., 1990). It was initially applied to lattice field theory simulations and it became popular in QCD studies (Sexton and Weingarten, 1992; Joó et al., 2000; Hasenbusch, 2001; Takaishi and Forcrand, 2006). HMC has also been used in computational chemistry simulations (Tuckerman et al., 1993; Hansmann, Okamoto, and Eisenmenger, 1996).

In the last years, HMC has achieved more popularity in other fields and it has been extensively studied and tested in the computational statistics community (Chen, Qin, and Liu, 2000; Neal, 2011; Girolami and Calderhead, 2011; Radivojević, 2016; Betancourt, 2017; Radivojević and Akhmatskaya, 2017). HMC remained unknown for statistical applications until 1994 when Neal used the method in neural network models in his Ph.D. thesis. In the computational statistics community, the method is usually called *Hamiltonian* Monte Carlo. Nowadays, HMC is used in a wide range of applications, from molecular simulations to statistical problems appearing in different fields, such as data assimilation or geophysics (Alexander, Eyink, and Restrepo, 2005; Mohamed, Christie, and Demyanov, 2010).

Still, the usage of HMC has been limited by the poor performance when the size of a simulated problem increases. A straightforward solution could be to decrease the time step

used in the numerical integration. However, this leads to an increase of the computational time of a simulation. In this dissertation, we propose different approaches to enhance the sampling of complex systems using HMC methods. Mainly, we suggest two solutions: *splitting integrators* and the *importance sampling*. In the following chapters, the state of the art of both approaches will be summarized, and the novel methodologies will be proposed and explained in detail.

2.4 Summary

In this chapter, the Hybrid Monte Carlo (HMC) algorithm has been presented. We also discuss some improvements and extensions of HMC. In particular, we summarize the Generalized Hybrid Monte Carlo (GHMC), which will be fundamental in the following chapters of this dissertation. The main technical details of the formulations of HMC and GHMC have been explained, and the proof of the validity of both methods has been provided. The useful references are also supplied.

Chapter 3

Enhancing Performance and Accuracy of HMC for Simulation of Complex Systems: Numerical Integrators

3.1 Overview

The efficiency and even the feasibility of molecular dynamics simulations depend crucially on the choice of a numerical integrator. As to the role of integrators in enhancing the performance of Hybrid Monte Carlo, it has been a subject of active research in recent years (McLachlan, 1995; Blanes, Casas, and Sanz-Serna, 2014; Chao et al., 2015; Campos and Sanz-Serna, 2017; Bou-Rabee and Sanz-Serna, 2017a). The velocity Verlet algorithm is currently the method of choice; its algorithmic simplicity and optimal stability properties make it very difficult to beat. Splitting integrators may, however, offer the possibility of improving on Verlet, at least in some circumstances. Those integrators evaluate the forces more than once per step and, due to their simple kick-drift structure, may be implemented easily by modifying existing implementations of the Verlet scheme. In this chapter, we survey some splitting schemes that, in our notation, are classified by the number of force evaluations per time step. We study in detail mainly two-stage integrators, which are those splitting schemes that perform two force evaluations per time step. The three-stage integrators are also presented.

Two-stage integrators form a one-parameter family (Blanes, Casas, and Sanz-Serna, 2014). The value of a parameter for a two-stage integrator that results in a method leading to the smallest energy error was first identified by McLachlan, 1995. While McLachlan's scheme is a good choice for many given problems if the time step Δt is very small, it turns out that its stability interval is not long. This entails that in molecular simulations, where small time steps are prohibitively expensive, McLachlan's method is likely not a good choice. Then, one has to sacrifice the size of the error constant to ensure that the integrator is able to operate satisfactorily with larger time steps.

Blanes, Casas, and Sanz-Serna, 2014 have suggested choosing a parameter value, for multi-stage integrators, so that a balance between good conservation of energy for reasonable values of Δt and accuracy for small Δt is achieved.

The parameter values of (McLachlan, 1995; Blanes, Casas, and Sanz-Serna, 2014) do not vary with the problem being considered or with the value of Δt attempted by the user. On the contrary, the method we propose here for two-stage integrators, and which we call Adaptive Integration Approach or AIA, automatically adjusts the parameter value for each problem and each choice of Δt. On stability grounds, for any given problem, there is a maximum possible value of Δt; beyond this maximum all integrators in the family are unstable. When

the time step chosen by the user is near the maximum value, AIA picks up an integrator that is (equivalent to) the standard Verlet scheme. As Δt decreases, AIA changes the integrator to ensure optimal conservation of energy; for Δt close to 0, AIA chooses McLachlan's scheme.

In the case of three-stage integrators, the two-parameters family presented in (Blanes, Casas, and Sanz-Serna, 2014) has been recently reduced to a one-parameter family in (Campos and Sanz-Serna, 2017) offering the extra stability conditions for the choice of such parameter.

This chapter begins with an introduction to symplectic integrators in Section 3.2. The special attention is devoted to the Verlet method, the splitting integrators (namely two- and three-stage schemes) and the Liouville propagator. In Section 3.3 the novel AIA algorithm is formulated in detail. Its implementation in the in-house code is discussed in Section 3.4. To prove the validity of the AIA method and illustrate its main functionalities tests are proposed in Section 3.5. The numerical results are shown in Section 3.6. The chapter ends with some conclusions and possible future work in Section 3.7.

3.2 Symplectic integrators

As it is explained in detail in (Sanz-Serna and Calvo, 1994), any symplectic integration scheme applied to a nonlinear autonomous system with Hamiltonian $H(\mathbf{q}, \mathbf{p})$ is equivalent to the exact sampling of some perturbed nonautonomous system with an effective Hamiltonian $H_{\text{eff}}(\mathbf{q}, \mathbf{p}, \Delta t)$, where

$$H_{\text{eff}}(\mathbf{q}, \mathbf{p}, \Delta t) = H(\mathbf{q}, \mathbf{p}) + \mathcal{O}((\Delta t)^{\nu}),$$

where ν denotes the order of the integration scheme.

A mapping Ψ that transforms coordinates and momenta at time t, $(\mathbf{q}(t), \mathbf{p}(t))$, to coordinates and momenta at time $t + \Delta t$, $(\mathbf{q}(t + \Delta t), \mathbf{p}(t + \Delta t))$ is a symplectic integrator if and only if its Jacobian matrix, Ψ_J, satisfies

$$\Psi_J^T J \Psi_J = J,$$

where Ψ_J is the matrix

$$\Psi_J = \begin{pmatrix} \partial \mathbf{q}(t + \Delta t)/\partial \mathbf{q}(t) & \partial \mathbf{q}(t + \Delta t)/\partial \mathbf{p}(t) \\ \partial \mathbf{p}(t + \Delta t)/\partial \mathbf{q}(t) & \partial \mathbf{p}(t + \Delta t)/\partial \mathbf{p}(t) \end{pmatrix}$$

and

$$J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}.$$

It can be shown that a composition of symplectic transformations is also symplectic and that the inverse of a symplectic mapping is symplectic. These are fundamental properties that will be used in the following sections. The symplectic property translates to good long-time behavior in practice: small fluctuations about the initial (conserved in theory) value of H and no systematic drift in energy.

3.2.1 The Verlet integrator

The integrator used in the original formulation of HMC (Duane et al., 1987) was the *leapfrog* scheme (Feynman, Leighton, and Sands, 1964). It can be written as

$$\begin{aligned} \mathbf{v}(t+\Delta t/2) &= \mathbf{v}(t-\Delta t/2) + \Delta t\ \frac{F(\mathbf{q}(t))}{M} \\ \mathbf{q}(t+\Delta t) &= \mathbf{q}(t) + \Delta t\ \mathbf{v}(t+\Delta t/2), \end{aligned} \tag{3.1}$$

where $\mathbf{v}$ are the velocities, F denotes the forces and M are the masses.

The leapfrog integrator is a second-order method commonly used in the simulation of dynamical systems in classical mechanics. The leapfrog is time-reversible. It is also symplectic (Okunbor and Skeel, 1994). Such properties make the integrator suitable for its use with the Hybrid Monte Carlo method (details can be found in Section 2.1). The leapfrog can also be extended to higher order versions and still be applicable for HMC (Creutz and Gocksch, 1989). However, as it follows from its formulation (3.1), leapfrog's main drawback is that the velocities (or momenta) and the positions are not synchronized in time. In the following sections, we will focus on the method equivalent to leapfrog, namely, velocity Verlet, which provides the synchronization of positions and velocities at the end of every time step.

Let us consider now a Taylor expansion of the position vector in time:

$$\begin{aligned} \mathbf{q}(t+\Delta t) &= \mathbf{q}(t) + \Delta t\ \frac{\mathrm{d}\mathbf{q}(t)}{\mathrm{d}t} + \frac{\Delta t^2}{2}\frac{\mathrm{d}^2\mathbf{q}(t)}{\mathrm{d}t^2} + \frac{\Delta t^3}{6}\frac{\mathrm{d}^3\mathbf{q}(t)}{\mathrm{d}t^3} + \mathcal{O}(\Delta t^4) \\ &= \mathbf{q}(t) + \Delta t\ \mathbf{v}(t) + \frac{\Delta t^2}{2}\frac{F(\mathbf{q}(t))}{M} + \frac{\Delta t^3}{6}\frac{\mathrm{d}^3\mathbf{q}(t)}{\mathrm{d}t^3} + \mathcal{O}(\Delta t^4). \end{aligned} \tag{3.2}$$

The Newton's equation of motion has been used to replace the acceleration with the force. Similarly,

$$\mathbf{q}(t-\Delta t) = \mathbf{q}(t) - \Delta t\ \mathbf{v}(t) + \frac{\Delta t^2}{2}\frac{F(\mathbf{q}(t))}{M} - \frac{\Delta t^3}{6}\frac{\mathrm{d}^3\mathbf{q}(t)}{\mathrm{d}t^3} + \mathcal{O}(\Delta t^4). \tag{3.3}$$

Then, one can sum (3.2) and (3.3) and rearrange them as

$$\mathbf{q}(t+\Delta t) = 2\mathbf{q}(t) - \mathbf{q}(t-\Delta t) + \Delta t^2\frac{F(\mathbf{q}(t))}{M} + \mathcal{O}(\Delta t^4). \tag{3.4}$$

Equation (3.4) is the formulation of the *Verlet integrator* that was first introduced by Verlet, 1967 and it is also known to be symplectic (Ruth, 1983).

The Verlet integrator in (3.4) does not use the velocities to determine the solution of the positions at the next time step. However, we can approximate the velocities using

$$\mathbf{v}(t) = \frac{\mathbf{q}(t+\Delta t) - \mathbf{q}(t-\Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^3),$$

which can be easily derived by substracting (3.3) from (3.2).

One disadvantage of the Verlet algorithm is that it requires storing in memory two sets of positions, $\mathbf{q}(t)$ and $\mathbf{q}(t-\Delta t)$. An alternative is the so-called *velocity Verlet integrator*, which is a reformulation of the Verlet algorithm that uses the velocities directly. The velocity Verlet integrator was first presented by Swope et al., 1982 and it can be obtained by manipulating

equation (3.4). The original formulation of the velocity Verlet scheme is the following:

$$\begin{aligned}\mathbf{q}(t+\Delta t) &= \mathbf{q}(t) + \Delta t\ \mathbf{v}(t) + \frac{\Delta t^2}{2}\frac{F(\mathbf{q}(t))}{M} \\ \mathbf{v}(t+\Delta t) &= \mathbf{v}(t) + \frac{\Delta t}{2}\frac{F(\mathbf{q}(t+\Delta t)) + F(\mathbf{q}(t))}{M}.\end{aligned} \tag{3.5}$$

Velocity Verlet is an *explicit* second-order integrator. Recall that in molecular simulation one often considers systems with a large number of particles, making implicit algorithms intractable. It can be shown that the error on the velocity Verlet is of the same order as that of the regular Verlet. Moreover, both methods are mathematically equivalent, but velocity Verlet is numerically more accurate (cf. (Swope et al., 1982; Tuckerman, Berne, and Martyna, 1992)).

The velocity Verlet algorithm is usually implemented in the following way:

1. $\mathbf{v}(t+\Delta t/2) = \mathbf{v}(t) + \frac{\Delta t}{2}\frac{F(\mathbf{q}(t))}{M}$;
2. $\mathbf{q}(t+\Delta t) = \mathbf{q}(t) + \Delta t\ \mathbf{v}(t+\Delta t/2)$;
3. $\mathbf{v}(t+\Delta t) = \mathbf{v}(t+\Delta t/2) + \frac{\Delta t}{2}\frac{F(\mathbf{q}(t+\Delta t))}{M}$.

Thus, the algorithm is very easy to implement. Moreover, in this formulation, it is not more expensive than in (3.5), since the forces, which are the most computationally demanding part of the integrator, are only computed once per time step, right after the update of positions $\mathbf{q}(t+\Delta t)$. Obviously, the algorithm above could be easily rewritten in terms of momenta.

The choice of an optimal time step Δt for the integration is not trivial. As it has been pointed out above (Section 2.1), the numerical time step plays a crucial role in the acceptance probability of the HMC schemes (cf. (2.12)) and, thus, it affects their sampling performance. A high rate of rejection would increase the cost of the simulation since many samples are discarded in this case. In the next section, we discuss the limitations on a choice of Δt based on the analysis of the harmonic oscillator.

3.2.1.1 Stability analysis of velocity Verlet: Harmonic oscillator

To illustrate the velocity Verlet scheme we consider as a case study the classic example of the harmonic oscillator with potential energy $U(q) = (k/2)q^2$, where $k > 0$ is the force constant. Thus, the forces are computed as $F(q) = -\omega^2 q$. The equations of motion are then

$$\frac{\mathrm{d}q}{\mathrm{d}t} = \frac{p}{M}, \qquad \frac{\mathrm{d}p}{\mathrm{d}t} = -kq. \tag{3.6}$$

The angular frequency is expressed in terms of the force constant as $\omega = \sqrt{k/M}$. We can assume for the sake of simplicity that the mass is trivial and then $p/M = v$ and $\omega = \sqrt{k}$.

A transformation S can be used to relate one phase point to the next (for more details see (Skeel, Zhang, and Schlick, 1997)). Then, for a time step Δt,

$$\begin{pmatrix}\omega q(t+\Delta t) \\ v(t+\Delta t)\end{pmatrix} = S\begin{pmatrix}\omega q(t) \\ v(t)\end{pmatrix}, \tag{3.7}$$

where S is defined as

$$S = \begin{pmatrix}1 - \frac{(\omega\Delta t)^2}{2} & \omega\Delta t \\ -\omega\Delta t + \frac{(\omega\Delta t)^3}{4} & 1 - \frac{(\omega\Delta t)^2}{2}\end{pmatrix}. \tag{3.8}$$

The details on the construction of S can be found in Appendix A.1. A numerical integrator is stable if the matrix S is power bounded. This is satisfied if the eigenvalues of S lie in the unit disc. The eigenvalues of S are

$$\lambda_1 = 1 - \frac{(\omega\Delta t)^2}{2} + \sqrt{\frac{(\omega\Delta t)^4}{4} - (\omega\Delta t)^2}, \quad \lambda_2 = 1 - \frac{(\omega\Delta t)^2}{2} - \sqrt{\frac{(\omega\Delta t)^4}{4} - (\omega\Delta t)^2}. \tag{3.9}$$

Thus, the matrix S is power bounded if and only if

$$(\omega\Delta t)^2 < 4,$$

or, equivalently,

$$\Delta t < 2/\omega. \tag{3.10}$$

The restriction on the time step above is the *linear stability* condition for Verlet. Then, we can define the *stability interval* of an integrator as the largest interval $(0, \Delta t_{\max})$ such that the method is stable for each time step Δt that satisfies $0 < \Delta t < \Delta t_{\max}$.

Under the linear stability assumption, the matrix S in (3.8) has eigenvalues $\exp(\pm i\theta)$, where[1]

$$\theta = 2\arcsin(\omega\Delta t/2) \tag{3.11}$$

$$= \omega\Delta t + \frac{1}{24}(\omega\Delta t)^3 + \mathcal{O}(\omega\Delta t)^5. \tag{3.12}$$

Thus, the angle θ depends on the time step and the frequency ω. To see that this transformation defines a rotation in phase space, we decompose the phase-space transforming matrix S as

$$S = DQD^{-1},$$

where

$$Q = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

defines a rotation of $-\theta$ radians in phase space, and D is the diagonal matrix

$$D = \begin{pmatrix} 1 & 0 \\ 0 & 1 + \tan^2\frac{\theta}{2} \end{pmatrix}.$$

Thus, the behavior of the integrator in time can be interpreted through analysis of the powers of S given by

$$S^n = DQ^nD^{-1}.$$

The time step-dependent behavior of the transformation S can be interpreted as follows. Equations (3.11) and (3.12) show that the integrator uses θ as an approximation to the exact rotation $\omega\Delta t$. The smaller the time step, the closer the approximation is. Thus, one can define the *effective rotation* θ_{eff} as

$$\theta_{\text{eff}} = \omega_{\text{eff}}\,\Delta t.$$

[1]The angle of the rotation matrix is computed from the trace as $\text{Tr}(S) = 2\cos\theta$. Thus, $\theta = \arccos\left(\frac{\text{Tr}(S)}{2}\right) = \arccos\left(1 - \frac{(\omega\Delta t)^2}{2}\right) = 2\arcsin\left(\omega\Delta t/2\right)$.

For the Verlet method, the effective rotation is given by equation (3.11)

$$\theta_{\text{eff}}^{\text{Verlet}} = 2\arcsin(\omega_{\text{eff}}\,\Delta t/2). \tag{3.13}$$

For periodic motion with natural frequency ω, nonphysical resonance (an artifact of the symplectic integrator) can occur when ω is related by relatively prime integers n and m to the forcing frequency $(2\pi/\Delta t)$ (cf. (Arnold, 1989)):

$$\frac{n}{m}\omega = \frac{2\pi}{\Delta t}.$$

Here n is the *resonance order*.

Now, if we recall that the Verlet method has the time step-dependent frequency ω_{eff} given by $\theta_{\text{eff}}^{\text{Verlet}}/\Delta t$ with $\theta_{\text{eff}}^{\text{Verlet}}$ as in (3.13), we see that the frequency ω_{eff} depends on the time step in a nonlinear way. Thus, the integrator-dependent resonance condition becomes

$$\frac{n}{m}\omega_{\text{eff}} = \frac{2\pi}{\Delta t}. \tag{3.14}$$

A *resonance of order* $n : m$ means that n phase space points are sampled in m revolutions:

$$n\,\theta_{\text{eff}} = n\,\Delta t\,\omega_{\text{eff}} = 2\pi m.$$

This special, finite-coverage of phase space can lead to incorrect, limited sampling of configuration space. As it has been shown by Mandziuk and Schlick, 1995, equation (3.13) can be used to formulate *a condition for a resonant time step* for the harmonic oscillator system. That is, using

$$\omega_{\text{eff}}^{\text{Verlet}} = \frac{2\sin^{-1}(\omega\,\Delta t/2)}{\Delta t},$$

with the resonance condition in (3.14), we have

$$\frac{\omega\,\Delta t}{2} = \sin\left(\frac{m\pi}{n}\right).$$

Equivalently,

$$\Delta t_{n:m}^{\text{Verlet}} = \frac{2}{\omega}\sin\left(\frac{m\pi}{n}\right).$$

To get the lowest-order resonances (which are the most severe) we take $m = 1$. It is easy to see that for $n = 2$, we recover the linear stability condition in (3.10).

It is clear that, since the limiting time steps $\Delta t_{n:1}$ for resonance orders $n > 2$ are smaller than the linear stability limit $\Delta t_{2:1}$, resonance limits the time step to values lower than classical stability. Since the third-order resonance leads to instability and the fourth-order resonance often leads to instability in molecular simulation[2], in practice it is usually required that $\Delta t < \Delta t_{4:1}$. This implies for Verlet a stricter restriction than (3.10)

$$\Delta t < \sqrt{2}/\omega,$$

which corresponds to the fourth-order resonance and is the *non linear stability* condition for Verlet. More resonance time step limits are summarized in Table 3.1.

[2]It was predicted by Arnold, 1989 that instabilities are not observed for resonances of orders higher than four and this assessment has been confirmed with experiments such as in (Mandziuk and Schlick, 1995).

Order n	$\Delta t_{n:1}(\omega)$
2	$2/\omega$
3	$\sqrt{3}/\omega$
4	$\sqrt{2}/\omega$
5	$1.176/\omega$
6	$1/\omega$

TABLE 3.1: Resonant time step limits for different orders n.

3.2.2 Splitting methods

The basic idea of *splitting methods* for the integration of ordinary differential equations can be formulated as follows. Given an initial value problem

$$x' = f(x), \qquad x_0 = x(0) \in \mathbb{R}^D, \tag{3.15}$$

with $f\colon \mathbb{R}^D \to \mathbb{R}^D$ and solution $\phi_t(x_0)$, let us suppose that f can be expressed as $f = \sum_{i=1}^{m} f^i$ for certain functions $f^i : \mathbb{R}^D \to \mathbb{R}^D$ in such a way that the equations

$$x' = f^i(x), \qquad x_0 = x(0) \in \mathbb{R}^D, \qquad i = 1, \ldots, m$$

can be integrated exactly with solutions $x(h) = \phi_h^i(x_0)$ at $t = h$. Then, one can combine these solutions as

$$\psi_h = \phi_h^m \circ \cdots \circ \phi_h^2 \circ \phi_h^1.$$

Expanding ψ in Taylor series, one gets $\psi_h(x_0) = \phi_h(x_0) + \mathcal{O}(h^2)$. Thus, ψ_h provides a first-order approximation to the exact solution. Therefore, splitting methods involve three steps: (i) choosing the set of functions f^i such that $f = \sum_{i=1}^{m} f^i$; (ii) solving either exactly or approximately each equation $x' = f^i(x)$; and (iii) combining these solutions to construct an approximation for (3.15). Obviously, the equations $x' = f^i(x)$ should be easier to integrate than (3.15). Informative reviews of splitting integrators can be found in (McLachlan and Quispel, 2002; Blanes, Casas, and Murua, 2008).

Here we introduce the notation h to refer to dimensionless time steps that are only used in theoretical scenarios in contrast to Δt that are simulation time steps expressed in units of time. These two notations will be consistent through the whole dissertation.

The ideas of the splitting methods can be easily extended to molecular simulations where separable Hamiltonians are considered. We introduce here the useful notation of writing the Hamiltonian as a sum $H \equiv A + B$ of two partial Hamiltonian functions:

$$A(\mathbf{q}, \mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}, \qquad B(\mathbf{q}, \mathbf{p}) = U(\mathbf{q}), \tag{3.16}$$

where A and B correspond to the kinetic and potential energies, respectively. Thus, the Hamilton equations of motion in (1.3) can be written as

$$\frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} = \nabla_{\mathbf{p}} A(\mathbf{q}, \mathbf{p}) = M^{-1}\mathbf{p}, \qquad \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = -\nabla_{\mathbf{q}} B(\mathbf{q}, \mathbf{p}) = -\nabla_{\mathbf{q}} U(\mathbf{q}). \tag{3.17}$$

These equations can be integrated in closed form and their solution flows at a time t are respectively given by

$$(\mathbf{q}(t), \mathbf{p}(t)) = \phi_t^A(\mathbf{q}(0), \mathbf{p}(0)), \quad \mathbf{q}(t) = \mathbf{q}(0) + t\ M^{-1}\mathbf{p}(0), \quad \mathbf{p}(t) = \mathbf{p}(0), \tag{3.18}$$

and

$$(\mathbf{q}(t), \mathbf{p}(t)) = \phi_t^B(\mathbf{q}(0), \mathbf{p}(0)), \quad \mathbf{q}(t) = \mathbf{q}(0), \quad \mathbf{p}(t) = \mathbf{p}(0) - t\ \nabla_{\mathbf{q}} U(\mathbf{q}(0)). \tag{3.19}$$

Here ϕ_t^A and ϕ_t^B denote the exact solution flows of the partial systems, i.e., the maps that associate the exact solution value $(\mathbf{q}(t), \mathbf{p}(t))$ with each initial condition $(\mathbf{q}(0), \mathbf{p}(0))$. Sometimes (3.18) might also be called a *drift* in the position and (3.19) a momentum *kick*.

Thus, a velocity Verlet time step, as the one in (3.5), corresponds to a transformation in phase space $(\mathbf{q}(t + \Delta t), \mathbf{p}(t + \Delta t)) = \psi_{\Delta t}(\mathbf{q}(t), \mathbf{p}(t))$ that can be written as

$$\psi_{\Delta t} = \phi^B_{\Delta t/2} \circ \phi^A_{\Delta t} \circ \phi^B_{\Delta t/2}. \tag{3.20}$$

This formulation is summarized in Figure 3.1.

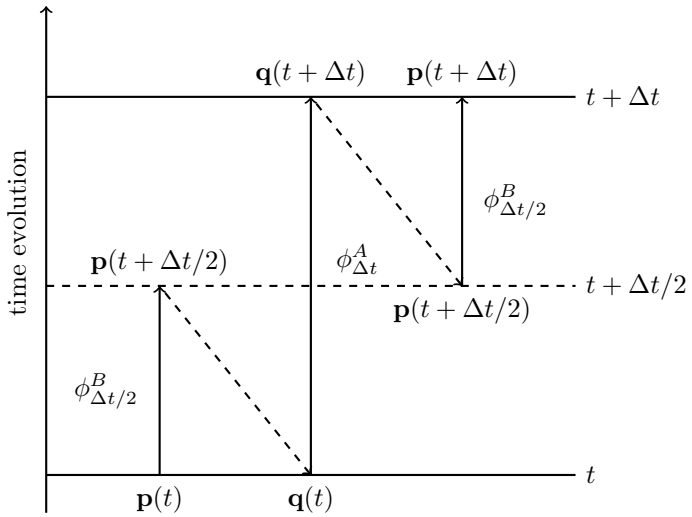


FIGURE 3.1: A step of the velocity Verlet integrator viewed as a splitting scheme with time step Δt.

Here $\psi_{\Delta t}$ is *volume preserving* as a composition of volume preserving Hamiltonian flows. Furthermore $\psi_{\Delta t}$ is *reversible* because $\phi^B_{\Delta t/2}$ and $\phi^A_{\Delta t}$ are both reversible and the right-hand side of (3.20) is a palindrome:

$$\begin{aligned}
\psi_{\Delta t}^{-1} &= (\phi^B_{\Delta t/2})^{-1} \circ (\phi^A_{\Delta t})^{-1} \circ (\phi^B_{\Delta t/2})^{-1} \\
&= (\mathcal{F} \circ \phi^B_{\Delta t/2} \circ \mathcal{F}) \circ (\mathcal{F} \circ \phi^A_{\Delta t} \circ \mathcal{F}) \circ (\mathcal{F} \circ \phi^B_{\Delta t/2} \circ \mathcal{F}) \\
&= \mathcal{F} \circ \psi_{\Delta t} \circ \mathcal{F},
\end{aligned}$$

where $\mathcal{F}$ denotes the momentum flip. Velocity Verlet is also *symplectic*. Its symplecticness is a direct consequence of two facts (cf. (Arnold, 1989; Sanz-Serna and Calvo, 1994; Leimkuhler and Reich, 2004; Hairer, Lubich, and Wanner, 2006)):

1. Hamiltonian flows like ϕ_t^A and ϕ_t^B are symplectic.
2. The composition of symplectic transformations is symplectic.

One can also define the transformation $\Psi = \Psi_{\Delta t, L}$ over L time steps as the composition

$$\Psi = \Psi_{\Delta t, L} = \underbrace{\psi_{\Delta t} \circ \cdots \circ \psi_{\Delta t}}_{L \text{ times}}.$$

The symplecticity and time-reversibility are preserved in this case for the same reasons.

Blanes, Casas, and Sanz-Serna, 2014 suggest to replace the Verlet formulas by more sophisticated palindromic compositions such as

$$\psi_{\Delta t} = \phi^B_{b_1 \Delta t} \circ \phi^A_{a_1 \Delta t} \circ \phi^B_{b_2 \Delta t} \circ \phi^A_{a_2 \Delta t} \circ \phi^B_{b_2 \Delta t} \circ \phi^A_{a_1 \Delta t} \circ \phi^B_{b_1 \Delta t} \tag{3.21}$$

or

$$\psi_{\Delta t} = \phi^A_{a_1 \Delta t} \circ \phi^B_{b_1 \Delta t} \circ \phi^A_{a_2 \Delta t} \circ \phi^B_{b_2 \Delta t} \circ \phi^a_{a_2 \Delta t} \circ \phi^B_{b_1 \Delta t} \circ \phi^A_{a_1 \Delta t}, \tag{3.22}$$

where a_1, a_2, b_1, b_2 are real numbers. The integration scheme in (3.21) is the *velocity* version of the scheme in (3.22), due to the fact that in each step the velocities are updated the first. Using a more simplified notation, where only the parameters of the flows ϕ are used, one may consider *r-stage* ($r = 1, 2, \ldots$) compositions

$$\underbrace{(b_1, a_1, b_2, \ldots, a_1, b_1)}_{2r+1 \text{ letters}} \tag{3.23}$$

and

$$\underbrace{(a_1, b_1, a_2, \ldots, b_1, a_1)}_{2r+1 \text{ letters}}. \tag{3.24}$$

It is clear that schemes such as (3.23) and (3.24) require r evaluations of forces $-\nabla_{\mathbf{q}} U$ at each time step. In (3.23) the positions are updated r times, and thus the forces are evaluated r times. On the other hand, in (3.24) there is an $r+1$th update of the positions but the force evaluation is calculated in the next step, and thus the forces are evaluated r times again. The force evaluations are the most computationally demanding part of molecular simulations and thus they drive the computational cost of any simulation. The term *r-stage* to refer to splitting schemes has been introduced in (Blanes, Casas, and Sanz-Serna, 2014) and will be used in this dissertation. In the following chapters, we will limit our studies to one-stage (leapfrog and velocity Verlet), two-stage and three-stage schemes in their *velocity* formulations.

3.2.2.1 Two-stage schemes

If we restrict ourselves to the velocity scenario, the two-stage splitting schemes have the form

$$\psi_{\Delta t} = \phi^B_{b_1 \Delta t} \circ \phi^A_{a_1 \Delta t} \circ \phi^B_{b_2 \Delta t} \circ \phi^A_{a_1 \Delta t} \circ \phi^B_{b_1 \Delta t}.$$

To be well defined[3], the integrators above have to satisfy $a_1 = 1/2$ and $b_2 = 1 - 2b_1$. This leaves the one-parameter family

$$\psi_{\Delta t} = \phi^B_{b \Delta t} \circ \phi^A_{\Delta t/2} \circ \phi^B_{(1-2b)\Delta t} \circ \phi^A_{\Delta t/2} \circ \phi^B_{b \Delta t}, \tag{3.25}$$

[3] $a_1 + a_1 = 1$ and $b_1 + b_2 + b_1 = 1$.

where, for simplicity we use the notation $b = b_1$.

As any other r-stage splitting scheme, the two-stage splitting integrators (3.25) are symplectic, being the composition of symplectic flows, and time-reversible due to their palindromic construction.

It is useful in what follows to rewrite (3.25) as

$$\psi_{\Delta t} = \left(\phi^B_{b\Delta t} \circ \phi^A_{\Delta t/2} \circ \phi^B_{(1/2-b)\Delta t}\right) \circ \left(\phi^B_{(1/2-b)\Delta t} \circ \phi^A_{\Delta t/2} \circ \phi^B_{b\Delta t}\right). \tag{3.26}$$

The map $\phi^B_{(1/2-b)\Delta t} \circ \phi^A_{\Delta t/2} \circ \phi^B_{b\Delta t}$ advances the solution over a first half step of length $\Delta t/2$ and is followed by the map $\phi^B_{b\Delta t} \circ \phi^A_{\Delta t/2} \circ \phi^B_{(1/2-b)\Delta t}$ that effects a second half step, also of length $\Delta t/2$. In the particular case $b = 1/4$ both of these maps correspond to a step of length $\Delta t/2$ of the velocity Verlet (VV) algorithm:

$$\psi_{\Delta t} = \left(\phi^B_{\Delta t/4} \circ \phi^A_{\Delta t/2} \circ \phi^B_{\Delta t/4}\right) \circ \left(\phi^B_{\Delta t/4} \circ \phi^A_{\Delta t/2} \circ \phi^B_{\Delta t/4}\right) = \psi^{\mathrm{VV}}_{\Delta t/2} \circ \psi^{\mathrm{VV}}_{\Delta t/2}.$$

For other values of b the half step maps in (3.26) do not coincide with the map of the velocity Verlet integrator, because the durations $b\Delta t$ and $(1/2-b)\Delta t$ differ. However, regardless of the choice of b, the half step maps have the same structure of velocity Verlet, which makes them easy to implement simply by modifying the velocity Verlet implementation (more details will be provided later).

The two-stage integrators are represented graphically in Figure 3.2.

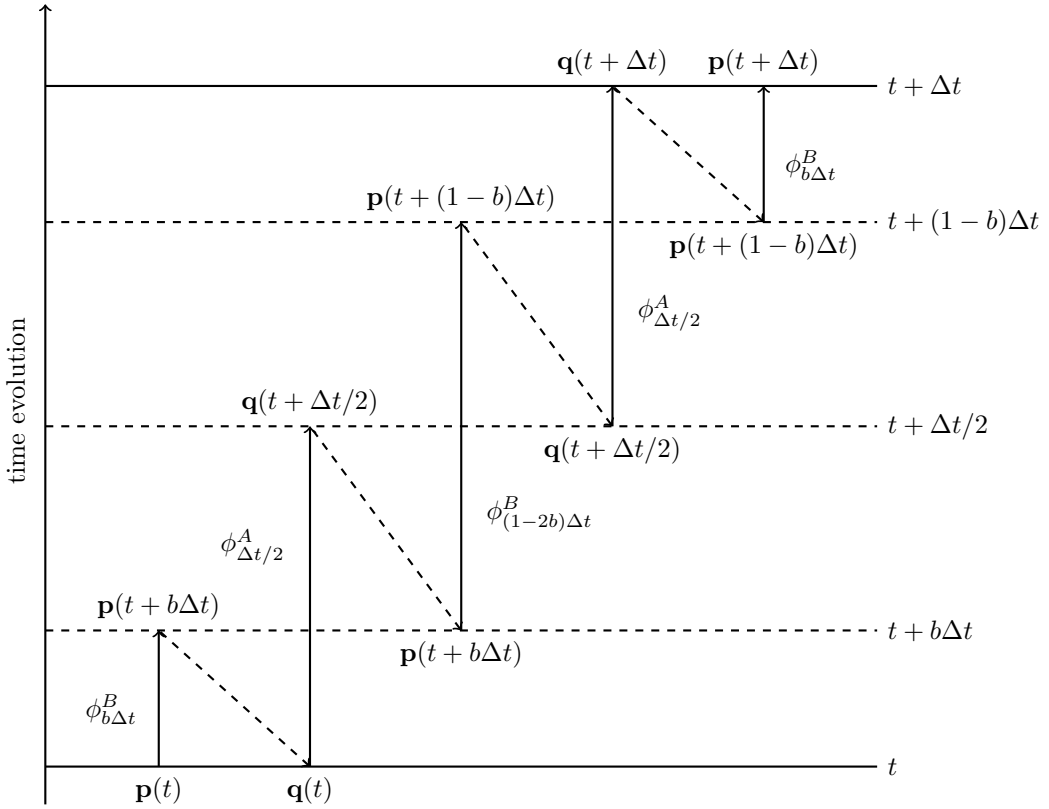


FIGURE 3.2: A step of a generic two-stage splitting scheme with parameter b and time step Δt.

3.2.2.2 Three-stage schemes

As in the previous case, if we restrict ourselves to the velocity scenario, the three-stage splitting schemes have the form

$$\psi_{\Delta t} = \phi^B_{b_1\Delta t} \circ \phi^A_{a_1\Delta t} \circ \phi^B_{b_2\Delta t} \circ \phi^A_{a_2\Delta t} \circ \phi^B_{b_2\Delta t} \circ \phi^A_{a_1\Delta t} \circ \phi^B_{b_1\Delta t}.$$

To be well defined[4], the integrators above have to satisfy $2a_1 + a_2 = 1$ and $b_1 + b_2 = 1/2$. Thus, the integrator can be rewritten as

$$\psi_{\Delta t} = \phi^B_{b\Delta t} \circ \phi^A_{a\Delta t} \circ \phi^B_{\left(\frac{1}{2}-b\right)\Delta t} \circ \phi^A_{(1-2a)\Delta t} \circ \phi^B_{\left(\frac{1}{2}-b\right)\Delta t} \circ \phi^A_{a\Delta t} \circ \phi^B_{b\Delta t}, \tag{3.27}$$

in which, for simplicity, we use the notation $b = b_1$ and $a = a_1$.

It has been recently proved in (Campos and Sanz-Serna, 2017) that the integrators that lie on the hyperbola

$$6ab - 2a - b + \frac{1}{2} = 0$$

have considerably longer stability limit than others. Thus, this condition can be written as

$$a = \frac{1 - 2b}{4(1 - 3b)}$$

and it leaves again a one-parameter family

$$\left(b,\ \frac{1-2b}{4(1-3b)},\ \frac{1}{2} - b,\ \frac{1-4b}{2(1-3b)},\ \frac{1}{2} - b,\ \frac{1-2b}{4(1-3b)},\ b\right).$$

One drawback related to the software implementation of integrators from this family is that they cannot be divided in three equal velocity Verlet substeps as in (3.26).

3.2.2.3 Stability analysis of splitting integrators: Harmonic oscillator

As in the case of the Verlet integrator (Section 3.2.1.1), we refer to the harmonic oscillator in order to discuss the stability properties of splitting schemes. In this case, we assume that the frequency and the mass are both one. Thus, we have the Hamiltonian

$$H = \frac{1}{2}p^2 + \frac{1}{2}q^2 \tag{3.28}$$

and the simpler equations of motion

$$\frac{\mathrm{d}}{\mathrm{d}t}q = p, \qquad \frac{\mathrm{d}}{\mathrm{d}t}p = -q. \tag{3.29}$$

In this section, we use the notation of h to refer to dimensionless time steps (cf. Section 3.2.2). It is clear that, since the frequency is assumed to be one, the product $\omega\Delta t$ in (3.8) ends up in a time step without dimensions. The relation between time steps and frequencies, in a non-trivial scenario, will be the matter of discussion in the following sections.

[4] $a_1 + a_2 + a_1 = 1$ and $b_1 + b_2 + b_2 + b_1 = 1$.

A step $(q(t+h), p(t+h)) = \psi_h(q(h), p(h))$ of an integrator may be expressed as

$$\begin{pmatrix} q(t+h) \\ p(t+h) \end{pmatrix} = S_h \begin{pmatrix} q(t) \\ p(t) \end{pmatrix}, \tag{3.30}$$

with

$$S_h = \begin{pmatrix} A_h & B_h \\ C_h & D_h \end{pmatrix}, \tag{3.31}$$

for suitable integrator-dependent coefficients A_h, B_h, C_h, D_h. For instance, for two-stage integrators (3.25), the resulting coefficients of S_h are

$$\begin{aligned} A_h = D_h =& 1 - \frac{h^2}{2} + b(1-2b)\frac{h^4}{4} \\ B_h =& h + (2b-1)\frac{h^3}{4} \\ C_h =& -h + b(1-b)h^3 - b^2(1-2b)\frac{h^5}{4}. \end{aligned} \tag{3.32}$$

Equivalently, for the three-stage integrators (3.27)

$$\begin{aligned} A_h = D_h =& 1 - \frac{h^2}{2} + a(1-4b^2-a(2-4b))\frac{h^4}{4} + a^2(2a-1)(1-2b)^2 b\frac{h^6}{4} \\ B_h =& h + a(1-a)(2b-1)h^3 + a^2(1-2a)(1-2b)^2\frac{h^5}{4} \\ C_h =& -h + (1-2a(1-2b)^2)\frac{h^3}{4} + a(2a(1-b)-1)b(1-2b)\frac{h^5}{2} \\ &+ a^2(1-2a)(1-2b)^2 b^2\frac{h^7}{4} \end{aligned} \tag{3.33}$$

The details can be found in Appendix A.1. The subindex h is used in the matrix S (cf. (3.7)) and all its elements to denote a dependence on the time step. Therefore, the evolution over time is given by

$$\begin{pmatrix} q(nh) \\ p(nh) \end{pmatrix} = S_h^n \begin{pmatrix} q(0) \\ p(0) \end{pmatrix}. \tag{3.34}$$

Since we are interested in simulations with HMC methods, time-reversibility and volume preservation[5] are desired properties (cf. Section 2.1):

- Time-reversibility: condition (2.10) leads to $A_h = D_h$.
- Volume preservation: condition (2.11) leads to $A_h D_h - B_h C_h = A_h^2 - B_h C_h = 1$.

The matrix S_h has two eigenvalues:

$$\lambda_1 = A_h + \sqrt{A_h^2 - 1}, \quad \lambda_2 = A_h - \sqrt{A_h^2 - 1}.$$

Clearly, they agree with the expressions for the eigenvalues previously found in (3.9). To ensure the stability of the method, both eigenvalues have to be in the unit disk :

- $|\lambda_1 \lambda_2| = A_h^2 - \left(\sqrt{A_h^2 - 1}\right)^2 = 1$.

[5]Which is equivalent to symplecticness in this case, since we are working in dimension 1.

- $|\lambda_1 + \lambda_2| \leq 2 \Rightarrow |2A_h| \leq 2 \Rightarrow |A_h| \leq 1$.

Thus, to assure a consistent and stable method, with h positive and sufficiently small (cf. (Blanes, Casas, and Sanz-Serna, 2014)),

$$A_h = 1 - h^2/2 + \mathcal{O}(h^3), \qquad h \to 0, \tag{3.35}$$

which is also in agreement with (3.8).

As in Section 3.2.1.1, for a time step h such that stability is satisfied, we can introduce an angle θ_h such that $A_h = \cos\theta_h$. Since for $|A_h| < 1$ it is clear that $\sin\theta_h \neq 0$, we can define

$$\chi_h = B_h / \sin\theta_h. \tag{3.36}$$

Then, the matrices S_h and S_h^i in equations (3.31) and (3.34) can be rewritten as

$$S_h = \begin{pmatrix} \cos\theta_h & \chi_h \sin\theta_h \\ -\chi_h^{-1}\sin\theta_h & \cos\theta_h \end{pmatrix}$$

and

$$S_h^n = \begin{pmatrix} \cos(n\theta_h) & \chi_h \sin(n\theta_h) \\ -\chi_h^{-1}\sin(n\theta_h) & \cos(n\theta_h) \end{pmatrix}.$$

A method with $\theta_h = h$ would have no phase error: the angular frequency of the rotation of the numerical solution would coincide with the true angular rotation of the harmonic oscillator. On the other hand, a method with $\chi_h = 1$ would have no energy error: the numerical solution would remain on the correct level curve of the Hamiltonian (3.28), i.e., on the circle $p^2 + q^2 = p_0^2 + q_0^2$.

In (Blanes, Casas, and Sanz-Serna, 2014) the authors find, for an integration of L steps, the value of the expectation of the energy error ΔH (cf. (2.8))

$$\mathbb{E}(\Delta H) = \sin^2(L\theta_h)\rho(h),$$

where

$$\rho(h) = \frac{1}{2}\left(\chi_h^2 + \frac{1}{\chi_h^2} - 2\right) = \frac{1}{2}\left(\chi_h - \frac{1}{\chi_h}\right)^2 \geq 0,$$

with χ_h as in (3.36). Thus, since the term $\sin^2(L\theta_h)$ is bounded by one, we get

$$0 \leq \mathbb{E}(\Delta H) \leq \rho(h). \tag{3.37}$$

As further proposed in (Blanes, Casas, and Sanz-Serna, 2014), ρ can be expressed in terms of the elements of the matrix (3.31) in the stable case ($|A_h| < 1$):

$$\rho(h) = \frac{(B_h + C_h)^2}{2(1 - A_h^2)}. \tag{3.38}$$

It has to be remarked that the stability condition $|A_h| < 1$ is equivalent to the positivity of the denominator in (3.38).

We will refer to this function in the following chapters due to its important role in bounding the expected energy error (cf. (3.37)).

The application of any r-stage method of the form (3.23) to the standard harmonic oscillator (3.28) results in a recursion of the form (3.30). Moreover, A_h in (3.31) is a polynomial of

degree $\leq r$ in $\zeta = h^2$ and, for consistent methods has the form of (3.35). It can be seen that $-1 \leq A_\zeta \leq 1$ cannot be satisfied for every $0 < \zeta < \zeta_{\max}$ if $\zeta_{\max} > 4r^2$ (Jeltsch and Nevanlinna, 1981; Sanz-Serna and Spijker, 1986). Thus, since the velocity Verlet algorithm has stability interval (0, 2) (cf. (3.10)), a concatenation $\psi_h = \phi^{\mathrm{VV}}_{h/r} \circ \cdots \circ \phi^{\mathrm{VV}}_{h/r}$ of r time steps of length h/r is a method of the form (3.23) that attains the optimal value $h_{\max} = 2r$. This is a key fact of the velocity Verlet integrator and we will refer to it in the following sections.

From now on, when comparing the size of stability intervals, the computational effort will be taken into account: with a given amount of computational work, an integrator with fewer function evaluations per time step may take shorter time steps to span a given time interval. Therefore, we will *normalize* the length $h_{\max}$ of the stability intervals by dividing by the number r of force evaluations per time step.

3.2.3 Trotter expansion of the Liouville propagator

In this section, we introduce a useful and common notation in the molecular dynamics field. The Trotter expansion of the classical Liouville propagator can be used to derive simple integrators (De Raedt and De Raedt, 1983). The Liouville formalism is a tool for building symplectic and reversible integrators. An introduction to the Liouville operator can be found in (Tuckerman, 2010). The Liouville operator $i\hat{L}$ is defined as

$$i\hat{L} = \sum_{i=1}^{D} \left[\frac{\partial H}{\partial p_i} \frac{\partial}{\partial q_i} - \frac{\partial H}{\partial q_i} \frac{\partial}{\partial p_i} \right] = \{\cdot, H\}, \tag{3.39}$$

where $\{\cdot, \cdot\}$ denotes the Poisson bracket (cf. (Arnold, 1989)). It is a linear Hermitian operator on the space of square integrable functions of the phase space. Thus, the classical propagator $u(t) = e^{i\hat{L}t}$ is a unitary operator. The unitarity of $u(t)$ implies time-reversibility. Let $\Gamma(t)$ denote the positions and momenta of the system at time t. Then,

$$\Gamma(t) = u(t)\Gamma(0) \Rightarrow u(-t)\Gamma(t) = u(-t)u(t)\Gamma(0) = \Gamma(0),$$

where the unitarity is used in the fact that $u(-t) = u^{-1}(t)$.

Since $u(t)$ is unitary, it is possible to show that its determinant is 1. In order to show this, consider working in a basis in which $u(t)$ is diagonal with elements $u_1(t), u_2(t), \ldots$. The determinant of $u(t)$ is

$$\det(u(t)) = \prod_i u_i(t).$$

Therefore, the determinant of $u^\dagger(t)$ can be written as

$$\det(u^\dagger(t)) = \prod_i u_i^*(t).$$

Since $u^\dagger(t) = u^{-1}(t)$

$$\prod_i u_i^*(t) = \frac{1}{\prod_i u_i(t)} \Rightarrow \prod_i |u_i(t)|^2 = 1.$$

Then, since $|u_i(t)|^2 = 1$, it follows that the determinant is 1. Therefore, the unitarity of the propagator $u(t)$ is consistent with Liouville's theorem, which states that the volume in phase space is preserved under Hamilton equations.

The Liouville operator in 3.39 can be decomposed in two parts such that

$$i\hat{L} = i\hat{L}_1 + i\hat{L}_2.$$

For this decomposition, the Trotter theorem yields (cf. (Trotter, 1959))

$$e^{i(\hat{L}_1+\hat{L}_2)t} = [e^{i(\hat{L}_1+\hat{L}_2)t/P}]^P = [e^{i\hat{L}_1\frac{\Delta t}{2}}e^{i\hat{L}_2\Delta t}e^{i\hat{L}_1\frac{\Delta t}{2}}]^P + \mathcal{O}(t^3/P^2),$$

where $\Delta t = t/P$. Then, the discrete time propagator can be defined as

$$G(\Delta t) = u_1\left(\frac{\Delta t}{2}\right)u_2(\Delta t)u_1\left(\frac{\Delta t}{2}\right) = e^{i\hat{L}_1\frac{\Delta t}{2}}e^{i\hat{L}_2\Delta t}e^{i\hat{L}_1\frac{\Delta t}{2}}. \tag{3.40}$$

Since the three factors in (3.40) are unitary it is easy to show that $G(t)$ is also unitary and therefore $G^{-1}(t) = G(-t)$. This means that *any integrator based on this Trotter factorization will be reversible*. We note that the Trotter expansion carried out to higher orders will yield higher order integrators.

We consider the following decomposition for the Liouville operator:

$$i\hat{L}_2 = M^{-1}\mathbf{p}\,\frac{\partial}{\partial \mathbf{q}}, \quad i\hat{L}_1 = F(\mathbf{q})\,\frac{\partial}{\partial \mathbf{p}}.$$

It is in agreement with the classical equations of motion (1.3) and the definition of the Liouville operator in (3.39). The decomposition above leads to the propagator

$$G(\Delta t) = \exp\left(F(\mathbf{q})\frac{\partial}{\partial \mathbf{p}}\frac{\Delta t}{2}\right)\exp\left(M^{-1}\mathbf{p}\frac{\partial}{\partial \mathbf{q}}\Delta t\right)\exp\left(F(\mathbf{q})\frac{\partial}{\partial \mathbf{p}}\frac{\Delta t}{2}\right). \tag{3.41}$$

We recall the property that any operator of the form $e^{c\partial/\partial x}$ satisfies

$$e^{c\partial/\partial x}f(x) = f(x+c),$$

where c is independent of x. This identity is a direct consequence of the definition of $e^{c\partial/\partial x}$:

$$e^{c\partial/\partial x}f(x) = \sum_{k=0}^{\infty}\left(c\frac{\partial}{\partial x}\right)^k\frac{f(x)}{k!} = \sum_{k=0}^{\infty}\frac{c^k}{k!}f^{(k)}(x),$$

which can be identified as the Taylor series expansion of $f(x+c)$. Then, it is clear that if we apply (3.41) to $(\mathbf{q}(t), \mathbf{p}(t))$ we obtain the velocity Verlet integrator in (3.5). Thus, we have seen that the Trotter formulation is a way of presenting the velocity Verlet integrator that proves that it is time-reversible. The diagram in Figure 3.1 for velocity Verlet can be adapted to the Trotter formulation as in Figure 3.3. A similar derivation as in (3.41) can be done for the position version of Verlet (Tuckerman, Berne, and Martyna, 1992). However, the original Verlet integrator (3.4) cannot be written using this formalism. In any case, it can be shown that it produces the same trajectories as velocity Verlet does[6]. We will use the Trotter formalism in some parts of this dissertation due to its flexibility to represent more sophisticated splitting schemes and to combine integrators with thermostats.

[6] A proof is suggested in (Tuckerman, Berne, and Martyna, 1992) by induction assuming that the initial condition for the standard Verlet is $\mathbf{q}(0) - \mathbf{q}(-\Delta t) = \mathbf{v}(0) - \frac{\Delta t^2}{2}\frac{F(\mathbf{q}(0))}{M}$.

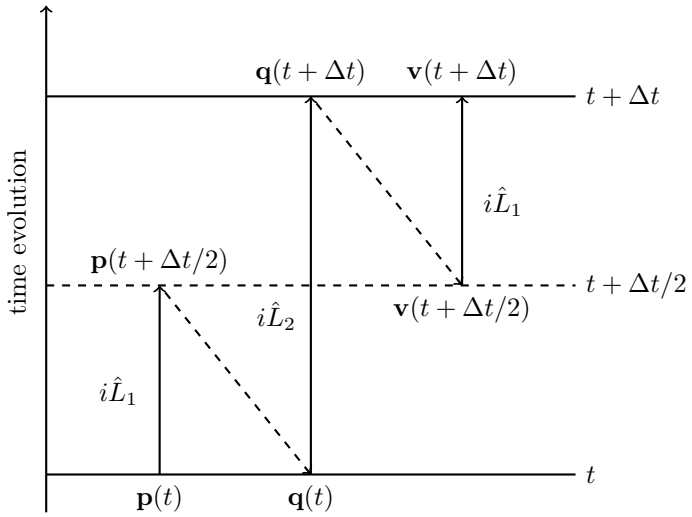


FIGURE 3.3: A step of the velocity Verlet integrator expressed in the Trotter formulation.

Tuckerman, Berne, and Martyna, 1992 and Bussi and Parrinello, 2007 adapted the Trotter formula to a Liouville operator $i\hat{L}$ decomposed in parts such that

$$i\hat{L} = \sum_j i\hat{L}_j.$$

For this decomposition, the Trotter theorem yields

$$e^{i\hat{L}\Delta t} \approx \prod_{j=1}^{M} e^{i\hat{L}_{M+1-j}\frac{\Delta t}{2}} \prod_{k=1}^{M} e^{i\hat{L}_k\frac{\Delta t}{2}}, \tag{3.42}$$

where M is the number of stages in the integrator. Since in general the $i\hat{L}_j$'s do not commute among themselves, the order in which the stages are applied is relevant. The key point here is that the stages $e^{i\hat{L}_l\frac{\Delta t}{2}}$ are chosen so that they can be integrated analytically, and then the Trotter splitting (3.42) is the only source of errors. Since $e^{i\hat{L}_l\frac{\Delta t}{2}}$ are unitary operators, the integrators based on this Trotter factorization will be reversible. It is clear that this notation can be used to represent splitting integrators such as those of (3.21) or (3.25).

3.3 Adaptive Integration Approach (AIA)

By that point, we assumed that parameters of the splitting schemes presented in Section 3.2.2 are predefined and no discussion on specific ways of choosing such parameters or their effect on the overall performance of multi-stage integrators was provided. In this section, we present a new Adaptive Integration Approach (AIA), which, given a molecular simulation problem and a time step Δt, automatically chooses the optimal parameter and therefore the optimal scheme out of an available family of numerical integrators. Thus, for the first time, a system-specific integrator is proposed in molecular simulation.

Although we focus on two-stage splitting integrators from Blanes, Casas, and Sanz-Serna, 2014, the idea may be used with more general families. The system-specific integrating scheme

identified by our approach is optimal in the sense that it provides the best conservation of energy for harmonic forces. For Hybrid Monte Carlo methods (Duane et al., 1987; Horowitz, 1991; Kennedy and Pendleton, 2001), the chosen scheme may be expected to achieve the highest possible acceptance rate in the Metropolis accept-reject test.

The ideas behind the AIA method are presented in this section. We also explain how to extend the algorithm to cases with holonomic constraints. In Section 3.4, we discuss the implementation of AIA in the MultiHMC-GROMACS software package. Section 3.5 presents the benchmarks and testing procedure designed for performance evaluation of the novel adaptive scheme in molecular dynamics and HMC simulations of constrained and unconstrained physical systems. Section 3.6 is devoted to numerical results. The performance of the AIA method is compared with the standard velocity Verlet algorithm, and the two-stage integrators with the fixed parameter values suggested in (Blanes, Casas, and Sanz-Serna, 2014) and in (Predescu et al., 2012). In all experiments and for each of the criteria employed, the performance of AIA is at least as good as, and often significantly better than, the performance of the Verlet scheme and the fixed parameter two-stage integrators. Our conclusions are presented in Section 3.7.

3.3.1 The one-parameter family of two-stage integrators

We consider Hamiltonians H that can be written as a sum $H = A+B$, where A and B are the functions defined in (3.16). The equations of motion associated with H, in this notation, can be written as (3.17). Such equations of motion may be integrated in closed form. In fact, for A the solution is a *drift* in position (3.18) and for B the solution is a momentum *kick* (3.19). The exact solution flows of the partial systems are denoted as ϕ_t^A and ϕ_t^B, respectively.

The integration schemes under study in this Chapter belong to the family of two-stage splitting methods of the form (cf. (Blanes, Casas, and Sanz-Serna, 2014), Section 3.2.2.1)

$$\psi_{\Delta t} = \phi^B_{b\Delta t} \circ \phi^A_{\Delta t/2} \circ \phi^B_{(1-2b)\Delta t} \circ \phi^A_{\Delta t/2} \circ \phi^B_{b\Delta t}. \tag{3.43}$$

Here b is a parameter, $0 < b < 1/2$, that identifies the particular integrator being considered and $\psi_{\Delta t}$ denotes the mapping that advances the numerical solution over one step of length Δt.[7] Note that $\psi_{\Delta t}$ is symplectic as the composition of symplectic mappings and it is time-reversible as a consequence of the palindromic structure of (3.43) (details and references can be found in Section 3.2.2). The transformation $\Psi = \Psi_{\Delta t,L}$ that advances the numerical solution over L steps is given by the composition

$$\Psi = \Psi_{\Delta t,L} = \underbrace{\psi_{\Delta t} \circ \psi_{\Delta t} \circ \cdots \circ \psi_{\Delta t}}_{L \text{ times}}.$$

We recall that, even though ϕ^B appears three times in (3.43), the methods essentially require *two* evaluations of the force $-\nabla_{\mathbf{q}} U$ per step: the evaluation implicit in the leftmost $\phi^B_{b\Delta t}$ in (3.43) at the current step is reused in the rightmost $\phi^B_{b\Delta t}$ at the next step. A fair comparison, in terms of computational cost, between an integration consisting of L steps of length Δt with a method of the form (3.43) and an integration with the standard Verlet

[7] It would be possible to consider *position* integrators obtained by swapping the symbols A and B in (3.43) as explained in Section 3.2.2; however the present study just uses the *velocity* form (3.43).

integrator, uses Verlet with $2L$ steps of length $\Delta t/2$ (which, in view of Verlet being second-order accurate, provides errors that are roughly $1/4$ of those given by Verlet with L steps of length Δt).

3.3.2 Nonadaptive choices of the parameter b

Let us now discuss the possible strategies for choosing the value of b. Regardless of the value of b, the method is second-order accurate, i.e., the size of the error over one step may be bounded by $C\Delta t^3 + \mathcal{O}(\Delta t^5)$, where $C > 0$ varies with b. McLachlan, 1995 pointed out that the minimum error constant C is achieved when $b \approx 0.1932$. This is then the optimal value in the limit $\Delta t \to 0$. In molecular dynamics, simulations with small values of Δt (relatively to the time scales present in the problems) are often unfeasible due to their cost. One may aim to operate with large values of Δt, provided that they are not so large that the integrations become unstable. Unfortunately, the minimum error constant method possesses a short stability interval $(0, 2.55)$[8] and therefore may not be the best choice when Δt is large. The stability of (3.43) is maximized when $b = 1/4$ with a stability interval $(0, 4)$ (see (Blanes, Casas, and Sanz-Serna, 2014) and Section 3.2.2.3 for details). As explained in Section 3.2.2.1, for this value of the parameter, integrations with (3.43) are Verlet integrations with time step $\Delta t/2$, hence, for $b = 1/4$, the stability interval of (3.43) is twice as long as the stability interval $(0, 2)$ of Verlet. In fact, it is well known that among all explicit integrators that use k force evaluations per step, the longest possible stability interval is obtained by concatenating k Verlet substeps each of length $\Delta t/k$ (see Section 3.2.2.3 for details).

In (Blanes, Casas, and Sanz-Serna, 2014) the authors recommend the intermediate value $b \approx 0.2113$. Let us review the ideas leading to this choice, as they will be used in the derivation of the new adaptive approach. Considered in (Blanes, Casas, and Sanz-Serna, 2014) is the use of algorithms of the form (3.43) for Hybrid Monte Carlo and related simulations. The aim is to minimize the energy error (cf. (2.8))

$$\Delta H = H(\Psi_{\Delta t,L}(\mathbf{q}, \mathbf{p})) - H(\mathbf{q}, \mathbf{p}).$$

The analysis in (Blanes, Casas, and Sanz-Serna, 2014) focuses on the model problem where the potential energy is quadratic (harmonic forces), which corresponds to Gaussian probability distributions. With the help of a change of variables, the study of the model problem may be reduced to that of the standard harmonic oscillator in nondimensional variables (standard univariate Gaussian) with the equations of motion (3.29). Assume then that the problem (3.29) is integrated using (3.43) and, as in Section 3.2.2.3, denote by h the nondimensional time step. The expectation or average $\mathbb{E}(\Delta H)$ of the energy error over all possible initial conditions is shown in (Blanes, Casas, and Sanz-Serna, 2014) to possess the bound (see Section 3.2.2.3 for an explanation)

$$0 \leq \mathbb{E}(\Delta H) \leq \rho(h, b),$$

where

$$\rho(h,\, b) = \frac{h^4(2b^2(1/2 - b)h^2 + 4b^2 - 6b + 1)^2}{8(2 - bh^2)(2 - (1/2 - b)h^2)(1 - b(1/2 - b)h^2)}.$$

Thus, choices of b and h that lead to a small value of ρ will result in small energy errors for (3.29).

[8] Through this dissertation, a stability interval is always defined using dimensionless time.

It is understood that $\rho = \infty$ for combinations of b and h leading to a denominator ≤ 0; these combinations correspond to unstable integrations. It agrees with what we observed before in the definition (3.38) (Section 3.2.2.3), which leads to the equivalence between stability and the positivity of the denominator. From (Blanes, Casas, and Sanz-Serna, 2014), we get the additional restriction $b \in (0,\, 1/2)$ which helps avoiding too big errors and too small stability intervals. Thus, it is easy to see that the stability interval for two-stage integrators is

$$0 < h < \min\left\{\sqrt{2/b},\ \sqrt{2/(1/2-b)}\right\},$$

which depends on the choice of b. The study of the function ρ is *more discriminating* than the study of the stability interval of the integrators: it is possible for two integrators to share a common stability interval and yet have very different values of ρ for a given value of h that is stable for both of them.

Let us now move from the scalar oscillator (3.29) to multidimensional linear oscillatory problems integrated with time step Δt and denote by ω_j, $j = 1, 2, \ldots$, the corresponding angular frequencies (the periods are then $T_j = 2\pi/\omega_j$). By superposing the different modes of the solution, one sees that if the (nondimensional) quantities $h_j = \omega_j \Delta t = 2\pi \Delta t/T_j$ are such that, as j varies, all the values $\rho(h_j, b)$ are small, then the energy errors will also be small. In (Blanes, Casas, and Sanz-Serna, 2014), the authors aimed to identify *one* value of b that would result in small values of $\rho(h,\, b)$ over a meaningful range of values of h. More precisely, the recommended $b = 0.2113$ was found by minimizing the function of b given by

$$\max_{0<h<2} \rho(h, b). \tag{3.44}$$

The range $0 < h < 2$ was chosen because, for the test problems considered, the standard Verlet method was found to perform well for $0 < \omega_j \Delta t = 2\pi \Delta t/T_j < 1$ (which is half the maximum allowed by the Verlet linear stability interval $(0, 2)$). As we emphasized already, (3.43) uses two force evaluations per step and Verlet only one. Thus, for (3.43) to be an improvement on standard Verlet, it must be demanded that it works well for twice as long values of Δt, i.e., for $0 < \omega_j \Delta t = 2\pi \Delta t/T_j < 2$. The values of the function ρ for the integrator with $b = 0.2113$ are compared to those for velocity Verlet in Figure 3.4 (left). One can observe that for the $(0, 2)$ range of time steps the values of ρ provided by the two-stage integrator (3.44) are smaller than those of Verlet. From now on, we will call BCSS this integrator derived from (3.44) in (Blanes, Casas, and Sanz-Serna, 2014).

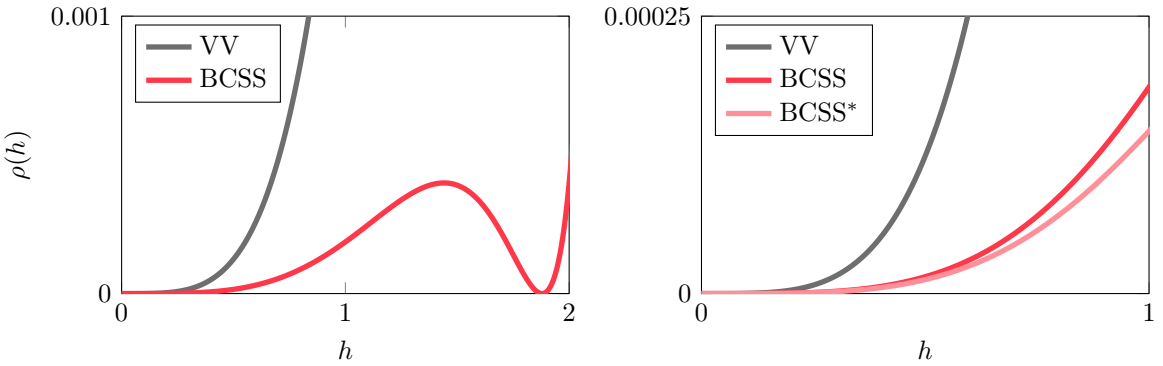


FIGURE 3.4: Comparison of the expected energy error bound of the two-stage integrator with $b = 0.2113$ and the classic velocity Verlet for the interval of time steps h between zero and two (left) and for h between zero and one with a modified version of BCSS (right).

Numerical tests in (Blanes, Casas, and Sanz-Serna, 2014) show the merit of the choice $b = 0.2113$. However the fact remains that, if, for a given problem and Δt, the maximum of $\omega_j \Delta t = 2\pi \Delta t / T_j$ as j varies is significantly smaller than 2, i.e., the chosen Δt is relatively small, then a smaller value of b would provide a better integrator. On the other hand, if that maximum is significantly larger than 2, then it would be advisable to increase b. For instance, in Figure 3.4 (right) the integrator BCSS* has been obtained by arbitrary choosing the parameter $b^* = 0.21$. By reducing an interval for h to $0 < h < 1$ one can observe how for time steps this choice leads to a more accurate integrator than BCSS. Clearly, the function ρ is very sensitive to even small changes in b. Using this observation, we propose a different approach. Rather than choosing a single value of b that is later applied in all simulations, we suggest an algorithm that, once the system to be integrated has been specified and the user has chosen a value of Δt, identifies the "best" b.

3.3.3 Adapting the integrator to the problem

Although the real physical systems that one wishes to simulate in practice are very complex, it is helpful to consider the case where the forces are two-body interactions. Note that the most stringent stability restrictions on Δt are likely to stem from stiff two-body forces, in particular from pairs of bonded atoms. For relatively small energy values, those stiff forces may be assumed to be harmonic.

For two particles attracting each other harmonically, the period of the oscillations is

$$T = 2\pi \sqrt{\frac{\mu}{k}}, \qquad \mu = \frac{m_1 m_2}{m_1 + m_2}, \tag{3.45}$$

where m_1, m_2 are the masses of the particles, μ the reduced mass and k the force constant.

The stability of the integration is of course determined by the highest frequency $\tilde{\omega}$ or, equivalently, the smallest period $\tilde{T}$ present in the system. For the standard Verlet integrator, the linear stability restriction is, as noted above,

$$\Delta t < \frac{2}{\tilde{\omega}} = \frac{\tilde{T}}{\pi}. \tag{3.46}$$

Due to nonlinear effects, including nonlinear resonances, and to other difficulties (see (Sanz-Serna, 1991; Mandziuk and Schlick, 1995; Schlick et al., 1998; Skeel, 1999; Schlick, 2002) and the example in Section 3.2.1.1 for more details), this requirement may be too weak to ensure stability in practice. Some authors suggest that the stability restriction for the Verlet integrator

$$\Delta t < \frac{\sqrt{2}}{\tilde{\omega}} = \frac{\tilde{T}}{\sqrt{2}\pi} \tag{3.47}$$

is more realistic in applications than (3.46) (Mazur, 1997). Note that moving from (3.46) to (3.47) may be seen as the result of multiplying the smallest period by a safety factor $1/\sqrt{2}$ (equivalently multiplying the frequency by $\sqrt{2}$). One can readily recognize the non-linear stability condition in the case of the fourth-order resonance (see Table 3.1).

In our adaptive method, if Δt is the time step attempted by the user, we exploit the stability restriction in (3.47) to form, similarly to the preceding section, the nondimensional quantity

$$\bar{h} = \sqrt{2}\tilde{\omega}\Delta t = \sqrt{2}\frac{2\pi}{\tilde{T}}\Delta t \tag{3.48}$$

and determine b so as to minimize (cf. (3.44))

$$\max_{0<h<\bar{h}} \rho(h, b). \tag{3.49}$$

Here the function ρ that bounds the energy error is minimized in the *shortest interval* $(0, \bar{h})$ that contains all the values $\sqrt{2}\omega_j\Delta t$, where ω_j are the frequencies in the problem being integrated. Let us illustrate how this works. If the user attempts a value of Δt slightly smaller than $\sqrt{2}\tilde{T}/\pi$, then $\bar{h}$ will be just below 4 and the minimization of (3.49) will lead to b close to 0.2500. For this value of b, I steps of length Δt are, as discussed above, equivalent to $2I$ steps of length $\Delta t = \tilde{T}/\sqrt{2}\pi$ of the velocity Verlet algorithm; in other words, the adaptive algorithm will run the optimally stable Verlet with the maximum Δt allowed by (3.47). As the value of Δt attempted by the user decreases from $\sqrt{2}\tilde{T}/\pi$ towards 0, the value of b will decrease from 0.2500 to McLachlan's 0.1932, thus improving the error constant. The length of the stability interval will shrink as b is decreased, but this will cause no problem because by construction all values $\omega_j\Delta t$ will fall in the stability interval (in fact, for safety, even the larger $\sqrt{2}\omega_j\Delta t$ will lie on the stability interval). Finally, if $\Delta t \geq \sqrt{2}\tilde{T}/\pi$, the quantity (3.49) will be ∞ for all values of b; this indicates that Δt is too large for the problem at hand.

3.3.4 Algorithm

Given a physical system and a value of Δt, AIA determines the value of the parameter b to be used in (3.43) as follows:

1. Use equation (3.45) to find the periods or frequencies of all two-body interactions in the system. Determine the minimum period $\tilde{T}$ and compute the nondimensional quantity $\bar{h}$ in (3.48).

2. Check whether $\bar{h} < 4$. If not, there is no value of b for which the scheme (3.43) is stable for the attempted time step Δt and the integration is aborted. In other case go to the next step.

3. Find the optimal value of the parameter b by minimizing (3.49) with the help of an optimization routine.

3.3.5 Extension to constrained dynamics

Holonomic constraints $g(\mathbf{q}) = 0$ allow the use of bigger time steps in the simulation of physical systems that contain high-frequency modes. By freezing those modes, it is possible to bypass the demanding restriction they would otherwise impose on the time step. SHAKE (Ryckaert, Ciccotti, and Berendsen, 1977) and its velocity extension RATTLE (Andersen, 1983) are widely used algorithms in this connection. We focus our attention on RATTLE which is symplectic and time-reversible and thus is appropriate for being used with Hybrid Monte Carlo methods (Leimkuhler and Skeel, 1994). Now we show how, by following the idea behind the original RATTLE, two-stage integrators of the family (3.43) may be applied to problems with constraints. In this way, the Adaptive Integration Approach may be extended to the constrained case.

The constrained equations of motion corresponding to (3.17) are

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q} &= M^{-1}\mathbf{p}, \\ \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{p} &= -\nabla_{\mathbf{q}} U(\mathbf{q}) + g'(\mathbf{q})^T \lambda, \\ g(\mathbf{q}) &= 0, \end{aligned}$$

where λ is the vector of Lagrange multipliers and $g'(\mathbf{q})^T\lambda$ represents the forces exerted by the constrains. The holonomic constraint implies, by differentiation with respect to time, a constraint on the velocities $(\mathrm{d}/\mathrm{d}t)\mathbf{q} = M^{-1}\mathbf{p}$:

$$g'(\mathbf{q})\ M^{-1}\ \mathbf{p} = 0.$$

As in (3.26), we divide one step into two half steps. For any time t, the equations for the first half step are

$$\begin{aligned} \mathbf{p}(t+bh) &= \mathbf{p}(t) - bh\ \nabla_{\mathbf{q}} U(\mathbf{q}(t)) + bh\ g'(\mathbf{q}(t))^T \lambda_t, \\ \mathbf{q}(t+h/2) &= \mathbf{q}(t) + \frac{h}{2} M^{-1}\mathbf{p}(t+bh), \end{aligned} \tag{3.50}$$

where the Lagrange multiplier λ_t is chosen to ensure

$$g(\mathbf{q}(t+h/2)) = 0,$$

and

$$\mathbf{p}(t+h/2) = \mathbf{p}(t+bh) - \left(\frac{1}{2} - b\right) h \nabla_{\mathbf{q}} U(\mathbf{q}(t+h/2)) + \left(\frac{1}{2} - b\right) h\ g'(\mathbf{q}(t+h/2))^T \lambda^{(v)}_{t+h/2}, \tag{3.51}$$

where the velocity Lagrange multiplier $\lambda^{(v)}_{t+h/2}$ is chosen so that

$$g'(\mathbf{q}(t+h/2))\ M^{-1}\ \mathbf{p}(t+h/2) = 0.$$

The equations for the second half step $(\mathbf{q}(t+h/2), \mathbf{p}(t+h/2)) \to (\mathbf{q}(t+h), \mathbf{p}(t+h))$ are similar. It is easy to see that, if we define $s := t + h/2$, then the second half step is written as $(\mathbf{q}(s), \mathbf{p}(s)) \to (\mathbf{q}(s+h/2), \mathbf{p}(s+h/2))$.

The proof of the symplecticness of RATTLE given by Leimkuhler and Skeel, 1994 may

be easily adapted to prove that each half step, $(\mathbf{q}(t), \mathbf{p}(t)) \to (\mathbf{q}(t+h/2), \mathbf{p}(t+h/2))$ and $(\mathbf{q}(t+h/2), \mathbf{p}(t+h/2)) \to (\mathbf{q}(t+h), \mathbf{p}(t+h))$, is symplectic. The proof consists in showing that the solutions generated by RATTLE at mesh points preserve the wedge product. The derivation for the first half step is shown here. From (3.51),

$$\begin{aligned} d\mathbf{q}(t+h/2) \wedge d\mathbf{p}(t+h/2) = d\mathbf{q}(t+h/2) \wedge d\mathbf{p}(t+bh) \\ - \left(\frac{1}{2} - b\right) h \; d\mathbf{q}(t+h/2) \wedge d\nabla_{\mathbf{q}} U(\mathbf{q}(t+h/2)) \\ + \left(\frac{1}{2} - b\right) h \; d\mathbf{q}(t+h/2) \wedge dg'(\mathbf{q}(t+h/2))^T \lambda^{(v)}_{t+h/2}. \end{aligned} \tag{3.52}$$

If we denote the Hessian of the potential energy as U'', then $d\nabla_{\mathbf{q}} U(\mathbf{q}) = U''(\mathbf{q}) \; d\mathbf{q}$. Thus, the equality in (3.52) can be rewritten as

$$\begin{aligned} d\mathbf{q}(t+h/2) \wedge d\mathbf{p}(t+h/2) = d\mathbf{q}(t+h/2) \wedge d\mathbf{p}(t+bh) \\ - \left(\frac{1}{2} - b\right) h \; d\mathbf{q}(t+h/2) \wedge U''(\mathbf{q}(t+h/2)) \; d\mathbf{q}(t+h/2) \\ + \left(\frac{1}{2} - b\right) h \; d\mathbf{q}(t+h/2) \wedge dg'(\mathbf{q}(t+h/2))^T \lambda^{(v)}_{t+h/2}. \end{aligned} \tag{3.53}$$

Now we use two technical results from (Leimkuhler and Skeel, 1994):

- Let du be an arbitrary differential in $\mathbb{R}^n$ and let A be any $n \times n$ real symmetric matrix, then $du \wedge (Adu) = 0$.
- Let τ be an arbitrary time, then $d\mathbf{q}(\tau) \wedge d(g'(\mathbf{q}(\tau))^T \lambda_\tau) = 0$.

Applying these results to (3.53) we get

$$d\mathbf{q}(t+h/2) \wedge d\mathbf{p}(t+h/2) = d\mathbf{q}(t+h/2) \wedge d\mathbf{p}(t+bh).$$

Then, from (3.50), we get

$$\begin{aligned} d\mathbf{q}(t+h/2) \wedge d\mathbf{p}(t+bh) &= \left(d\mathbf{q}(t) + \frac{h}{2} M^{-1} d\mathbf{p}(t+bh)\right) \wedge d\mathbf{p}(t+bh) \\ &= d\mathbf{q}(t) \wedge d\mathbf{p}(t+bh) \\ &= d\mathbf{q}(t) \wedge (\mathbf{p}(t) - bh \; d\nabla_{\mathbf{q}} U(\mathbf{q}(t)) + bh \; dg'(\mathbf{q}(t))^T \lambda_t). \end{aligned}$$

Applying again the same results as above, we obtain that the wedge product is preserved

$$d\mathbf{q}(t+h/2) \wedge d\mathbf{p}(t+h/2) = d\mathbf{q}(t) \wedge d\mathbf{p}(t).$$

It is straightforward to proof the equivalent property for the second stage of the integrator. Therefore, the whole step $(\mathbf{q}(t), \mathbf{p}(t)) \to (\mathbf{q}(t+h), \mathbf{p}(t+h))$ is also symplectic.

It is clear that $2I$ steps of length $h/2$ of the Verlet integrator supplemented with the constraining technique envisaged here are as expensive as I steps of length h of the extension of two-stage schemes to constrained dynamics we have just described.

Hybrid Monte Carlo methods can be easily used in constrained dynamics. Only one consideration has to be made: right after the Metropolis test (step 4 in Algorithm 1 and

step 4 in Algorithm 2), when the momenta $\mathbf{p}^*$ are resampled, the constraint $g'(\mathbf{q})M^{-1}\mathbf{p}^* = 0$ has to be fulfilled. Further details can be found in (Hartmann, 2008).

3.4 Implementation

AIA has been implemented in the MultiHMC-GROMACS software code, an in-house modified version of GROMACS (Berendsen, van der Spoel, and van Drunen, 1995; Hess et al., 2008), which is a popular software package for molecular dynamics simulations. MultiHMC-GROMACS has been developed to achieve better accuracy and sampling performance in GROMACS through the use of Hybrid Monte Carlo methods and multi-stage numerical integrators. The detailed description of the package can be found in Chapter 7. Here we just summarize the features related to AIA implementation.

AIA has been implemented in the GROMACS preprocessing module, `grompp`, which has to be run once before simulating and thus does not introduce extra computational costs in the simulation itself (see Figure 7.1).

In the original GROMACS code, the module `grompp` reads the GROMACS input files and processes them for further use in the molecular dynamics module, `mdrun`. It also checks input data and, if necessary, generates warnings that allow the users to reconsider their chosen setup. For example, the input time step Δt is inspected for its ability to provide a stable numerical integration in molecular dynamics. This check is implemented in the `check_bonds_timestep()` routine and consists of two main steps. First, for each pair of bonded particles, the corresponding period T is calculated with the help of (3.45). Then, for the given Δt and T, the Verlet stability condition $5\Delta t < T$ (see (Mazur, 1997) and Appendix A.2 for details) is checked. If the condition does not hold, an error message is issued and the simulation is not allowed. It is easy to see that this restriction is in agreement with condition (3.47) since $1/(\sqrt{2}\pi) \approx 1/5$. Otherwise, if $10\Delta t \geq T$ the code issues a message warning that instabilities may arise and recommending to decrease Δt or to use a constrained algorithm (Mazur, 1997). Once a warning or error message appears, the search for further problematic oscillations stops.

For our purposes, we modified this part of the code in such a way that the search continues until the period of the fastest oscillation $\tilde{T}$ is found. Its value is used to define $\bar{h}$ in (3.48). Then the optimal parameter value b is calculated using (3.49). A particle swarm optimization algorithm driven by a golden section search (Oh and Hori, 2006) is used to perform the required minimization. The parameter b is stored in the *input record* structure of GROMACS so that it can be accessed from every routine in the package after running the `grompp` preprocessing module.

In standard GROMACS, molecular dynamics simulations are performed with the `mdrun` module using the input file *.tpr* generated by `grompp`. The velocity Verlet integrator is implemented in the `update_coords(·)` function, which is called from `do_md()` sequentially to update velocities, positions and velocities again. The procedure is repeated as many times as desired. More details can be found in Section 7.4.

The integrators resulting from the Adaptive Integration Approach described above belong to the family (3.43) and thus are naturally included in the list of integrators implemented in MultiHMC-GROMACS (see Section 7.4 for details). The parameter `integrator` used in the *.mdp* file is `aia`.

It is useful to present a multi-stage scheme in kick/drift factorization form to efficiently implement multi-stage integrators in the GROMACS package (Pronk et al., 2013). For example, two-stage integrators are best rewritten in the form (3.26), which is more suitable for its implementation inside the `mdrun` module in GROMACS (details can be found in Section 7.4.2). The scheme can be implemented with six evaluations of the `update_coords()` function, alternating velocity and position updates in which modified parameters such as b, $1/2$ and $1/2 - b$ are used. With our implementation, multi-stage integrators have computational costs equal to those of the standard Verlet method, provided that the latter is run with the choice of time step that equalizes the number of force evaluations.

For simulations of constrained dynamics, we use the SHAKE algorithm as implemented in the released version of GROMACS. This implementation relies on the original approach of Ryckaert, Ciccotti, and Berendsen, 1977, combined with the Lagrange multipliers procedure of Lippert et al., 2007 for improving the accuracy in the calculation of velocities of constrained particles (Hess et al., 2008). The implementation of the RATTLE step in GROMACS is done following the algorithm in (Andersen, 1983). The modifications explained in Section 3.3.5 for the two-stage integrators for constrained dynamics are combined with the implementation of the released version of GROMACS. Any further developments regarding performance, parallelization or formulation have not been considered so far.

The flowchart in Figure 3.5 summarizes the AIA implementation.

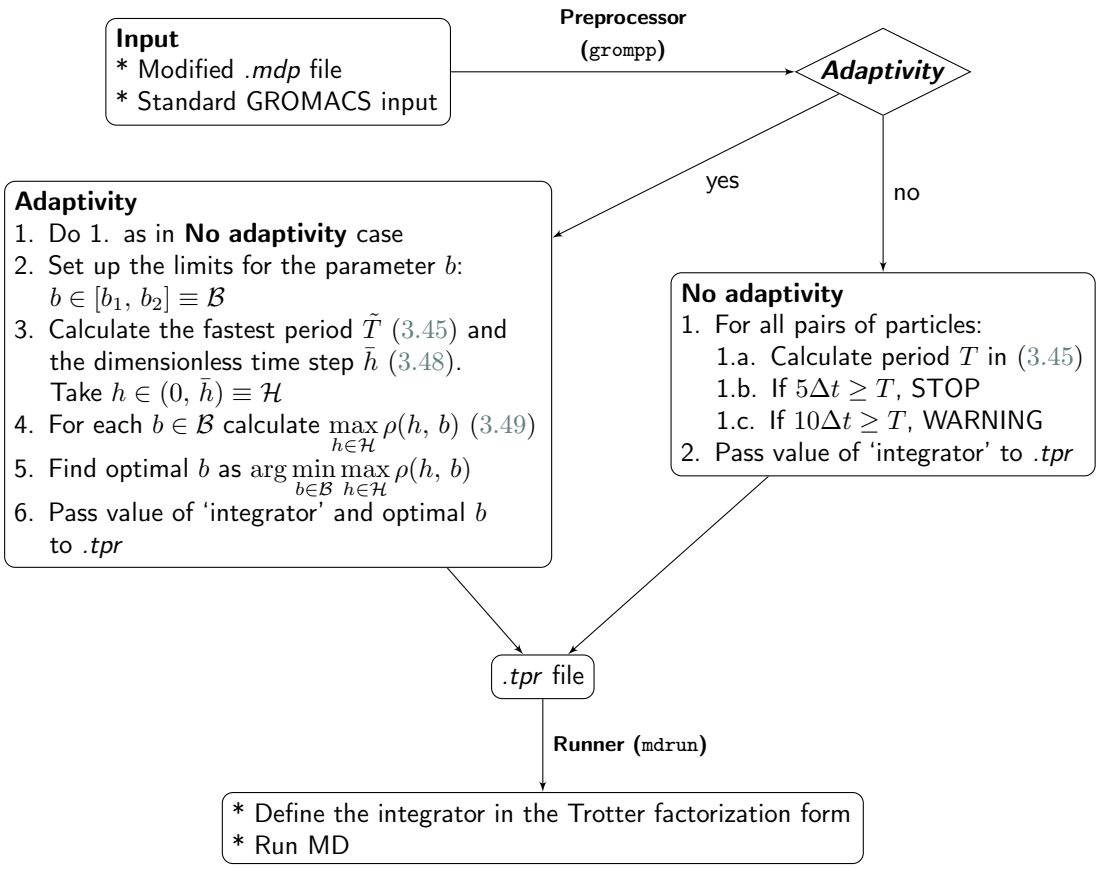


FIGURE 3.5: Flowchart of the Adaptive Integration Approach (AIA) as implemented in MultiHMC-GROMACS.

3.5 Numerical experiments

3.5.1 Testing procedure

In order to evaluate the efficiency of the proposed AIA scheme, we compared it in accuracy and performance with the velocity Verlet integrator and with the two-stage integrator (BCSS) of Blanes, Casas, and Sanz-Serna, 2014. In addition, some selected tests also involved the two-stage HOH scheme by Predescu et al., 2012.

All tests probing various integrating schemes have been repeated with three different simulation techniques, MD combined with the v-rescale thermostat, HMC and GHMC. We omit here the data obtained with GHMC for two reasons. First, as expected, HMC and GHMC showed very similar behavioral trends. On the other hand, the GHMC method possesses an extra parameter that needs to be tuned properly to guarantee optimal performance. Such tuning is likely to be time-consuming and was not attempted. We, therefore, decided to avoid reporting data that may not correspond to the best possible performance of GHMC.

The following points have been taken into account to ensure a comparison as clear as possible.

As we have repeatedly explained (see Section 3.3 for details), whenever a two-stage splitting scheme (AIA or not) and velocity Verlet are used on the same problem, the comparisons reported here are fair (in computational cost terms). Verlet is run with half the time step and a double number of steps.

In Hybrid Monte Carlo (HMC and GHMC) simulations, the number of Metropolis tests was also kept constant regardless of the acceptance rate achieved. For two-stage integrators, the number of MD time steps between two successive Monte Carlo tests was chosen half of the corresponding number for Verlet.

A broad range of time steps has been tested for two benchmark systems with the aim of observing the dependence of the optimal parameter b in AIA on the value of Δt. Different lengths of MD trajectories in HMC simulations were also explored. Each test has been repeated 10 times for unconstrained dynamics and 15 times for constrained dynamics and every single point in the reported data here was obtained by averaging over the multiple runs to reduce statistical errors.

3.5.2 Benchmarks and Simulation setup

Two test systems were chosen for the numerical experiments: one describes the non-constrained coarse-grained VSTx1 toxin in a POPC bilayer (Jung et al., 2005) and the other the constrained atomistic 35-residue villin headpiece protein subdomain (Bazari et al., 1988; McKnight, Matsudaira, and Kim, 1997). We will refer to these systems as toxin and villin, respectively.

In the coarse-grained toxin system, four heavy particles on average were represented as one sphere (Wallace and Sansom, 2007; Shih et al., 2006), which produced a total number of 7810 particles. For both Coulomb and van der Waals interactions the shift algorithm was used (van der Spoel and van Maaren, 2006). Both potential energies were shifted to 0 $\mathrm{kJ\,mol^{-1}}$ at a radius of 1.2 nm. Periodic boundary conditions were considered in all directions. No constraint algorithm was applied to this system. The total length of all simulation runs was 20 ns, which was sufficient, with stable time steps, for a complete equilibration of the system.

The villin protein was composed of 389 atoms and the system was solvated with 3000 water molecules. Coulomb interactions were solved with the PME algorithm of order 6 (Darden, York, and Pedersen, 1993; Essmann et al., 1995) and van der Waals interactions were

considered as in the toxin system, with the only difference of a radius of 0.8 nm. Periodic boundary conditions were again defined in all directions. The bonds involving hydrogens were constrained. Instead of constraining all atoms, as it is commonly suggested in the literature (see (van der Spoel and Lindahl, 2003) for instance), we have only constrained the hydrogens, because it is the only case that allows the integration algorithm to perform in parallel with domain decomposition (Hess et al., 2008). Constraining only the hydrogen atoms does not affect the accuracy of the simulation but allows bigger time steps for the integration. Since villin system is an atomistic model, simulations are expected to be slower than for the coarse-grained toxin. However, an exhaustive study of the complete folding process of the villin protein is out of the scope of this work. Thus, with the available computational resources, simulations were run only to observe the effect of the AIA on accuracy and performance of a constrained atomistic system. It has to be also remarked that there are examples in the literature of similar tests for which a weak coupling thermostat and a barostat were used to have more realistic results (van der Spoel and Lindahl, 2003). Barostats are not considered in this study since the aim is to compare the performance of the AIA scheme with that of velocity Verlet when both integrators sample in the NVT ensemble. The total length of all experiments performed for this system was 5 ns.

The temperature in MD simulations was controlled by the standard v-rescale algorithm for both benchmarks. The reference temperatures were 310 K for toxin and 300 K for villin. The same temperatures were used in HMC and GHMC. No thermostat is required in HMC or GHMC simulations.

3.6 Results

We stress that throughout this section the different setups used for the simulation will be expressed in terms of parameters appropriate for the velocity Verlet (one-stage) integrator. This implies that for two-stage schemes the time steps are doubled and the trajectory lengths are halved which guarantees the fair comparison between these integrators. For improving the readability, all the plots have been created following the same criteria.

3.6.1 Unconstrained system

We first present the results of the unconstrained test system.

The tests were run using the following set of time steps for the Verlet integrator {10 fs, 15 fs, 20 fs, 22.5 fs, 25 fs} (recall that for two-stage integrators these values are doubled). Two different number of steps in the MD trajectories, L, have been tested in the HMC experiments for each Δt. In the case of velocity Verlet, the values of L were 2000 and 4000 for all Δt except when $\Delta t = 25$ fs, where L was chosen to be 1000 and 2000. The corresponding values of L for the two-stage schemes are, as pointed out repeatedly above, halved. The acceptance rates that appear in Figure 3.6 were obtained by averaging over all experiments with the same Δt, regardless of the choice of L.

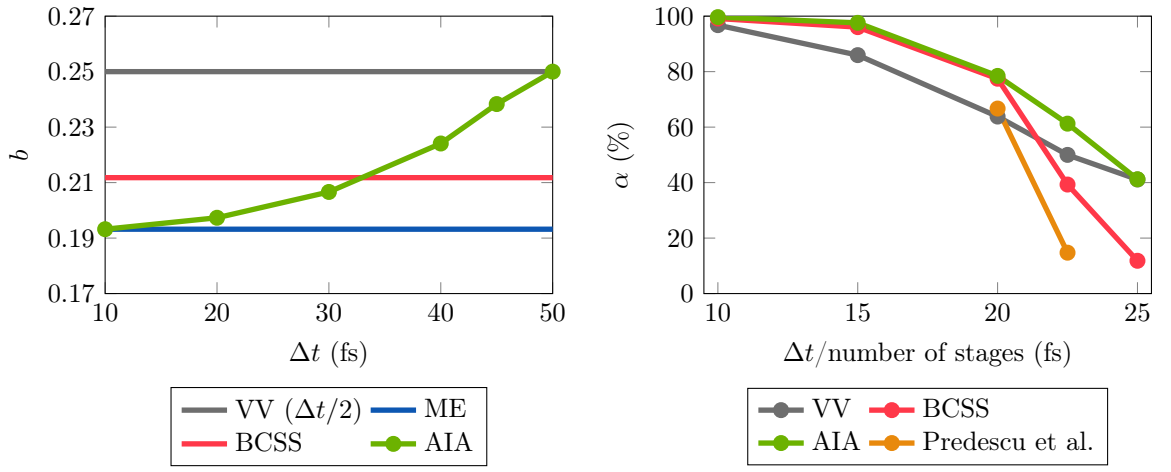


FIGURE 3.6: Toxin. Dependence of the parameter b on the choice of Δt (left) and its effect on resulting acceptance rates in HMC simulations (right). "Number of stages" as appears in x-axis label refers to 1 for velocity Verlet and 2 for all two-stage integrators.

As stated earlier, AIA finds, for a given physical system and a chosen time step, the unique value of the parameter b in (3.43) that provides the best energy conservation achievable with the members of the family (3.43). Figure 3.6 presents the parameter b determined by AIA, as a function of Δt, for simulations of toxin and compares them with the ones previously identified for different two-stage integrating schemes. As it was intended, for small Δt, AIA chooses McLachlan's minimum error constant method, and, as Δt increases, b approaches 0.2500, a value which, as discussed in Section 3.3, essentially yields the Verlet integrator. The two-stage integrator BCSS is the optimal choice for time steps roughly twice smaller than the stability limit of the velocity Verlet integrator.

We then investigated the effect of the AIA on the performance of HMC simulations by monitoring acceptance rates as functions of Δt with different two-stage integrators. Conservation of energy has a direct impact on acceptance or rejection in the Metropolis test of the Hybrid Monte Carlo methods: the better the energy is preserved, the more proposed trajectories are accepted (Beskos et al., 2013). Thus, by design, AIA has to provide, at least for Gaussian distributions, the highest acceptance rates for any choice of Δt. This is demonstrated in Figure 3.6. The two-stage schemes of (Blanes, Casas, and Sanz-Serna, 2014) and (Predescu et al., 2012) ensure higher acceptance rates than velocity Verlet for time steps significantly smaller than the Verlet stability limit. However, the performance of those two-stage schemes drops dramatically for larger time steps. AIA yields acceptance rates that are as good as those of BCSS when Δt is small and as good as those of Verlet near the Verlet stability limit. In particular, AIA does not yield worse results than Verlet for any values of Δt. The trend observed for the HMC method as shown in Figure 3.6 was also apparent in GHMC tests.

To compare the impact of different integrating schemes on the accuracy of HMC and MD simulations, we calculated averages for two thermodynamic observables: the temperature T and the distance d traveled by the toxin from the center of the membrane to the preferred location at the surface of the membrane. The expected average values of the distance are around ∼2.48 nm (Jung et al., 2005; Wee et al., 2008), whereas the target temperature was chosen to be 310 K. The performed simulations had a fixed total length of 20 ns, which was

long enough for equilibrating the system if stable time steps were used, but not sufficient for obtaining accurate averages. So, the tests are meaningful for observing trends rather than getting good production results. For HMC we found more informative to plot the RMSD between the target temperature and the observed temperatures rather than the average temperatures themselves. For MD simulations the overall fluctuations are smaller and the trends for averages, even in short simulations, are clearer than in HMC simulations. Thus, we plot the temperatures.

Figure 3.7 and Figure 3.8 summarize the averages for the two observables, distance and temperature. From now on, we plot the properties obtained with HMC simulations versus the product $\Delta t \times L$ of the time step and the number of steps in an MD trajectory. This is due to the important role this product plays in the overall acceptance rate and the correlation in HMC simulations as it has been studied for instance in (Bou-Rabee and Sanz-Serna, 2017b).

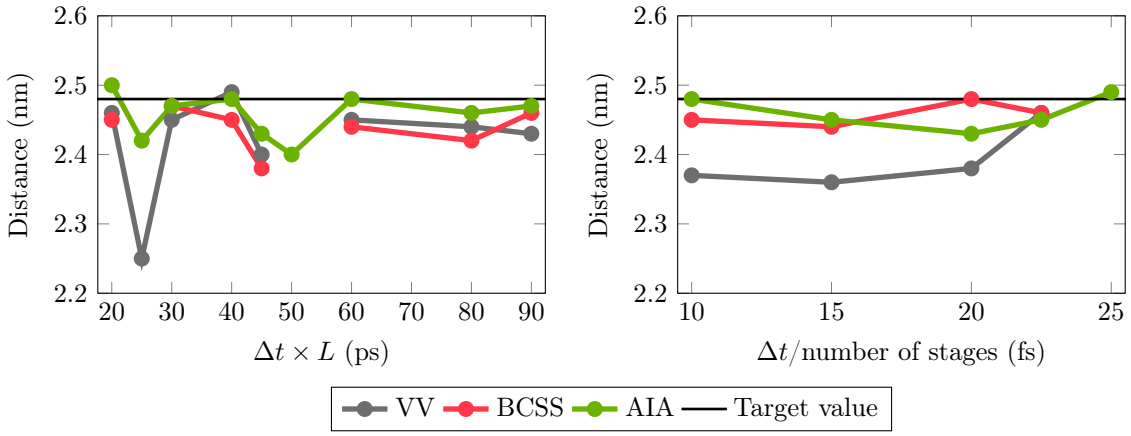


FIGURE 3.7: Toxin. Distance between the c.o.m. of the toxin and the c.o.m. of the bilayer (expected to be ∼2.48 nm) predicted by HMC simulations with different lengths of trajectories L, time steps Δt and integrating schemes (left), and by MD simulations using various time steps Δt and integrators (right).

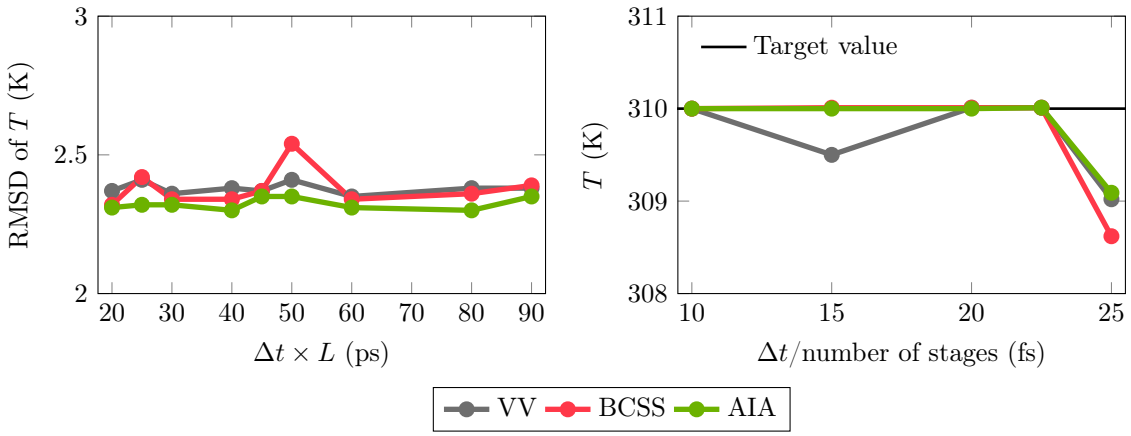


FIGURE 3.8: Toxin. Temperature RMSD with respect to the target temperature observed in HMC simulations with different lengths of trajectories L, time steps Δt and integrating schemes (left), and the average temperature in MD simulations using various time steps Δt and integrators (right). The target temperature was set to 310 K. The v-rescale thermostat was applied in MD.

As follows from Figure 3.7 and Figure 3.8, for both properties, d and T, the accuracy of AIA is comparable to but typically better than, the accuracy provided by BCSS and velocity Verlet for time steps distant from the Verlet stability limit. However near the stability limit the accuracy of all integrators decreases – more dramatically for BCSS and less noticeably for AIA. Interestingly, longer MD trajectories ($L = 2000$) in HMC allow AIA to be accurate at such large values of Δt (see Figure 3.7 at $\Delta t \times L = 50$ ps). In contrast, the accuracy in simulations with BCSS and Verlet is rather sensitive to the choice of Δt. The former failed to produce meaningful averages for $\Delta t = 25$ fs. Less dramatic differences but similar trends were observed for molecular dynamics simulations (right panels of Figure 3.7 and Figure 3.8).

Finally, we inspected the role of numerical integrators in the sampling efficiency of HMC and MD simulations.

In Figure 3.9 the distance d between the c.o.m. of the toxin and the c.o.m. of the bilayer is shown as a function of time for a single choice of the time step $\Delta t = 15$ fs and the trajectory length $L = 4000$ in HMC, and for $\Delta t = 10$ fs in MD. The superiority of the AIA method is clearly demonstrated in both HMC and MD since AIA makes the toxin reach the target destination earlier than the rest of the integration schemes do.

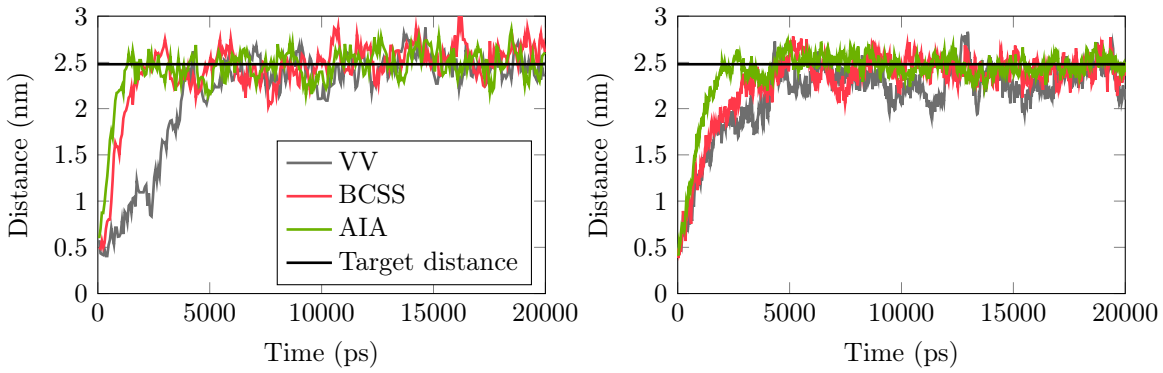


FIGURE 3.9: Toxin. Distance between the c.o.m. of the toxin and the c.o.m. of the bilayer as a function of time obtained in HMC simulations with time step $\Delta t = 15$ fs, trajectory length $L = 4000$ and different integrators (left) and in MD simulations with time step $\Delta t = 10$ fs and the same integrators (right). The expected value is ∼2.48 nm.

Figure 3.10 presents the distributions of the distances d collected from simulations with different integrators (AIA, VV and BCSS) and compares them with the "true" distribution obtained from the HMC simulation with velocity Verlet at $\Delta t = 15$ fs and $L = 4000$ of 200 ns length (ten times longer than the other ones). It can be seen that AIA samples more closely to this distribution. As for all tests in this section, the plotted data are resulted from averaging over several repetitive runs (see Section 3.5 for more details).

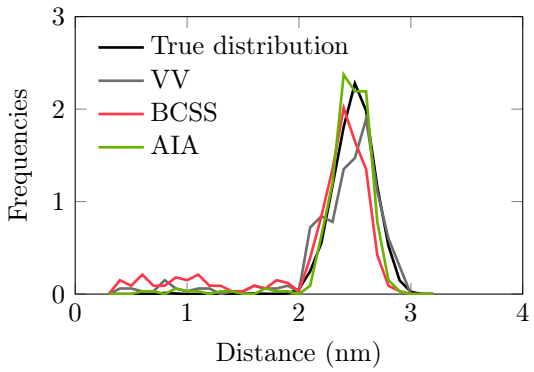


FIGURE 3.10: Toxin. Distribution of the distances between the c.o.m. of the toxin and the c.o.m. of the bilayer observed in HMC simulations of 20 ns length with time step $\Delta t = 15$ fs, trajectory length $L = 4000$ using different integrators. The solid black line presents the "true" distribution produced with a ten times longer simulation (200 ns) that used the same input. The y-axis presents frequencies which are calculated as the normalized numbers of hits registered for a distance bin within a simulation. Here normalization is performed with respect to a product of a total number of samples and the size of a distance bin (0.1 in this particular case).

Finally, the integrated autocorrelation function IACF of the drift of the toxin to the preferred interfacial location was measured during the equilibration stage of the simulations for the range of time steps and trajectory lengths. The autocorrelation function (ACF) is a commonly used tool for evaluating sampling efficiency in molecular dynamics simulations

(Allen and Tildesley, 1989; Kennedy and Pendleton, 2001), statistics and other fields. For a certain property f depending on time, it is defined as

$$\mathrm{ACF}(f(t)) = \langle (f(\xi) - \hat{f})(f(\xi + t) - \hat{f}) \rangle_{\xi},$$

where $\hat{f}$ is the mean value of the observable f. For simplicity in the calculations, the values of f are normalized as $\tilde{f}(t) = f(t) - \hat{f}$ for all time t. Then, in practice, correlation functions are calculated based on data points with discrete time intervals Δt, so that the ACF from an MD simulation is:

$$\mathrm{ACF}(f(j\Delta t)) = \frac{1}{N-j} \sum_{i=0}^{N-1-j} \tilde{f}(i\Delta t)\tilde{f}((i+j)\Delta t),$$

where N is the number of available samples for the calculation. The integral of the correlation function over time is called the *integrated autocorrelation function* (IACF)

$$\mathrm{IACF}(f(t)) = \int_0^{\infty} \mathrm{ACF}(f(t))\mathrm{d}t.$$

The IACF is very similar to the *integrated autocorrelation time*, which is calculated similarly but normalizing the ACF's by ACF(0), which is the variance (Straatsma, Berendsen, and Stam, 1986). Intuitively, the integrated autocorrelation time can be understood as measuring the time needed, on average, for generating a non-correlated sample. It can be seen as the inverse of the *effective sample size* (ESS) (Geyer, 1992), a measure often used in statistical applications of Monte Carlo methods. In practice, all the correlation functions are calculated for discrete values. Low values of measured IACFs mean low correlations between the generated samples and thus better sampling.

Figure 3.11 presents the IACF measured in HMC and MD with different integrating schemes for the same range of time steps and trajectory lengths described above. Note, in the vertical axis, that computational time is used to normalize the results. The IACF values for 25 fs/50 fs are not plotted since the lack of stability at those step lengths in all integrating schemes produces poor, non-informative results.

In Figure 3.11 we use different symbols for different values of Δt to provide a better feeling for the relation between Δt and the efficiency achieved. Two different symbols corresponding to the same $\Delta t \times L$ mean that two different combinations of Δt and L are possible to get the same number on the x-axis.

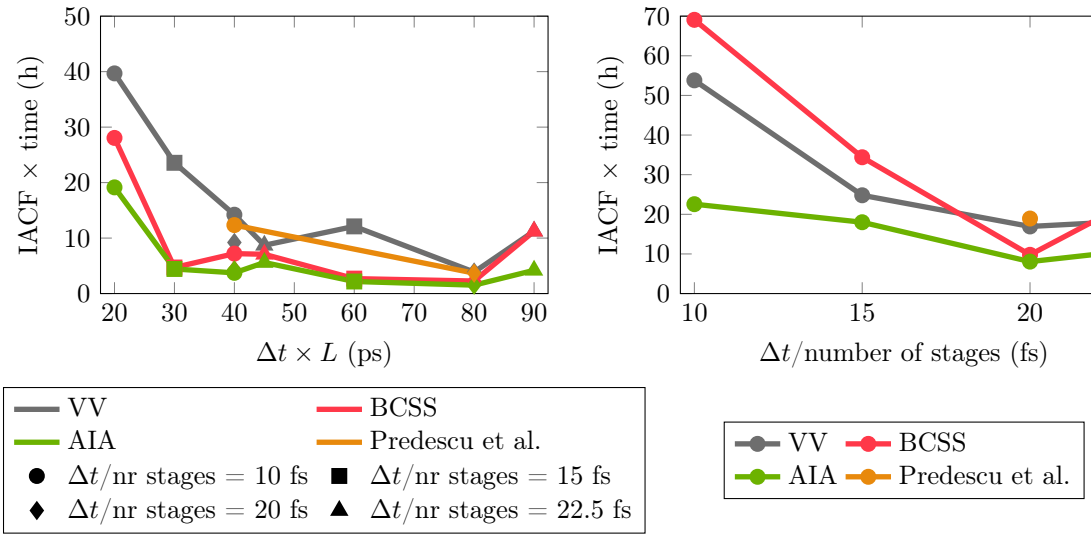


FIGURE 3.11: Toxin. IACF of the drift of the toxin to the preferred interfacial location evaluated as a function of L and Δt in HMC tests (left) and as a function of Δt in MD runs (right). Four integrating schemes were tested in HMC and MD simulations: velocity Verlet, the two-stage integrator BCSS, the HOH-integrator of Predescu et al. and the AIA integrators.

As seen from Figure 3.11, for all combinations of Δt and L, both HMC and MD simulations using the AIA integrators decorrelated faster than the corresponding simulations that used the velocity Verlet integrator, BCSS or the method of Predescu et al. In fact, for some specific choices of Δt the AIA integrators led to an efficiency several times higher than that of the velocity Verlet or any of the tested two-stage integrators. This applies to both simulation methods, HMC and MD. The fact that the better energy conservation of AIA led to better sampling efficiency in Hybrid Monte Carlo simulations was not surprising. For molecular dynamics, better conservation energy guarantees better accuracy but not necessarily better sampling. However, Figure 3.11 clearly demonstrates the positive impact of energy conservation on the sampling performance of MD. Still, comparison of the two plots in Figure 3.11 reveals the clear superiority in sampling efficiency of HMC over MD for the tested system.

A few more useful observations may be extracted from Figure 3.11. Analyzing the IACF calculated for HMC simulations with different combinations of Δt and L, one can conclude that, for fixed Δt, a larger L gives better performance for all integrators. Moreover, to achieve better performance, the choice of the product of Δt and L is more important than Δt itself. For instance, $\Delta t = 30$ fs and $L = 2000$ is a better choice than $\Delta t = 40$ fs and $L = 1000$.

At this stage, we can conclude that the Adaptive Integration Approach outperforms the other tested schemes in accuracy, stability and sampling efficiency for all tested time steps. As one can expect, long time steps, close to the maximum allowed by stability, lead to accuracy and performance degradation in all schemes. For the adaptive scheme, this effect is much smoother.

These conclusions are also supported by the results obtained in HMC and MD simulations of 216 molecules of water at 300 K. The model used is the flexible version of SPC (Berendsen et al., 1981). Taking into account the important role water plays in biomolecular simulations, we include here two plots in Figure 3.12 showing the advantage of AIA over other integrating schemes in sampling with HMC (left) and MD (right) simulations. We notice that a time step

of 2 fs, chosen for MD simulation, was close to the stability limit of all considered integrators but since BCSS has the shortest limit its performance was affected the most.

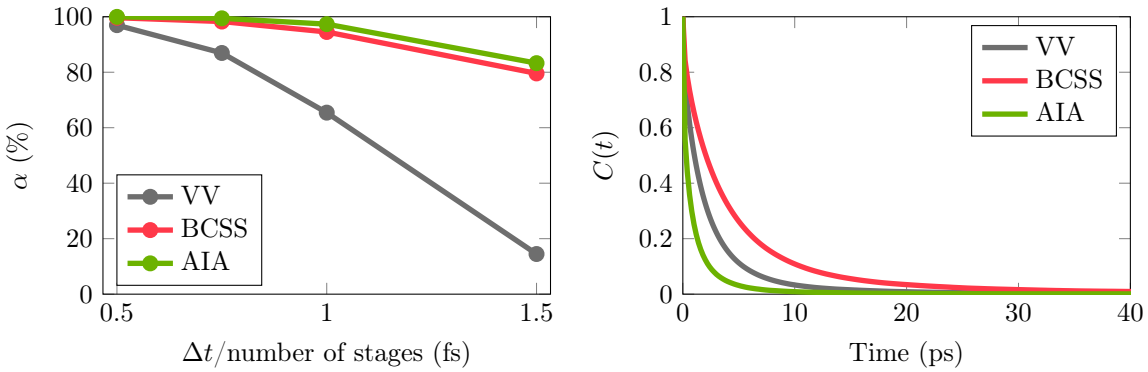


FIGURE 3.12: Water. Effect of the parameter b on the resulting acceptance rates in HMC simulations (left) and autocorrelation functions of the hydrogen bonding in MD simulations (right) for $\Delta t = 2$ fs. The two-stage integrator loses performance at the chosen time step whereas the AIA not only outperforms this integrator but also shows faster convergence than the standard velocity Verlet provides. The IACF's are: VV = 12.31, BCSS = 22.92, AIA = 5.66.

3.6.2 Constrained system

For testing efficiency of the AIA integrators in simulations of constrained systems, we followed the same strategy as in Section 3.6.1. The chosen time steps for the tests in this case, however, were in the range typical for time steps used in atomistic simulations and thus differed from those considered in coarse-grained experiments in Section 3.6.1. More specifically, we tested the following time steps, $\Delta t/nr$ ($nr = 1$ for Verlet and 2 otherwise): 1 fs, 1.5 fs, 2 fs, 2.5 fs. The numbers of steps in MD trajectories in HMC were the same as in Section 3.6.1, i.e., 2000 and 4000 in the tests with Verlet, and 1000 and 2000 for the two-stage methods. The measured acceptance rates were averaged over different lengths L for each Δt.

To our satisfaction, the positive impact of the AIA strategy on the quality of simulations demonstrated in unconstrained systems has also been observed in the case of constrained dynamics.

Figure 3.13 shows trends that match those summarized in Figure 3.6. The only significant difference is for BCSS; where the loss in performance at larger Δt is smaller for villin (Figure 3.13) than for toxin (Figure 3.6).

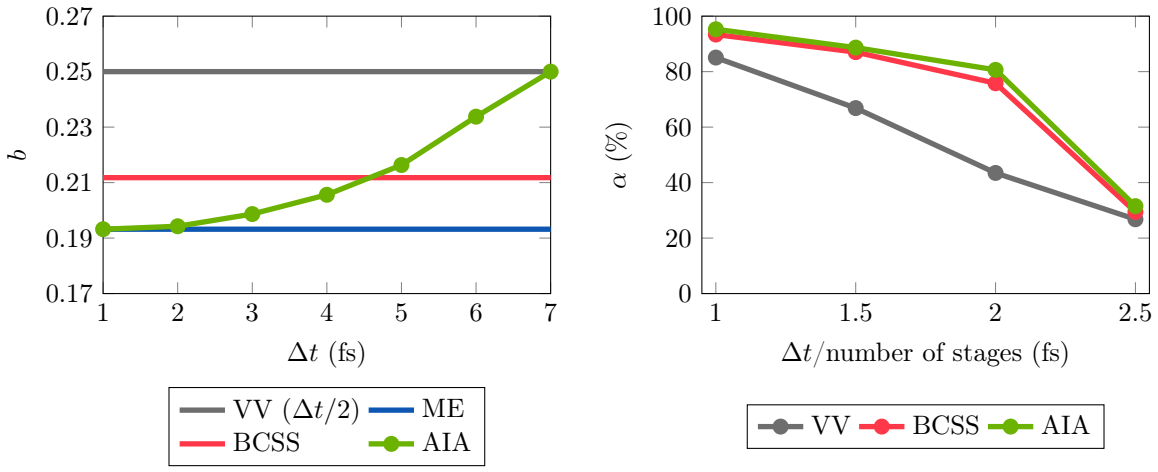


FIGURE 3.13: Villin. Dependence of the parameter b on the choice Δt (left) and its effect on the resulting acceptance rates in HMC simulations (right).

As in the case of the unconstrained system, the "convergence" of AIA to the velocity Verlet integrator was also observed (at around 2.25 fs/4.5 fs), but the resulting acceptance rates were so low in all tests that such experiments have been excluded from consideration.

Villin system is a popular benchmark for studying folding processes, due to its comparatively fast folding times. In the results presented here, we did not aim to investigate in full the folding of villin. Rather, the fast folding helped us to design computationally feasible tests for measuring accuracy and efficiency of the different numerical integrators.

Calculated averages of simulated temperatures in HMC and MD tests were used for evaluating the accuracy provided by the velocity Verlet integrator and the two-stage integrating schemes of interest. As in Section 3.6.1, the length of tests with HMC and MD simulations was fixed and sufficient to analyze the effect of Δt on the level of accuracy achieved in simulations, but not to guarantee low statistical errors.

As in Section 3.6.1, Figure 3.14 shows the dependence of the temperature RMSD with respect to the target temperature on the chosen Δt, trajectory lengths and integrators for HMC, and the average temperatures with the v-rescale thermostat for different time steps and integrators in MD. Evidently, AIA provided the smallest fluctuations of averages as a function of Δt within the inspected range of time steps, even though the differences in the data obtained with the different integrators were less marked than in the case of toxin in Section 3.6.1. Degradation of accuracy was observed for larger Δt in all simulations but was less visible for AIA than for Verlet or BCSS. The data collected at $\Delta t/nr = 2.5$ fs showed poor accuracy for all tests.

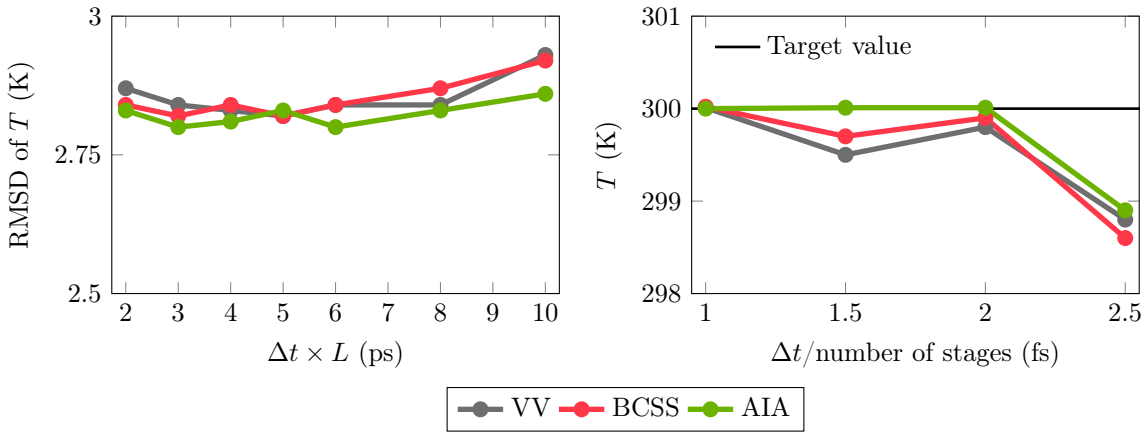


FIGURE 3.14: Villin. Temperature RMSD with respect to the target temperature in HMC simulations with different lengths of trajectories L, time steps Δt and integrating schemes (left), and the average temperature in MD simulations using various time steps Δt and integrators (right). The target temperature was set to 300 K. The v-rescale thermostat was applied in MD.

We completed our testing of AIA for constrained dynamics with an analysis of its impact on the sampling performance of HMC and MD. We chose to measure the quality of sampling through the positional RMSD from the native structure as a function of the simulation steps in both HMC and MD cases. The state of a protein folding can be understood by computing the root-mean-square deviation (RMSD) of the α-carbon. It can be used to make a comparison between the structure of a partially folded protein and the structure of the native state. The RMSD of certain atoms in a molecule with respect to a reference structure is calculated as

$$\text{RMSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\delta_i^2},$$

where δ_i is the distance between the atoms i in the two structures compared. As it is done in (van der Spoel and Lindahl, 2003), we have calculated what the authors call RMST, the maximum RMSD of the α-carbon between any two structures in a simulation. The idea is to roughly measure the extent of the conformational space sampled in a simulation. As in the unconstrained case, we have also plotted these values for the different combinations of time step and length of trajectories $\Delta t \times L$. In Figure 3.15 the simulation results obtained with different integrators are compared. It can be observed, in both HMC and MD cases, that AIA leads to a broader sampling of the conformational space no matter the choice of time step or trajectory length. The largest difference with respect to velocity Verlet can be observed when the biggest time step $\Delta t = 2$ fs is used.

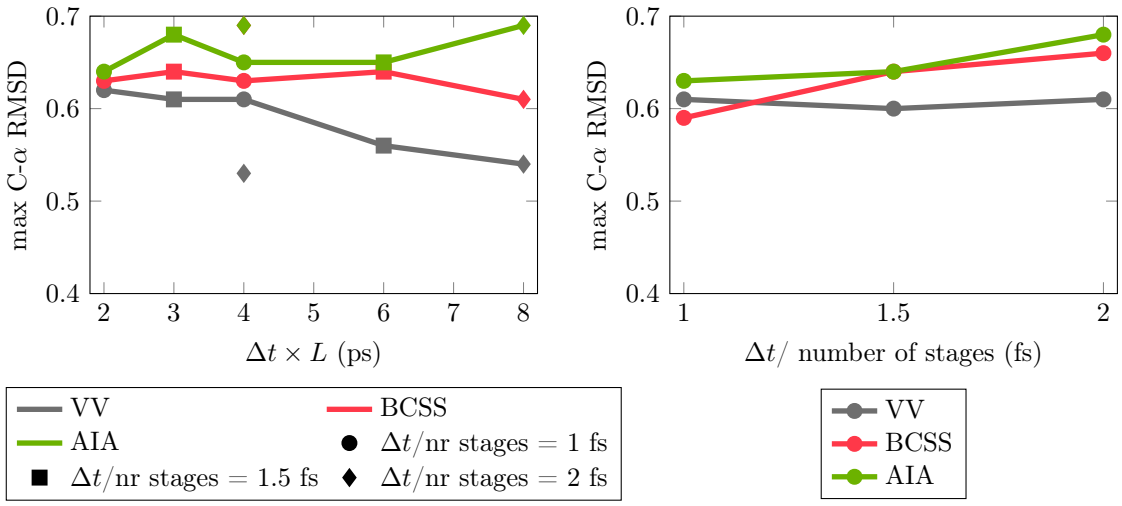


FIGURE 3.15: Villin. Maximum α-carbon RMSD between any two structures in HMC simulations with different lengths of trajectories L, time steps Δt and integrating schemes (left) and in MD simulations using various time steps Δt and integrators (right).

We have also computed the radius of gyration, which provides an estimation of the compactness of the desired structure. As in (van der Spoel and Lindahl, 2003), we have considered the experimental value 0.94 nm (McKnight, Matsudaira, and Kim, 1997) as a target value. The simulations performed are not long enough to observe any proper convergence to the value. However, the tendency of the protein evolution can be seen through the comparison of the simulated radius of gyration with the target one. In Figure 3.16 the average radii of gyration obtained from HMC (left) and MD (right) simulations using different integrators and different values of simulation time steps and trajectory lengths are presented. While the results associated with the velocity Verlet and BCSS integrators are still far from the target value, the averages produced with AIA are, regardless a choice of simulation parameters, always closer to 0.94 nm both in HMC and in MD.

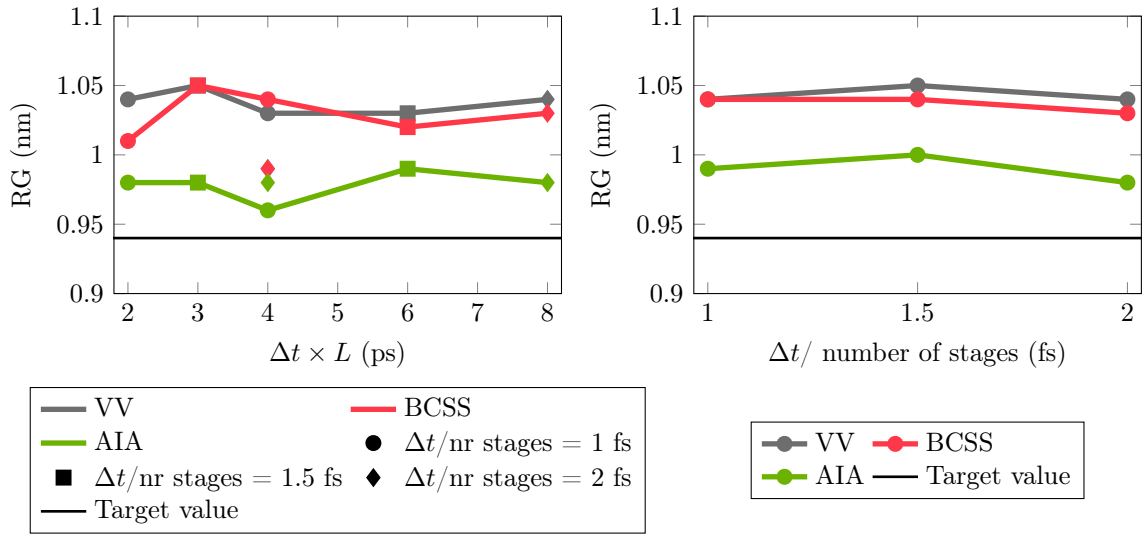


FIGURE 3.16: Villin. Average radii of gyration in HMC simulations with different time steps Δt, lengths of trajectories L and integrating schemes (left) and in MD simulations using various time steps Δt and integrators (right). The experimental target radius of gyration is 0.94 nm.

Similar trends were seen in GHMC simulations. The results are not shown (see Section 3.5).

It is impossible, with basis on these short tests, to make precise conclusions about features of the folding process, e.g., about the folding rate. More detailed studies of the protein folding are advisable. However, what can be concluded without hesitation is that sampling in molecular simulations of atomistic constrained systems with HMC and MD benefits from integrators that guarantee the best possible conservation of energy, as is the case with AIA.

3.7 Conclusions

In this chapter, we have presented an alternative to the standard velocity Verlet integrator, known to be the state-of-the-art method for numerical integration of the Hamiltonian equations in molecular dynamics. The novel methodology, which we call the Adaptive Integration Approach, or AIA, offers, for any chosen time step, a system-specific integrator which guarantees the best energy conservation for harmonic forces achievable by an integrator from the family of two-stage splitting schemes, including Verlet. While improvements in energy conservation do not necessarily imply dramatic changes in sampling, they improve acceptance rates in Hybrid Monte Carlo methods. The experiments performed in the present study also show that in molecular dynamics AIA leads to improvements of sampling as measured by the metrics considered. The improved sampling may arise as a consequence of either enhanced accuracy with a given time step or due to the possibility of longer time steps.

The AIA scheme can be implemented, without introducing computational overheads in simulations, in any software package which includes MD and/or HMC. In this study, we implemented the AIA method in MultiHMC-GROMACS, a modified version of the popular GROMACS code, and tested the new algorithm in HMC and MD simulations of unconstrained and constrained dynamics. The tests demonstrated the superiority of the novel scheme over Verlet, BCSS and the HOH-integrator of Predescu et al., 2012. For a wide range of time steps

and MD trajectory lengths, AIA outperformed other tested integrating schemes in accuracy and sampling efficiency. The analysis of integrated autocorrelation functions and folding evolution demonstrated, for selected sizes of time steps, that AIA possesses up to 5 times better sampling performance than the other tested schemes.

The idea proposed here may be extended in a natural way to multiple-time step (MTS) algorithms such as those based on Reversible multiple time scale molecular dynamics (Tuckerman, Berne, and Martyna, 1992), the Generalized Hybrid Monte Carlo method (Escribano et al., 2015), the Stochastic, resonance-free multiple time step algorithm (Leimkuhler, Margul, and Tuckerman, 2013), etc.

In summary, the proposed Adaptive Integration Approach introduces a rational control on integrating the equations of motions in molecular dynamics simulations, leading to enhanced accuracy and performance. To our knowledge, this feature was desired but missing by the molecular simulation community.

3.8 Published paper

1. **M. Fernández-Pendás**, E. Akhmatskaya, and J. M. Sanz-Serna (2016). "Adaptive multi-stage integrators for optimal energy conservation in molecular simulations". In: *Journal of Computational Physics* 327, pp. 434–449. URL: http://www.sciencedirect.com/science/article/pii/S0021999116304569

Chapter 4

Enhancing Performance and Accuracy of HMC for Simulation of Complex Systems: Importance Sampling

4.1 Overview

A way of improving sampling performance of HMC methods is to introduce importance sampling as suggested in different works such as (Izaguirre and Hampton, 2004; Sweet et al., 2009; Akhmatskaya and Reich, 2008; Escribano et al., 2015; Akhmatskaya and Reich, 2011a; Radivojević, 2016). Taking advantage of the fact that symplectic integrators preserve modified Hamiltonians more accurately than true Hamiltonians, the authors proposed to sample with respect to modified/shadow Hamiltonians and to recover the desired distribution by reweighting. The resulting algorithms are capable of maintaining high acceptance rates and usually exhibit better efficiency than their predecessor HMC as explained in (Radivojević, 2016; Akhmatskaya and Reich, 2011b; Wee et al., 2008; Escribano et al., 2017). Moreover, in many applications, using the velocity Verlet integrator is sufficient to provide the number of accepted proposals adequate for generating good statistics even with the parameter settings in which HMC may fail.

In this chapter, the general family of HMC methods combined with importance sampling is presented. We call them Modified Hamiltonian Monte Carlo methods and they are described in Section 4.2. In Section 4.3 the Generalized Shadow Hybrid Monte Carlo method, a particular case of the Modified Hamiltonian Monte Carlo methods, and its particular features are summarized. The algorithms presented here will be studied closely in the following chapters.

4.2 Modified Hamiltonian Monte Carlo methods (MHMC)

The family of modified Hamiltonian Monte Carlo (MHMC) methods consists of HMC algorithms which, instead of sampling from the target canonical distribution

$$\pi(\mathbf{q},\mathbf{p}) \propto \exp\left(-\beta H(\mathbf{q},\mathbf{p})\right) \tag{4.1}$$

known up to a multiplicative constant, sample from an auxiliary importance canonical density

$$\tilde{\pi}(\mathbf{q},\mathbf{p}) \propto \exp\left(-\beta \tilde{H}^{[k]}(\mathbf{q},\mathbf{p})\right). \tag{4.2}$$

Here $\tilde{H}^{[k]}$ denotes a truncated modified Hamiltonian to be described later. Such methods take advantage of two facts in order to enhance sampling efficiency of HMC. First, the closeness of $\tilde{H}^{[k]}$ to H makes it possible to implement an importance sampling approach and use samples of $\tilde{\pi}$ as a means towards computing expectations with respect to π. Second, the fact that the integrator preserves $\tilde{H}^{[k]}$ better than it does preserve H leads to a more favorable value of the acceptance probability in the algorithms.

Symplectic integrators for the Hamiltonian dynamics with Hamiltonian function $H(\mathbf{q}, \mathbf{p})$, while not preserving the value of H exactly along the computed trajectory, do preserve exactly the value of a so-called modified Hamiltonian (cf. (Sanz-Serna and Calvo, 1994; Leimkuhler and Reich, 2004; Hairer, Lubich, and Wanner, 2006))

$$\tilde{H} = H + \Delta t H_2 + \Delta t^2 H_3 + \cdots ,$$

where Δt is the integration time step. For an integrator of order p, $\tilde{H} = H + \mathcal{O}(\Delta t^p)$, so that $H_2, \ldots, H_p$ vanish. In (4.2), $\tilde{H}^{[k]}$, $k > p$, is the truncation of $\tilde{H}$ given by

$$\tilde{H}^{[k]} = H + \Delta t^p H_{p+1} + \cdots + \Delta t^{k-1} H_k.$$

One can define the *modified energy error* as

$$\Delta \tilde{H}^{[k]} = \tilde{H}^{[k]} \left(\Psi_{\Delta t, L}(\mathbf{q}, \mathbf{p}) \right) - \tilde{H}^{[k]}(\mathbf{q}, \mathbf{p}). \tag{4.3}$$

The expectation of the increment of H in an integration leg satisfies

$$\mathbb{E}_{\pi}[\Delta H] = \mathcal{O}\left(D \Delta t^{2p} \right), \tag{4.4}$$

while

$$\mathbb{E}_{\tilde{\pi}}[\Delta \tilde{H}^{[k]}] = \mathcal{O}\left(D \Delta t^{2k} \right), \tag{4.5}$$

with $k > p$ (Beskos et al., 2013) and therefore MHMC algorithms may benefit from high acceptance rates due to the better conservation of $\tilde{H}^{[k]}$. Equation (4.5) shows that the energy error depends on the order of the modified Hamiltonian rather than on the order of an integrator as in (4.4). Thus, an increase in the dimension D of the simulated system can be counterbalanced by an increase in the order k of the modified Hamiltonian. This allows for maintaining high acceptance rates without increasing the order of the integrator.

The objective of a modified Hamiltonian Monte Carlo method is to sample from a distribution with probability density function

$$\pi(\mathbf{q}) \propto \exp(-\beta U(\mathbf{q})). \tag{4.6}$$

This is achieved indirectly through sampling from the modified distribution (4.2). In our studies we are considering Hamiltonians of the form of (1.2), therefore, under the target (4.1), the variable position $\mathbf{q}$ has the marginal density (4.6).

Since in MHMC methods the samples are generated with respect to the modified or importance density, the computation of averages with respect to the target density after completion of the sampling procedure requires reweighting. If Ω_n, $n = 1, 2, \ldots, N$, are the values of an observable along a sequence of states $(\mathbf{q}^n, \mathbf{p}^n)$ drawn from $\tilde{\pi}$ (4.2), the averages with respect to π (4.1) are calculated as

$$\langle \Omega \rangle = \frac{\sum_{n=1}^{N} w_n \Omega_n}{\sum_{n=1}^{N} w_n}, \tag{4.7}$$

where the *importance weights* are given by

$$w_n = \exp\Big(-\beta\big(H(\mathbf{q}^n,\mathbf{p}^n) - \tilde{H}^{[k]}(\mathbf{q}^n,\mathbf{p}^n)\big)\Big).$$

If the target density π and the importance density $\tilde{\pi}$ were not close, one would typically encounter high variability among weights, which would lead to large errors in the expectation, as many samples would not contribute significantly to the computation of $\langle\Omega\rangle$.

Let us now describe a generic algorithm for an MHMC method. Given a sample $(\mathbf{q},\mathbf{p})$ from the joint distribution $\tilde{\pi}$, the next sample $(\mathbf{q}^{\text{new}},\mathbf{p}^{\text{new}})$ is defined as follows

- Obtain the new momentum $\mathbf{p}^*$ by applying a momentum update procedure that preserves the importance density $\tilde{\pi}$.
- Generate a proposal $(\mathbf{q}',\mathbf{p}')$ by simulating Hamiltonian dynamics with the initial condition $(\mathbf{q},\mathbf{p}^*)$ using a symplectic and reversible numerical integrator.
- Choose the next sample $(\mathbf{q}^{\text{new}},\mathbf{p}^{\text{new}})$ to be $(\mathbf{q}',\mathbf{p}')$ with the probability

$$\alpha = \min\left\{1, \frac{\tilde{\pi}(\mathbf{q}',\mathbf{p}')}{\tilde{\pi}(\mathbf{q},\mathbf{p}^*)}\right\}. \tag{4.8}$$

  Otherwise set $(\mathbf{q}^{\text{new}},\mathbf{p}^{\text{new}})$ to $\mathcal{F}(\mathbf{q},\mathbf{p}^*)$, where $\mathcal{F}(\mathbf{q},\mathbf{p}^*)$ flips the momentum $\mathbf{p}^*$, i.e., $\mathcal{F}(\mathbf{q},\mathbf{p}^*) = (\mathbf{q},-\mathbf{p}^*)$.

Since

$$\frac{\tilde{\pi}(\mathbf{q}',\mathbf{p}')}{\tilde{\pi}(\mathbf{q},\mathbf{p}^*)} = \exp\left(-\beta\left(\tilde{H}^{[k]}(\mathbf{q}',\mathbf{p}') - \tilde{H}^{[k]}(\mathbf{q},\mathbf{p}^*)\right)\right) = \exp\big(-\beta\Delta\tilde{H}^{[k]}(\mathbf{q},\mathbf{p}^*)\big),$$

one may expect, in view of (4.4)-(4.5), fewer rejections/momentum flips, and thus better sampling/more accurate dynamics when sampling with $\tilde{H}^{[k]}$ instead of H (Akhmatskaya and Reich, 2008; Akhmatskaya, Bou-Rabee, and Reich, 2009).

The first methods of the MHMC class were derived for atomistic simulations and differed from each other in the ways of refreshing the momentum, computing modified Hamiltonians and integrating the Hamiltonian dynamics. For example, in the (Separable) Shadow Hybrid Monte Carlo methods presented in (Izaguirre and Hampton, 2004; Sweet et al., 2009), a full momentum update is used, whereas in the Targeted Shadow Hybrid Monte Carlo (Akhmatskaya and Reich, 2006) and the Generalized Shadow Hybrid Monte Carlo (GSHMC) (Akhmatskaya and Reich, 2008), suitable modifications of the partial momentum update of Horowitz, 1991 are advocated in order to mimic the dynamics better and enhance sampling. More recent MHMC methods aim at specific applications, such as multi-scale (MTS-GSHMC) and mesoscale (meso-GSHMC) simulations in (Escribano et al., 2015) and (Akhmatskaya and Reich, 2011a), respectively; and computational statistics (Mix&Match Hamiltonian Monte Carlo) in (Radivojević, 2016; Radivojević and Akhmatskaya, 2017). As demonstrated in the original papers, for some particular problems, the use of MHMC methods resulted in a sampling efficiency several times higher than that observed with the conventional sampling techniques, such as MD, Monte Carlo (MC) and HMC.

In the following section, we focus our attention on a particular case of the MHMC methods, the Generalized Shadow Hybrid Monte Carlo. The main features of the algorithm are provided, namely the modified Hamiltonians used in the importance sampling and its momentum update procedure. The algorithm is presented in detail.

4.3 Generalized Shadow Hybrid Monte Carlo (GSHMC)

In this section, we provide the details of the Generalized Shadow Hybrid Monte Carlo (GSHMC) algorithm that will be extensively used in the following chapters. The GSHMC algorithm was first introduced by Akhmatskaya and Reich, 2008 for sampling in molecular simulation. Its purpose was to enable sampling of large complex systems while retaining dynamical information. This is achieved by employing the modified energy for sampling and by partially updating momentum. As it has been explained above, the former leads to lower discretization errors, which implies higher acceptance rates for large system sizes as well as a reduced negative impact of the undesired momentum flips.

GSHMC proved to be successful in simulations of complex molecular systems in Biology and Chemistry (Wee et al., 2008; Akhmatskaya and Reich, 2011b; Escribano, Akhmatskaya, and Mujika, 2013; Akhmatskaya et al., 2013) and has been adapted for multi-scale simulations in MTS-GSHMC (Escribano et al., 2015), mesoscale simulations in Meso-GSHMC (Akhmatskaya and Reich, 2011a) and solid-state simulations in RSM-GSHMC (Escribano et al., 2017).

The GSHMC method involves two major steps: the Partial Momentum Monte Carlo (PMMC) step, and the Molecular Dynamics Monte Carlo (MDMC) step[1]. The partial momentum update allows for keeping the dynamical information during the simulation similar to a stochastic Langevin dynamics simulation, in which the friction coefficient restricts the noise added to the momentum. A modified Metropolis test is introduced in PMMC step to preserve the desired modified density $\tilde{\pi}$. As to the MDMC step, the only difference with the one of the GHMC method is that in the Metropolis test the modified Hamiltonian is used instead of the true Hamiltonian.

Now we explain in detail the main features of the GSHMC method that make it a particular case of the MHMC family introduced in Section 4.2.

Modified or shadow Hamiltonians The original GSHMC method in (Akhmatskaya and Reich, 2008) employs a Lagrangian formulation of modified Hamiltonians of an arbitrary kth order for the leapfrog integrator. In the case of $k = 4$, the shadow Hamiltonians have the form

$$\tilde{H}^{[4]} = \frac{1}{2}\dot{\mathbf{Q}}[M\dot{\mathbf{Q}}] + U(\mathbf{Q}) + \frac{\Delta t^2}{24}\left(2\dot{\mathbf{Q}}[M\mathbf{Q}^{(3)}] - \ddot{\mathbf{Q}}[M\ddot{\mathbf{Q}}]\right), \tag{4.9}$$

where $\mathbf{Q}(t) \in \mathbb{R}^D$ is the unique interpolation polynomial of degree four, constructed for t_n, $n \in \{0, L\}$ from a given numerical trajectory $\{\mathbf{q}^i\}_{i=-2}^{L+2}$ passing through points

$$\mathbf{Q}(t_i) = \mathbf{q}^i, \qquad i = n-2, \ldots, n, \ldots, n+2.$$

The derivatives of the positions in (4.9) are approximated by the central differences method (Fornberg, 1988). More details will be presented later on in Section 7.3.1.

More recently, in (Radivojević, 2016) a method for constructing the shadow Hamiltonians for splitting integrators (of two, three and four stages) has been presented. The

[1] The names of the steps are taken from the original paper (Akhmatskaya and Reich, 2008).

fourth-order modified Hamiltonian has the shape:

$$\tilde{H}^{[4]}(\mathbf{q},\mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + U(\mathbf{q}) + \Delta t^2 \left(\lambda \mathbf{p}^T M^{-1}\nabla_{\mathbf{q}}\dot{U}(\mathbf{q}) + \mu \nabla_{\mathbf{q}} U(\mathbf{q})^T M^{-1}\nabla_{\mathbf{q}} U(\mathbf{q})\right), \tag{4.10}$$

where λ and μ are quantities that depend on the parameters of the integrator used. For the two-stage integrators of the family (3.43) such quantities read as

$$\lambda = \frac{6b-1}{24}, \qquad \mu = \frac{6b^2-6b+1}{12}.$$

For the appropriate choice of b, i.e., $b = 1/4$, the shadow Hamiltonian (4.10) can be also used with Verlet. The coefficients λ and μ for the three-stage integrators of the family (3.27) are

$$\lambda = \frac{1-6a(1-a)(1-2b)}{12}, \qquad \mu = \frac{6a(1-2b)^2-1}{24}.$$

As in the two-stage case, for the appropriate choice of a and b, i.e., $a = 1/3$ and $b = 1/6$, the shadow Hamiltonian (4.10) can be used with Verlet.

Momentum update (PMMC step) Following the ideas of (Horowitz, 1991; Kennedy and Pendleton, 2001) (see Section 2.2 for details), the momenta in GSHMC are only partially updated before starting a new Hamiltonian trajectory for generating a next proposal:

$$\begin{aligned} \mathbf{p}^* &= \quad \cos\varphi\ \mathbf{p} + \sin\varphi\ \mathbf{u} \\ \mathbf{u}^* &= -\sin\varphi\ \mathbf{p} + \cos\varphi\ \mathbf{u}, \end{aligned} \tag{4.11}$$

with the angle $\varphi \in (0, \pi/2]$ controlling the amount of introduced noise and the noise vector $\mathbf{u}$ is drawn from the normal distribution $\mathcal{N}(0, \beta^{-1}M)$. A low value of φ will result in $\mathbf{p}^*$ being close to $\mathbf{p}$ and the behavior of the algorithm will be close to conventional MD. For φ near $\pi/2$, $\mathbf{p}^*$ will be very different from $\mathbf{p}$, just recovering HMC behavior.

The proposed trial momentum $\mathbf{p}^*$ is accepted ($\bar{\mathbf{p}} = \mathbf{p}^*$) with probability

$$\alpha_p = \min\left\{1, \frac{\hat{\pi}(\mathbf{q},\mathbf{p}^*,\mathbf{u}^*)}{\hat{\pi}(\mathbf{q},\mathbf{p},\mathbf{u})}\right\}, \tag{4.12}$$

where $\hat{\pi}$ is the extended p.d.f.

$$\hat{\pi}(\mathbf{q},\mathbf{p},\mathbf{u}) \propto \exp\left(-\beta \hat{H}(\mathbf{q},\mathbf{p},\mathbf{u})\right)$$

corresponding to the *extended Hamiltonian*

$$\hat{H}(\mathbf{q},\mathbf{p},\mathbf{u}) = \tilde{H}^{[k]}(\mathbf{q},\mathbf{p}) + \frac{1}{2}\mathbf{u}^T M^{-1}\mathbf{u}. \tag{4.13}$$

In case of rejection we set $\bar{\mathbf{p}} = \mathbf{p}$.

This step can be considered as a standard HMC method in which the vector $\mathbf{q}$ is fixed, the vector $\mathbf{p}$ plays a role of the "position" and the noise vector $\mathbf{u}$ becomes "conjugate momenta".

The algorithm of GSHMC can be summarized as follows:

Algorithm 3 Generalized Shadow Hybrid Monte Carlo

Input: M: mass matrix
Δt: time step
L: number of integration steps
Ψ: discretization scheme
N: number of MC iterations
T: temperature
$\varphi \in (0, \pi/2]$: noise angle
k: order of the shadow Hamiltonian

initialize $(\mathbf{q}^0, \mathbf{p}^0)$
for $n = 1, \ldots, N$ **do**
    calculate the shadow Hamiltonian at $(\mathbf{q}, \mathbf{p}) = (\mathbf{q}^{n-1}, \mathbf{p}^{n-1})$
PMMC step
    generate a proposal by the partial momentum update

$$\begin{aligned} \mathbf{p}^* &= \quad \cos\varphi\ \mathbf{p} + \sin\varphi\ \mathbf{u} \\ \mathbf{u}^* &= -\sin\varphi\ \mathbf{p} + \cos\varphi\ \mathbf{u} \end{aligned}$$

    where $\mathbf{u} \sim \mathcal{N}(0, \beta^{-1}M)$
    calculate the shadow Hamiltonian at $(\mathbf{q}, \mathbf{p}^*)$
    calculate the acceptance probability

$$\alpha_p = \min\left\{1, \frac{\exp\left(-\beta\left(\tilde{H}^{[k]}(\mathbf{q}, \mathbf{p}^*) + \frac{1}{2}(\mathbf{u}^*)^T M^{-1}\mathbf{u}^*\right)\right)}{\exp\left(-\beta\left(\tilde{H}^{[k]}(\mathbf{q}, \mathbf{p}) + \frac{1}{2}\mathbf{u}^T M^{-1}\mathbf{u}\right)\right)}\right\}$$

    Modified Metropolis test

$$(\mathbf{q}, \bar{\mathbf{p}}) = \begin{cases} (\mathbf{q}, \mathbf{p}^*) & \text{with probability } \alpha_p \\ (\mathbf{q}, \mathbf{p}) & \text{otherwise} \end{cases}$$

MDMC step
    generate a proposal by integrating Hamiltonian dynamics

$$(\mathbf{q}', \mathbf{p}') = \Psi_{\Delta t, L}(\mathbf{q}, \bar{\mathbf{p}})$$

    calculate the shadow Hamiltonian at $(\mathbf{q}', \mathbf{p}')$
    calculate the acceptance probability

$$\alpha = \min\left\{1, \exp\left(-\beta\left(\tilde{H}^{[k]}(\mathbf{q}', \mathbf{p}') - \tilde{H}^{[k]}(\mathbf{q}, \bar{\mathbf{p}})\right)\right)\right\}$$

    Metropolis test

$$(\mathbf{q}^n, \mathbf{p}^n) = \begin{cases} (\mathbf{q}', \mathbf{p}') & \text{with probability } \alpha \\ \mathcal{F}(\mathbf{q}, \bar{\mathbf{p}}) & \text{otherwise} \end{cases}$$

    compute the weight

$$w_n = \exp\left(-\beta\left(H(\mathbf{q}^n, \mathbf{p}^n) - \tilde{H}^{[k]}(\mathbf{q}^n, \mathbf{p}^n)\right)\right)$$

13: **end for**
14: calculate the average of an observable $\Omega(\mathbf{q}, \mathbf{p})$

$$\langle \Omega \rangle = \frac{\sum_{n=1}^{N} w_n \Omega_n}{\sum_{n=1}^{N} w_n}$$

The GSHMC method introduces computational overheads compared to HMC/GHMC due to the two evaluations of the shadow Hamiltonian as well as two Metropolis tests per MC step. We will come back to this issue later when we deal with the applications of this method.

A choice of parameters, such as $\Delta t, L, \varphi$, can also affect the accuracy and performance of GSHMC. For instance, if a time step Δt, used for the integration of the equation of motion, is chosen to be too short, the computational cost of the simulation increases, while choices of too long Δt lead to less accurate integration and, potentially, to higher rejection rates.

Similarly to HMC and GHMC, too small values of L reduce the sampling efficiency. They also imply more frequent calculations of shadow Hamiltonians, which might introduce significant computational overheads.

Small values of the angle φ are advisable for maintaining the dynamics of the simulated system. However, too small values may reduce sampling efficiency. On the other hand, too large values increase momenta rejection rates and do not reproduce the dynamics of the system. Some choices of φ which lead to negative effects are summarized in Table 4.1.

conditions	**observations**	**consequences**
AR $\sim 100\%$, $\varphi \sim 0$	MD behavior	poor sampling, thermalization
AR $\sim 0\%$, φ any	MD behavior	poor sampling, thermalization
AR $\sim 100\%$, $\varphi \neq 0$	dynamics are not preserved	poor sampling

TABLE 4.1: Possible negative effects of too small choices of the angle φ. AR stands for the acceptance rate for momenta.

Large orders of the shadow Hamiltonians might be computationally demanding since higher order derivatives have to be computed (4.9) (Akhmatskaya and Reich, 2008). However, for some problems, using too small orders may not provide a good approximation of the true Hamiltonian and, consequently, simulation properties.

4.3.1 Implementation of GSHMC in MultiHMC-GROMACS

The GSHMC method was patented by Fujitsu in the UK and the US (Akhmatskaya, Reich, and Nobes, 2009; Akhmatskaya, Nobes, and Reich, 2011). Due to IPR issues, there were difficulties with the implementation of the method in open source software. This changed in November 2015, when Fujitsu issued the license giving permission to use the patented method in open source software and permission to Elena Akhmatskaya to implement and use know-how. The first implementation of GSHMC in the BCAM in-house software package MultiHMC-GROMACS was presented in (Escribano, Akhmatskaya, and Mujika, 2013). This is the basis for the current implementation. More details will be provided in Section 7.3.

Two types of modified Hamiltonians are currently available in MultiHMC-GROMACS: the shadow Hamiltonian in a Lagrangian formulation (4.9) as presented for the Verlet/leapfrog

integrator in (Akhmatskaya and Reich, 2008) and the modified Hamiltonian (4.10), derived in (Radivojević, 2016), for two- and three-stage integrators of the families (3.43) and (3.27).

Here we propose an efficient implementation of the modified Hamiltonians in both formulations. We limit our discussion to shadow Hamiltonians for two-stage integrators. The same ideas were applied to shadow Hamiltonians associated with splitting integrators of more stages.

In the shadow Hamiltonian (4.10) the Hessian $\nabla_{\mathbf{q}}\dot{U}(\mathbf{q})$ appears. It is not feasible to compute it in the molecular simulations. Thus, one can rewrite (4.10) in terms of derivatives of the position instead of time derivatives as

$$\tilde{H}^{[4]}(\mathbf{q},\mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + U(\mathbf{q}) + \Delta t^2\left(\lambda \mathbf{p}^T M^{-1} U_{\mathbf{qq}}(\mathbf{q}) M^{-1}\mathbf{p} + \mu U_{\mathbf{q}}(\mathbf{q})^T M^{-1} U_{\mathbf{q}}(\mathbf{q})\right).$$

Then, using the relations in Appendix B, we obtain

$$\tilde{H}^{[4]}(\mathbf{q},\mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + U(\mathbf{q}) + \Delta t^2\left(-\lambda \dot{\mathbf{q}}^T M \dddot{\mathbf{q}} + \mu \ddot{\mathbf{q}}^T M \ddot{\mathbf{q}}\right),$$

which can also be expressed in terms of derivatives of momenta as

$$\tilde{H}^{[4]}(\mathbf{q},\mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + U(\mathbf{q}) + \Delta t^2\left(-\lambda \mathbf{p}^T M^{-1} \ddot{\mathbf{p}} + \mu \dot{\mathbf{p}}^T M^{-1} \dot{\mathbf{p}}\right).$$

Similarly to (Akhmatskaya and Reich, 2008), one can consider

$$\tilde{H}^{[4]} = \frac{1}{2}\mathbf{P}[M\mathbf{P}] + U(\mathbf{Q}) + \Delta t^2\left(-\lambda \mathbf{P}[M\ddot{\mathbf{P}}] + \mu \dot{\mathbf{P}}[M\dot{\mathbf{P}}]\right), \tag{4.14}$$

where $\mathbf{P}(t) \in \mathbb{R}^D$ is the unique interpolation polynomial of degree four, constructed for t_n, $n \in \{0, L\}$ from the momenta associated to a given numerical trajectory $\{\mathbf{q}^i\}_{i=-2}^{L+2}$ passing through points

$$\mathbf{Q}(t_i) = \mathbf{q}^i, \qquad i = n-2, \ldots, n, \ldots, n+2.$$

The shadow Hamiltonian in (4.14) is more useful in practice than the original (4.9) since the order of the biggest derivatives is reduced by one: in (4.9) the third derivative of the position appears while in (4.14) it is reduced to a second derivative of the momenta. This second derivative can be computed numerically with central differences performing one step forward and one backward. It is less expensive than in the original implementation where two steps forward and two backward were required (see Section 7.3.1 for details). Thus, the computational cost of the computation of shadow Hamiltonians is reduced.

The original implementation of GSHMC in MultiHMC-GROMACS uses the leapfrog integrator in combination with the shadow Hamiltonian (4.9). We introduced in the MultiHMC-GROMACS package a family of advanced multi-stage integrators compatible with GSHMC, which we will discuss in detail in Chapters 6 and 7.

4.4 Summary

In this chapter, the family of modified Hamiltonian Monte Carlo (MHMC) methods has been presented. Importance sampling is used in MHMC as a way of enhancing the sampling performance of the traditional Hybrid Monte Carlo algorithms. The special attention has been paid to a particular MHMC method, the Generalized Shadow Hybrid Monte Carlo (GSHMC), first

introduced in (Akhmatskaya and Reich, 2008). We summarize GSHMC algorithm, analyze its performance potential and propose a new formulation and implementation of modified Hamiltonians for future use with multi-stage integrators.

Chapter 5

Extension of GSHMC to Various Statistical Ensembles

Isobaric-isothermal Ensemble

5.1 Introduction

The isobaric-isoenthalpic and isobaric-isothermal ensembles (also called NPH and NPT ensembles, respectively) are the statistical ensembles where the number of particles N, the pressure P as well as either the enthalpy H or the temperature T are each fixed to particular values. These ensembles play a fundamental role in chemistry and biology where many processes are carried out at constant pressure. Mathematical techniques called barostats are developed to keep constant pressure during a molecular simulation. In the case of NPT ensembles, barostats are combined with thermostats responsible for temperature maintenance. Numerical values of physical properties such as enthalpies, entropies and free energies of formation, redox potentials, equilibrium constants (e.g., acid ionization constants, solubility products, inhibition constants) and other similar data, are often reported under conditions of constant temperature and pressure.

In this study, we focus on the Andersen barostat (Andersen, 1980). In this approach, the system is coupled to a fictitious "pressure bath" using an extended Lagrangian, in which the volume acts as an additional variable. The coupling mimics the action of an imaginary external *piston* on a simulated system and the new variable plays the role of the coordinate of a *piston* linked to an external constant reference pressure. The resulting equations of motion produce trajectories which sample the NPH ensemble. The purpose of this chapter is to explain how the Andersen barostat can be combined with the GSHMC method in order to allow the GSHMC simulations in the NPT ensemble. We then propose the efficient implementation of the resulting NPT-GSHMC method in the GROMACS package and compare its accuracy and sampling efficiency with those offered by MD and NVT-GSHMC[1].

5.2 NPT-GSHMC

5.2.1 Formulation

The NPT-GSHMC method has been already mathematically formulated by Akhmatskaya and Reich, 2008. The method combines the Generalized Shadow Hybrid Monte Carlo (GSHMC)

[1]For simplicity, in this chapter we will use the NVT-GSHMC notation to refer to the GSHMC method sampling in the NVT ensemble.

methodology (Akhmatskaya and Reich, 2008) with the Andersen barostat (Andersen, 1980). In this section, we summarize the major steps that should be taken to extend the GSHMC algorithm to simulation at constant pressure.

The GSHMC method, as a modification of GHMC, consists of two alternating steps: (i) a generation of short molecular dynamics trajectories in the NVE ensemble, i.e., at a constant number of particles N, a constant volume V and a constant energy E; and (ii) a partial momentum update preceding each molecular dynamics trajectory. The decision on accepting/rejecting a proposal in steps (i) and (ii) is made using the appropriate Metropolis function with the true Hamiltonian replaced by the shadow Hamiltonian, $\tilde{H}(\mathbf{q},\mathbf{p})$. The shadow Hamiltonian used here is obtained from a truncated Taylor expansion of the usual Lagrangian following the standard Legendre transform (Akhmatskaya and Reich, 2008). In this chapter, we will use the fourth-order shadow Hamiltonian. The objective of the GSHMC method is to reduce the number of rejected trajectories through the use of shadow Hamiltonians while retaining dynamical information by only partially refreshing momenta. We recall that the GSHMC algorithm was summarized in Algorithm 3.

The following modifications are required to extend this methodology to simulations in the NPT ensemble. First, the MD simulations have to be performed in the NPE ensemble rather than in the NVE ensemble. If the barostat chosen in the NPE dynamics leads to the modification of Hamiltonian, then the shadow Hamiltonians will be different from those suggested for simulations in NVT ensembles and have to be derived specifically for this case. The integrator used for solving the associated modified equations of motions has to be symplectic as in the original GSHMC method. Below we briefly show how all those problems were addressed in the new NPT-GSHMC method.

The Andersen barostat has been chosen for maintaining the pressure constant in MD simulations. The Andersen barostat is based on the introduction of a new extended variable, which physical meaning is the (dynamic) value of the volume of the simulation box. The extended variable is an additional degree of freedom and it must be included in the Lagrangian to derive the new equations of motion. It is also used as a rescaling factor for the positions. Following Andersen's terminology, we refer to the extended variable as the *piston*.

More specifically, if we write the classical equations of motion (1.3) in dot notation,

$$\dot{\mathbf{q}} = \frac{\partial H}{\partial \mathbf{p}}, \qquad \dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{q}} \tag{5.1}$$

the coordinate vector $\mathbf{q} \in \mathbb{R}^{3D}$ is replaced by a scaled vector $\mathbf{d} \in \mathbb{R}^{3D}$ defined as

$$\mathbf{d} = \mathbf{q}/V^{1/3},$$

where V is the volume of the simulation box.

As the volume V is allowed to change in order to keep constant pressure, we introduce r as the dynamic value of the volume.

The extended Lagrangian density then reads as

$$\mathcal{L}(\mathbf{d}, r, \dot{\mathbf{d}}, \dot{r}) = \left\{ \frac{1}{2} r^{2/3} \dot{\mathbf{d}} \cdot [M\dot{\mathbf{d}}] - U(r^{1/3}\mathbf{d}) + \frac{\mu}{2}\dot{r}^2 - \alpha r \right\}, \tag{5.2}$$

where α is the external pressure acting on the system and $\mu > 0$ is the mass of the *piston*. The last two terms of (5.2) are in fact the kinetic and potential energies associated with the *piston*.

The Hamiltonian H, derived from (5.2) through Legendre transformation, is given by

$$H = \dot{\mathbf{d}} \cdot \nabla_{\dot{\mathbf{d}}}\mathcal{L} + \dot{r}\ \nabla_{\dot{r}}\mathcal{L} - \mathcal{L} = \frac{1}{2}\mathbf{p}_{\mathbf{q}} \cdot [M^{-1}\mathbf{p}_{\mathbf{q}}] + U(\mathbf{q}) + \frac{1}{2\mu}p_r^2 + \alpha r, \tag{5.3}$$

where

$$\mathbf{p}_{\mathbf{d}} = r^{2/3}M\dot{\mathbf{d}}, \qquad p_r = \mu\dot{r} \tag{5.4}$$

are the conjugate momenta in the NPE formulation, whereas $\mathbf{p}_{\mathbf{q}} = M\dot{\mathbf{q}} = \mathbf{p}_{\mathbf{d}}/r^{1/3}$ is the NVE momentum vector (5.1). The associated NPE equations of motion now can be obtained using (5.1), (5.3) and (5.4).

A time-reversible and symplectic method for integrating the NPE equations of motion is suggested by Akhmatskaya and Reich, 2008 and summarized below:

For a time step Δt, we can define the finite difference approximation for the time derivatives $\dot{\mathbf{d}}$ as

$$\delta_{\Delta t}\mathbf{d}^i = \frac{\mathbf{d}^{i+1} - \mathbf{d}^i}{\Delta t}.$$

Then, the extended Lagrangian in (5.2) can be approximated discretely with

$$\mathcal{L}_{\Delta t}(\{\mathbf{d}^i\}, \{r^i\}) = \sum_i \mathcal{L}_{\Delta t}(\mathbf{d}^i, r^i, \delta_{\Delta t}\mathbf{d}^i)\Delta t,$$

where

$$\begin{aligned}\mathcal{L}_{\Delta t}(\mathbf{d}^i, r^i, \delta_{\Delta t}\mathbf{d}^i)\Delta t = \frac{1}{2}\Big\{&(r^i)^{2/3}\delta_{\Delta t}\mathbf{d}^i \cdot [M\delta_{\Delta t}\mathbf{d}^i] - \left[U((r^i)^{1/3}\mathbf{d}) + U((r^{i+1})^{1/3}\mathbf{d})\right] \\ &+\frac{\mu}{2}\left(\frac{r^{i+1} - r^i}{\Delta t}\right)^2 - (\alpha r^i + \alpha r^{i+1})\Big\}\Delta t.\end{aligned} \tag{5.5}$$

Then, the discrete approximation in (5.5) is used as a generating function (Hairer, Lubich, and Wanner, 2006). Given $(\mathbf{d}^i, r^i, \mathbf{p}_{\mathbf{d}}^i, p_r^i)$, we get $\mathbf{d}^{i+1}$ and r^{i+1} from

$$\mathbf{p}_{\mathbf{d}}^i = \frac{1}{2}\left[(r^{i+1})^{2/3} + (r^i)^{2/3}\right] M\left(\frac{\mathbf{d}^{i+1} - \mathbf{d}^i}{\Delta t}\right) + \frac{\Delta t}{2}\nabla_{\mathbf{d}}U((r^i)^{1/3}\mathbf{d}^i) \tag{5.6}$$

$$\begin{aligned}p_r^i = &\mu\left(\frac{r^{i+1} - r^i}{\Delta t}\right) - \frac{\Delta t}{6}(r^i)^{-1/3}\left(\frac{\mathbf{d}^{i+1} - \mathbf{d}^i}{\Delta t}\right) \cdot \left[M\left(\frac{\mathbf{d}^{i+1} - \mathbf{d}^i}{\Delta t}\right)\right] \\ &+ \frac{\Delta t}{2}\left[\nabla_r U((r^i)^{1/3}\mathbf{d}^i) + \alpha\right].\end{aligned}$$

Then, to complete one step, the values of $\mathbf{p}_{\mathbf{d}}^{i+1}$ and p_r^{i+1} are explicitly obtained from

$$\mathbf{p}_{\mathbf{d}}^{i+1} = \frac{1}{2}\left[(r^{i+1})^{2/3} + (r^i)^{2/3}\right] M\left(\frac{\mathbf{d}^{i+1} - \mathbf{d}^i}{\Delta t}\right) - \frac{\Delta t}{2}\nabla_{\mathbf{d}}U((r^{i+1})^{1/3}\mathbf{d}^{i+1})$$

$$\begin{aligned}p_r^{i+1} = &\mu\left(\frac{r^{i+1} - r^i}{\Delta t}\right) + \frac{\Delta t}{6}(r^i)^{-1/3}\left(\frac{\mathbf{d}^{i+1} - \mathbf{d}^i}{\Delta t}\right) \cdot \left[M\left(\frac{\mathbf{d}^{i+1} - \mathbf{d}^i}{\Delta t}\right)\right] \\ &- \frac{\Delta t}{2}\left[\nabla_r U((r^{i+1})^{1/3}\mathbf{d}^{i+1}) + \alpha\right].\end{aligned} \tag{5.7}$$

Finally, the expression for the fourth-order shadow Hamiltonian associated with the real Hamiltonian $\mathcal{H}$ is (cf. (Akhmatskaya and Reich, 2008))

$$\begin{aligned}\tilde{H}^{[4]} = H + \frac{\Delta t^2}{24}\left\{2\mu\dot{R}R^{(3)} - \mu\ddot{R}^2 + 2R^{2/3}\dot{\mathbf{D}}\cdot[M\mathbf{D}^{(3)}] - R^{2/3}\ddot{\mathbf{D}}\cdot[M\ddot{\mathbf{D}}]\right\} \\ + \frac{\Delta t^2}{12}\left\{\left(\frac{4\ddot{R}}{3R^{1/3}} - \frac{4\dot{R}^2}{9R^{4/3}}\right)\dot{\mathbf{D}}\cdot[M\dot{\mathbf{D}}] - \frac{2}{3R^{1/3}}\dot{R}\dot{\mathbf{D}}\cdot[M\ddot{\mathbf{D}}]\right\},\end{aligned} \tag{5.8}$$

where $R(t)$ and $\mathbf{D}(t)$, analogously to $\mathbf{Q}(t)$ in (4.9), are the interpolation polynomials along numerical trajectories $\{r^i\}$ and $\{\mathbf{d}^i\}$, respectively.

It should be noticed here that the introduction of the Andersen barostat in GSHMC leads also to the modification of the partial momentum update step, namely, updating the *piston* momentum should be also included:

$$\begin{aligned} p_r^* &= \cos\varphi\ p_r + \sin\varphi\ u_r \\ u_r^* &= -\sin\varphi\ p_r + \cos\varphi\ u_r \end{aligned} \tag{5.9}$$

where $u_r \sim \mathcal{N}(0, \beta^{-1}\mu)$.

The complete algorithm for the NPT-GSHMC method now can be summarized as follows:

Algorithm 4 NPT Generalized Shadow Hybrid Monte Carlo

Input: M: mass matrix
Δt: time step
L: number of integration steps
Ψ: discretization scheme
N: number of MC iterations
T: temperature
$\varphi \in (0, \pi/2]$: noise angle
k: order of the shadow Hamiltonian
α: external pressure
μ: mass of the *piston*

1: initialize $(\mathbf{d}^0, r^0, \mathbf{p}_\mathbf{d}^0, p_r^0)$
2: **for** $n = 1, \ldots, N$ **do**
3: calculate the shadow Hamiltonian at $(\mathbf{d}, r, \mathbf{p}_\mathbf{d}, p_r) = (\mathbf{d}^{n-1}, r^{n-1}, \mathbf{p}_\mathbf{d}^{n-1}, p_r^{n-1})$
PMMC step
4: generate a proposal by the partial momentum update

$$\begin{aligned} \mathbf{p}^* &= \cos\varphi\ \mathbf{p} + \sin\varphi\ \mathbf{u} \\ \mathbf{u}^* &= -\sin\varphi\ \mathbf{p} + \cos\varphi\ \mathbf{u} \\ p_r^* &= \cos\varphi\ p_r + \sin\varphi\ u_r \\ u_r^* &= -\sin\varphi\ p_r + \cos\varphi\ u_r \end{aligned}$$

where $\mathbf{u} \sim \mathcal{N}(0, \beta^{-1}M)$ and $u_r \sim \mathcal{N}(0, \beta^{-1}\mu)$
5: calculate the shadow Hamiltonian at $(\mathbf{d}, r, \mathbf{p}^*, p_r^*)$

6: calculate the acceptance probability

$$\alpha_p = \min\left\{1, \frac{\exp\left(-\beta\left[\tilde{H}^{[k]}(\mathbf{d}, r, \mathbf{p}^*_{\mathbf{d}}, p^*_r) + \frac{1}{2}(\mathbf{u}^*)^T M^{-1}\mathbf{u}^* + \frac{1}{2\mu}(u^*_r)^2\right]\right)}{\exp\left(-\beta\left[\tilde{H}^{[k]}(\mathbf{d}, r, \mathbf{p}_{\mathbf{d}}, p_r) + \frac{1}{2}\mathbf{u}^T M^{-1}\mathbf{u} + \frac{1}{2\mu}u_r^2\right]\right)}\right\}$$

7: Modified Metropolis test

$$(\mathbf{d}, r, \bar{\mathbf{p}}_{\mathbf{d}}, \bar{p}_r) = \begin{cases} (\mathbf{q}, r, \mathbf{p}^*_{\mathbf{d}}, p^*_r) & \text{with probability } \alpha_p \\ (\mathbf{q}, r, \mathbf{p}_{\mathbf{d}}, p_r) & \text{otherwise} \end{cases}$$

MDMC step

8: generate a proposal by integrating Hamiltonian dynamics

$$(\mathbf{d}', r', \mathbf{p}'_{\mathbf{d}}, p'_r) = \Psi_{\Delta t, L}(\mathbf{d}, r, \bar{\mathbf{p}}_{\mathbf{d}}, \bar{p}_r)$$

9: calculate the shadow Hamiltonian at $(\mathbf{d}', r', \mathbf{p}'_{\mathbf{d}}, p'_r)$

10: calculate the acceptance probability[2]

$$\alpha_q = \min\left\{1, \exp\left(-\beta\left(\tilde{H}^{[k]}(\mathbf{d}', r', \mathbf{p}'_{\mathbf{d}}, p'_r) - \tilde{H}^{[k]}(\mathbf{d}, r, \bar{\mathbf{p}}_{\mathbf{d}}, \bar{p}_r)\right)\right)\right\}$$

11: Metropolis test

$$(\mathbf{d}^n, r^n, \mathbf{p}^n_{\mathbf{d}}, p^n_r) = \begin{cases} (\mathbf{d}', r', \mathbf{p}'_{\mathbf{d}}, p'_r) & \text{with probability } \alpha_q \\ \mathcal{F}(\mathbf{d}, r, \bar{\mathbf{p}}_{\mathbf{d}}, \bar{p}_r) & \text{otherwise} \end{cases}$$

12: compute the weight

$$w_n = \exp\left(-\beta\left(H(\mathbf{d}^n, r^n, \mathbf{p}^n_{\mathbf{d}}, p^n_r) - \tilde{H}^{[k]}(\mathbf{d}^n, r^n, \mathbf{p}^n_{\mathbf{d}}, p^n_r)\right)\right)$$

13: **end for**

14: calculate the average of an observable $\Omega(\mathbf{d}, r, \mathbf{p}_{\mathbf{d}}, p_r)$

$$\langle\Omega\rangle = \frac{\sum_{n=1}^N w_n \Omega_n}{\sum_{n=1}^N w_n}$$

It has to be remarked that the flip function $\mathcal{F}$ in step 11 is an extension of that in (2.14) to include the *piston* momenta:

$$\mathcal{F}(\mathbf{d}, r, \mathbf{p}_{\mathbf{d}}, p_r) = (\mathbf{d}, -\mathbf{p}_{\mathbf{d}}, r, -p_r).$$

A change of variable option aiming to increase a momenta acceptance rate is implemented in this algorithm as explained by Akhmatskaya and Reich, 2008.

At the end of simulation, reweighting of expectation values is performed as in (4.7) to recover the Boltzmann distribution.

In the next section, the implementation of this algorithm is explained in detail.

[2] Note that in this chapter we denote the acceptance probability of the MDMC step as α_q to avoid confusions with the external pressure α.

5.2.2 Implementation

We implemented the NPT-GSHMC method in the modified MultiHMC-GROMACS software package. Previously, the NVT-GSHMC had been implemented in MultiHMC-GROMACS (Escribano, Akhmatskaya, and Mujika, 2013), which helped us to perform a straightforward comparison of the accuracy and performance of both Hybrid Monte Carlo methodologies.

The Generalized Shadow Hybrid Monte Carlo (GSHMC) algorithm, provides a rigorous method for performing constant temperature simulations and can be served as a thermostat itself. To achieve a constant temperature and constant pressure simulation, one also needs to have the Andersen barostat at hand as well as the specific features of the NPT-GHSMC method implemented. No additional thermostat is required.

The Andersen barostat is not available in the released version of GROMACS though the MTTK, an Andersen-based barostat (Martyna et al., 1996), has been implemented there. This barostat must be combined with a Nosé-Hoover thermostat for running simulations in the NPT ensemble and it does not allow using a different thermostat, such as GSHMC. Thus, in principle, it could not serve our purposes, and it was necessary to implement the original formulation of Andersen barostat in MultiHMC-GROMACS. In practice, it means the implementation of a new symplectic and time-reversible integrator (5.6)-(5.7). For simplicity and consistency, the new integrator was introduced as a modification of the existing velocity Verlet algorithm.

Other modifications included:

- Evaluation of an NPT shadow Hamiltonian (5.8).
- Adding a new momentum update procedure (5.9), specific to the NPT-GSHMC algorithm.
- Adding new options to the *.mdp* configuration file.

Symplectic integrator: The symplectic time-reversible integrator has been extended to the case of the Andersen equations of motion. The updating scheme is the following (for further details the reader can consult (Kolb and Dünweg, 1999)).

We begin with performing a half step for the velocities:

- $\dot{\mathbf{q}}^{i+1/2} = \dot{\mathbf{q}}^{i} + \frac{\Delta t}{2}\frac{1}{M}F^{i}$, with the force F^i evaluated at the position $\mathbf{q}^i$,
- $\dot{\mathbf{d}}^{i+1/2} = \frac{\dot{\mathbf{q}}^{i+1/2}}{(r^i)^{1/3}}$,
- $\dot{r}^{i+1/2} = \dot{r}^{i} + \frac{\Delta t}{2}\frac{1}{\mu}(P-\alpha)$, with the pressure P evaluated, using the Virial theorem, taking the old positions $\mathbf{q}^i$, the old volume r^i but the already updated velocities $\dot{\mathbf{q}}^{i+1/2}$.

Then perform a full step for the positions:

- $r^{i+1/2} = r^{i} + \frac{\Delta t}{2}\dot{r}^{i+1/2}$,
- $\mathbf{q}^{i+1} = \mathbf{q}^{i} + \Delta t \frac{(r^i)^{2/3}}{(r^{i+1/2})^{2/3}}\dot{\mathbf{q}}^{i+1/2}$,
- $\mathbf{d}^{i+1} = \mathbf{d}^{i} + \Delta t\ \dot{\mathbf{d}}^{i+1/2}$,

- $r^{i+1} = r^{i+1/2} + \dfrac{\Delta t}{2}\dot{r}^{i+1/2}$.

Now two rescaling steps follow:

- $\dot{\mathbf{q}}^{i+1/2} = \dot{\mathbf{q}}^{i+1/2}\dfrac{(r^i)^{1/3}}{(r^{i+1})^{1/3}}$,
- $\mathbf{q}^{i+1} = \mathbf{q}^{i+1}\dfrac{(r^{i+1})^{1/3}}{(r^i)^{1/3}}$.

And finally we complete the full step for the velocities:

- $\dot{r}^{i+1} = \dot{r}^{i+1/2} + \dfrac{\Delta t}{2}\dfrac{1}{\mu}(P-\alpha)$, with the pressure P evaluated taking the new positions $\mathbf{q}^{i+1}$, the new volume r^{i+1} but the half-step velocities $\dot{\mathbf{q}}^{i+1/2}$,
- $\dot{\mathbf{q}}^{i+1} = \dot{\mathbf{q}}^{i+1/2} + \dfrac{\Delta t}{2}\dfrac{1}{M}F^{i+1}$, with the force F^{i+1} evaluated at the position $\mathbf{q}^{i+1}$,
- $\dot{\mathbf{d}}^{i+1} = \dfrac{\dot{\mathbf{q}}^{i+1}}{(r^{i+1})^{1/3}}$.

Since in the updating scheme above the dynamic value of volume q is changing, one has to make sure that the simulation box is also changing to fit this volume. In the code, it is done by re-scaling the box dimensions with the new value of the dynamic volume in the function `update_box()`. This implementation only applies to the case of a simulation box changing isotropically.

It is important to mention that the values of pressure and forces have to be updated every time step to make the integration scheme (5.6)-(5.7) working. This is done in the original version of the GROMACS code. However, the frequency of the pressure updates has to be specified by a user in the GROMACS parameter file. For using the NPT-GSHMC within the GROMACS code, such a parameter should always be set to 1. Such a choice does not introduce a critical computational overhead as can be seen from the numerical tests in the following section.

It is also noteworthy that GROMACS works with velocities instead of momenta. That is why the theoretical formulation (5.4) is slightly modified in the above scheme taking into account the relation between velocities and momenta.

The current version of the GROMACS software offers the velocity Verlet integrator. The new integrator (5.6)-(5.7) is placed in the same part of the code. The both GROMACS routines for updating positions and velocities need to be modified in the function `update_coords()`, but the modifications are straightforward, mainly related to a change of parameters of the subroutines.

There is also another important issue to consider: in GROMACS, when dealing with pressures, a rescaling factor is used (`PRESFAC` in the code). It has to be included in the time integration of equations of motion for the volume q, and in the calculation of the additional energy terms.

Shadow Hamiltonian: In order to introduce the NPT Shadow Hamiltonians (5.8) in the MultiHMC-GROMACS code, the shadow Hamiltonian implemented by Escribano, Akhmatskaya, and Mujika, 2013 can be taken as a starting point (details will be provided in Section 7.3). The shadow Hamiltonian appears in the subroutine `shadow()` as in the NVT implementation (cf. Section 7.3). As stated in Section 5.2.1, in the NPT ensemble

one has to consider a different shadow Hamiltonian (5.8) where the extended variable (the *piston* volume) introduces new terms. However, this modification does not entail a significant complexity since the NPT shadow Hamiltonians are calculated in a similar way as the NVT shadow Hamiltonians. Both types of shadow Hamiltonians are currently available in the code and can be chosen at runtime according to the parameters of the simulation. Thus, a user does not have to specify them.

Momentum refreshment: In comparison with the original GSHMC, the momentum refreshment procedure for the NPT-GSHMC also requires the update of the momentum p_r for the *piston*. This is a relatively simple extension of the original implementation. The algorithmic details can be found in Section 5.2.1.

Parameter file: GROMACS needs to receive two new parameters through the *.mdp* parameter file, the *piston* mass μ and the reference pressure α. As it was stated before, when tuning the parameter file, pressure updates have to be specified to be done for every time step. Additionally, the Andersen barostat has to be recognized as an isotropic pressure coupling method. These modifications were done in the standard way described in the GROMACS Developer's Guide. The specific parameters in the *.mdp* file look like this:

```
; Andersen barostat =
Pcoupl                  = Andersen;  Andersen / no
Pcoupltype              = isotropic; isotropic
mu_mass                 = 100;       any positive rational
alpha_press             = 1;         any positive rational
```

Details on GSHMC input parameters can be found in Section 7.3.

5.3 Results

We tested the NPT-GSHMC method by comparing it with the NVT-GSHMC implementation (Escribano, Akhmatskaya, and Mujika, 2013) and NPT-MD using the v-rescale thermostat (Bussi, Donadio, and Parrinello, 2007), the Parrinello-Rahman barostat (Parrinello and Rahman, 1981) and the position leapfrog integrator (as required by the chosen barostat). The same code, MultiHMC-GROMACS, with the appropriate choice of parameters for each case was used for running all three simulations.

As testing systems, we chose the coarse-grained toxin and the atomistic villin. Both systems have been described in Section 3.5.2. In the coarse-grained system, integration time step was set to 20 fs for optimal accuracy and 30 fs for optimal sampling. Both Coulomb and van der Waals interactions were defined as in Section 3.5.2. For the NPT-GSHMC particular case, we used the angle φ equal to 0.32 and the trajectory length L equal to 100. The reference temperature was 310 K. The *piston* mass μ was set to 100 and the reference temperature α was 1 bar. The integration time step for villin was set to the standard 1 fs in all cases. Coulomb and van der Waals interactions and periodic boundary conditions were considered as in the previous system. However, for villin, the hydrogen bonds were converted to constraints and the constraint algorithm used was LINCS (Hess et al., 1997). The specific NPT-GSHMC parameters were the same as taken for the coarse-grained system but with the angle φ equal to 0.4 and the reference temperature equal to 300 K.

5.3.1 Accuracy

In order to test the accuracy of the new method, we calculate averages for several thermodynamic observables in similar simulations with the three methods. As it was discussed before (see Section 4.2), the simulations involving the GSHMC method need to reweight statistical averages to compensate for the disturbance introduced by the use of shadow Hamiltonians.

In the case of the toxin system, 30 ns simulations were performed with an integrator time step of 20 fs, with the target temperature of 310 K and the target pressure of α=1 bar. It should be noted that the efficiency and precision of all three methods can vary according to several tuning parameters. In Table 5.1 typical results for all methods are shown with a set of parameters chosen for optimizing the accuracy of results.

	averages				**acc. rates**	
simulation	d (nm)	T (K)	P (bar)	U_{pb} (kJ mol^{-1})	A_r(%)	A_p(%)
NPT-GSHMC	2.3±0.4	308.5±0.3	1.2±0.5	-16.3±2.0	97	83
NVT-GSHMC	2.4±0.3	308.4±0.1	–	-14.9±0.6	100	85
NPT-MD	2.4±0.4	309.9±0.1	0.6±0.4	-15.8±0.2	–	–

Table 5.1: Toxin. Statistical averages.

We choose to monitor four properties of the toxin system: (i) the distance d traveled by the toxin from the centre of the membrane to the preferred location at the surface of the membrane; (ii) the temperature T; (iii) the pressure P; and (iv) the Coulomb energy between the protein and the bilayer U_{pb}. For the GSHMC methods, the reweighted averages are given. All calculated properties are in a good agreement. Error estimates correspond to the standard deviation as provided by GROMACS.

The Coulomb potential energy between the protein and the bilayer has been measured before for similar coarse-grained simulations, with resulting values close to -16 kJ/mol (Wee et al., 2008), which is consistent with our results (see Table 5.1). The Andersen barostat shows slightly more accurate pressure than the Parrinello-Rahman in our tests. These particular systems exhibit pressure oscillations of considerable amplitude, so we consider that the reported values for both barostats are sufficiently accurate. The NVT-GSHMC has no pressure coupling, so the measured average is disregarded.

Table 5.2 shows the test results for the villin system. Simulations were run for 1 ns with a time step of 1 fs, the target temperature of 300 K and the reference pressure of 1 bar. The observed average temperatures, T, and dihedral potential energies, U_{dih}, agree well for all simulation methods. Similar average values of pressure are achieved with both NPT simulations, NPT-MD and NPT-GSHMC.

	averages			**acc. rates**	
simulation	T (K)	P (bar)	U_{dih} (kJ mol^{-1})	A_r(%)	A_p(%)
NPT-GSHMC	299.5±0.7	1.4±0.9	276±1	95	94
NVT-GSHMC	299.9±0.9	–	276±3	100	97
NPT-MD	299.9±0.7	1.7±0.4	282±1	–	–

Table 5.2: Villin. Statistical averages.

5.3.2 Sampling

The GSHMC method and the Andersen barostat have several tuning parameters that can affect their performance. The two most important parameters in the case of GSHMC are the length of the MD trajectories L and the angle in the partial momentum update procedure φ. When the length of trajectories is too long, the gain over MD in terms of sampling efficiency is less noticeable. But if the length is too short, then the computational time spent on frequent calculations of shadow Hamiltonians becomes too long. The value of φ must be between 0 and $\pi/2$. If it is too small, then the temperature coupling might be too weak, but larger values interfere with the dynamics and can yield very low acceptance rates. The optimal values for L and φ are usually found through trial and error. Other parameters such as a time step used in the integrator, the order of shadow Hamiltonians or the type of momentum flip upon rejection are discussed elsewhere (see for instance (Escribano, Akhmatskaya, and Mujika, 2013; Wagoner and Pande, 2012)).

The Andersen barostat introduces two additional parameters: the mass of the *piston* μ and the reference or target pressure α. The reference pressure is used in the integrator for updating the *piston* velocity (see Section 5.2.2), as well as in the additional potential energy term in the Hamiltonian (5.3). When simulating biological experiments, this pressure is commonly set to 1 bar. μ represents the inertial mass of the extended coordinate and has a strong influence on the performance of the barostat. Figure 5.1 shows the effect of μ on the amplitude and frequency of the total energy of the villin system. Small *piston* masses can lead to wild oscillations in volume that could not only cause stability problems but also keep simulation from reaching its target pressure. But if the *piston* mass is too big, then the volume of the box barely changes and an NVT simulation is recovered with a pressure that very slowly tends to α. For a complete discussion on an optimal choice of μ see (Andersen, 1980; Kolb and Dünweg, 1999).

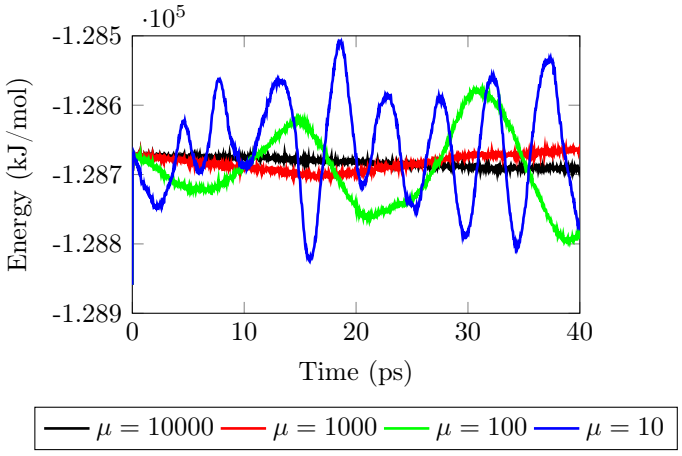


FIGURE 5.1: Villin. Total energy oscillations using NPT-GSHMC with varying *piston* masses μ.

One of the most important advantages of using GSHMC instead of standard MD is the noticeable improvement in sampling efficiency (Akhmatskaya and Reich, 2008; Escribano, Akhmatskaya, and Mujika, 2013; Wee et al., 2008). The distance traveled by the toxin towards the POPC bilayer in the coarse-grained system was measured to test the efficiency gain of the new NPT-GSHMC method. In Figure 5.2 the time evolution of this distance and

the corresponding autocorrelation functions are shown for the three methods. In general, GSHMC methods are expected to decorrelate faster and hence sample better. In this case, both NVT and NPT-GSHMC arrive together at the ∼2.48 nm distance (the position of the bilayer) in approximately half the time required by NPT-MD. This performance is consistent with the previous work (Wee et al., 2008).

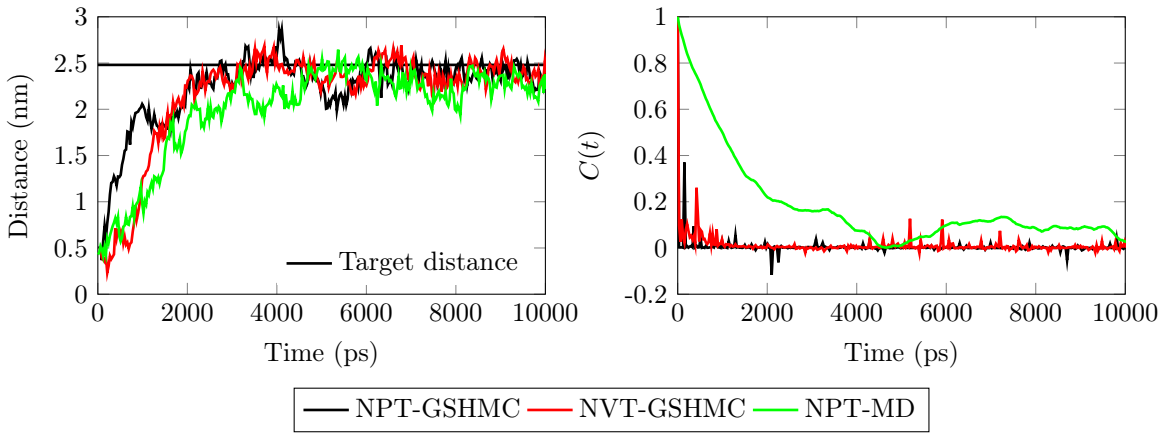


FIGURE 5.2: Toxin. Comparison for the time evolution of the distance traveled by the toxin towards the membrane bilayer with the three different methods (left) and the autocorrelation function for said distance (right).

A better way to measure the sampling efficiency is to calculate the integrated autocorrelation function IACF for distance d during the equilibration phase of the simulation (see for example (Kennedy and Pendleton, 2001)). Lower values of IACF indicate lower correlations and hence better sampling. The values obtained for this case with different simulation methods are shown in Table 5.3, which correspond to the IACF for the first 5000 ps of simulation. In this particular case, the integrator time step was set to 30 fs for optimal sampling efficiency. It is clear that both GSHMC methods outperform MD.

	NPT-GSHMC	**NVT-GSHMC**	**NPT-MD**
IACF	1.9	4.7	21.4

TABLE 5.3: Toxin. Integrated autocorrelation for toxin-bilayer distance.

Another way to test the sampling efficiency of the new method is to plot Ramachandran histograms (Ramachandran, Ramakrishnan, and Sasisekharan, 1963) for the amino acid residues in the villin system. These histograms show how the $\phi-\psi$ phase space of a particular residue is explored during the simulation. As a representative example, Figure 5.3 compares the resulting plots for the Met13 residue extracted from a 1 ns simulations using the all three simulation techniques. One can immediately see that both GSHMC methods are exploring a larger portion of the configurational space compared with MD. Most other residues show a similar improvement in sampling efficiency and several examples have been included in the Supplementary Material of (Fernández-Pendás et al., 2014).

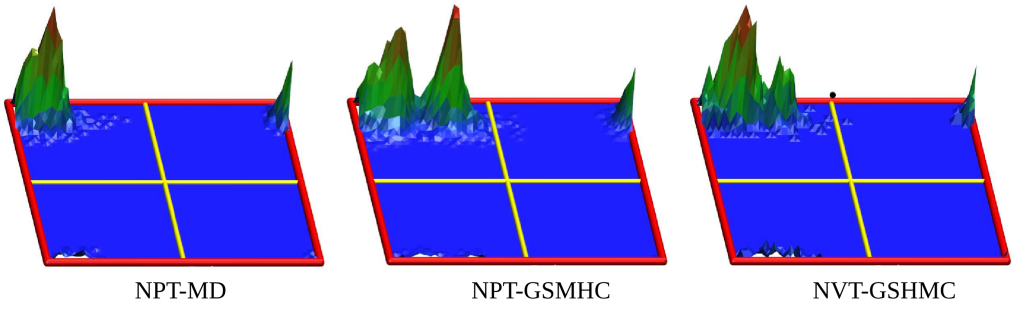


FIGURE 5.3: Villin. Ramachandran plots for the Met13 dihedral. Left: NPT-MD; Middle: NPT-GSHMC; Right: NVT-GSHMC

As a final comparison, it is necessary to weight the computational expense introduced by the Andersen barostat. A way to do so is by comparing the computational times employed to complete a 30 ns simulation of the toxin system and a 1 ns simulation of the villin system using an 8 processor node. The results, in this case, confirm what was measured previously for NVT-GSHMC implementation (Escribano, Akhmatskaya, and Mujika, 2013). The NVT-GSHMC method introduces on average an additional 2-4% computational overhead compared to the NPT-MD simulation. The NPT-GSHMC takes approximately the same computational time as NVT-GSHMC, which comes to show that the Andersen barostat implementation introduces almost no overhead and is fully compatible with the MPI parallelization in GROMACS. See Table 5.4 for a comparison of computational times.

	toxin		**villin**	
simulation	time (s)	ns/day	time (s)	ns/day
NPT-GSHMC	5766	749	11222	7.69
NVT-GSHMC	5747	751	11550	7.48
NPT-MD	5645	765	11087	7.79

TABLE 5.4: Comparison of computational times for all methods.

5.4 Conclusions

The GSHMC method has been adapted to the NPT ensemble using an Andersen barostat and implemented in the in-house software MultiHMC-GROMACS. The implementation has been tested against the NPT-MD and NVT-GSHMC simulation methods available in the MultiHMC-GROMACS suite (Escribano, Akhmatskaya, and Mujika, 2013). NPT-GSHMC shows the same level of accuracy as demonstrated by NPT-MD and NVT-GSHMC in the calculation of the thermodynamic properties of the tested systems, such as the toxin in a POPC bilayer and the protein villin at constant pressure and temperature.

The NPT-GSHMC method has also been proven to achieve a comparable sampling efficiency to NVT-GSHMC, as was expected from the theoretical formulation. The introduction of a barostat does not limit the benefits over MD that were previously obtained by the use of NVT-GSHMC.

The method does not introduce any noticeable computational load and is fully compatible with the highly optimized parallelization for multiple processors and threads already available in GROMACS.

In summary, all advantages offered by the Generalized Shadow Hybrid Monte Carlo methods, such as rigorous temperature control, sampling efficiency, are now available in MultiHMC-GROMACS for simulation of real-life experiments at constant pressure and constant temperature without a loss of computational efficiency.

Grand Canonical Ensemble

5.5 Introduction

In the Grand Canonical (GC) ensemble the chemical potential μ, the volume V and the temperature T are fixed while the number of particles is allowed to change. The GC ensemble describes the possible states of a system of volume V surrounded by a large "open" heat bath, meaning that both heat and matter can be transported across the walls of the system. Thus, the thermodynamic variables that characterize the system are V, T and μ. That is why the GC ensemble is also called μVT. The number of particles D is not fixed, and it is allowed to fluctuate around a mean value $\bar{D}$. The GC ensemble is as useful as the isothermal-isobaric and canonical ensembles are; numerous physical situations correspond to a system in which the particles number varies. These include liquid-vapor equilibrium, capillary condensation, and, notably, molecular electronics and batteries, in which a device is assumed to be coupled with an electron source. In computational molecular design, one seeks to sample a complete "chemical space" of compounds in order to optimize a particular property (e.g., binding energy to a target), which requires varying both the number and chemical identity of the constituent atoms.

Many Monte Carlo (MC) methods for simulation in the GC ensemble have been proposed in the literature (Norman and Filinov, 1969; Adams, 1974; Rowley, Nicholson, and Parsonage, 1975; Yao, Greenkorn, and Chao, 1982) since MC is the easiest approach for dealing with the change of the number of particles. The useful summary of MC in the GC ensemble can be found in (Allen and Tildesley, 1989). Later, molecular dynamics (MD) has been adapted to sampling in the GC ensemble (Cağin and Pettitt, 1991; Lo and Palmer, 1995).

In this chapter we propose three novel algorithms that sample the GC ensemble by combining generated MD trajectories with Metropolis Monte Carlo steps. Thus, they belong to the class of Hybrid Monte Carlo methods.

First, we formulate a GC version of the original Hybrid Monte Carlo (HMC) (Duane et al., 1987). Then we extend it to GC Generalized Hybrid Monte Carlo (GHMC) (Horowitz, 1991; Kennedy and Pendleton, 2001) and GC Generalized Shadow Hybrid Monte Carlo (GSHMC) (Akhmatskaya and Reich, 2008). All these methodologies were originally formulated in the NVT ensemble and later extended to NPT (cf. Section 5.2.1). However, this is the first time when the three methodologies are extended to the μVT ensemble.

Simulation in the grand canonical ensemble are performed in a box which is allowed to exchange particles with a reservoir. The box contains real particles whereas the reservoir is full of fictitious or ghost particles. In the proposed approach, we allow migrating the reservoir (or ghost) particles to the box to become real. The placement is done by a simple rescaling of the positions they had in the reservoir. This corresponds to an insertion of a new particle. Similarly, the box particles moving to the reservoir become fictitious and their positions are

rescaled accordingly. This corresponds to a deletion of an already existing particle. Thus, an insertion or deletion of any particle is equivalent to an exchange between the ghost and the real particles, meaning that a particle changes from one state to the other. The algorithms proposed here are limited to simulations of either homogeneous systems formed by one kind of particles, or such systems in which only one species is allowed to exchange with the reservoir. The ghost particles have the same mass and holonomic constraints as the real exchangeable particles, but they interact neither with each other nor with the real particles.

The way that the new HMC algorithms sample the GC ensemble has been inspired by the ideas of a seminal work by Norman and Filinov, 1969. This manuscript was originally published in the Soviet Union and hardly known in the western world at that time. It took some time before the new works exploring these ideas appeared (Adams, 1975; Rowley, Nicholson, and Parsonage, 1975; Yao, Greenkorn, and Chao, 1982). All these algorithms are pure MC methods, where no dynamics are performed. However, as will be shown later, some useful concepts introduced in those papers, such as a move definition or acceptance rules can be successfully applied to the new hybrid methods. Moreover, the drawbacks associated with the above Monte Carlo approaches can be reduced in the corresponding Hybrid Monte Carlo methods. Thus, the "memory" effect spotted in the approach of Rowley, Nicholson, and Parsonage, 1975 by Barker and Henderson, 1976 can be lessened dramatically by replacing local MC moves with global (larger) MD induced moves as suggested in Hybrid Monte Carlo methods. Indeed, the "memory" effect is caused by a high probability for a deleted particle become real again, due to its close location to the previous "real" position. Performing a global move should reduce such a probability significantly.

In the algorithms presented here, two kinds of transitions of states are considered: (ii) transitions in which the positions and momenta are changed, but the number of real particles is kept constant; and (ii) transitions in which the number of particles as well as the positions and momenta are changed. The first kind of moves is performed as in the canonical HMC methods (Duane et al., 1987; Horowitz, 1991; Kennedy and Pendleton, 2001; Akhmatskaya and Reich, 2008) where one runs molecular dynamics trajectories with a periodic resampling of the momenta. However, in (2) the number of particles changes during the MD trajectory. Thus, special attention has to be paid to the calculation of the Hamiltonian that, in this case, will be time-dependent (Stern, 2007). A particle to be inserted/deleted is chosen randomly among all the possibilities. The two kinds of transitions are accepted or rejected with Metropolis tests that ensure the sampling in the grand canonical ensemble. Details about the acceptance tests and the moves can be found in the following sections.

The chapter is structured as follows. In Section 5.6 we provide a short introduction into the relevant thermodynamic concepts where some useful and well-known quantities of interest are derived. References to the literature are also supplied. In Section 5.7.1 and 5.7.2 we explain in detail the main components of the Grand Canonical Hybrid Monte Carlo (GC-HMC) algorithm, namely the moves (Section 5.7.1) and the corresponding Metropolis tests (Section 5.7.2). The GC Hybrid Monte Carlo algorithm is presented in Sections 5.7.3. Its extension to a GC Generalized Hybrid Monte Carlo (GC-GHMC) algorithm, in which a partial momentum update procedure is used, is presented in Section 5.7.4. In Section 5.7.5, the importance sampling is introduced in the GHMC to construct a GC Generalized Shadow Hybrid Monte Carlo (GC-GSHMC) algorithm. Numerical results are shown in Section 5.8.

5.6 Thermodynamic considerations

We consider a D particles system with the vector of positions $\mathbf{q} \in \mathbb{R}^{3D}$ and the vector of momenta $\mathbf{p} \in \mathbb{R}^{3D}$. The Hamiltonian of the system has the usual separable form $H(\mathbf{q}, \mathbf{p}) = K(\mathbf{p}) + U(\mathbf{q})$, where K represents the kinetic energy and U the potential energy.

The canonical partition function in a volume V at temperature T is (cf. (Hill, 1960))

$$Q = \frac{1}{D! h^{3D}} \int_V \int_{-\infty}^{\infty} e^{-\beta H(\mathbf{q},\mathbf{p})} \mathrm{d}\mathbf{q}\, \mathrm{d}\mathbf{p},$$

where h is the Planck constant and $\beta = (k_B T)^{-1}$ with k_B being the Boltzmann constant. As in Section 2.1, if the integration over momenta is performed, then the canonical partition function becomes (2.1). In the absence of intermolecular forces, i.e., in the ideal gas scenario where $U(\mathbf{q}) = 0$, the partition function in (2.1) would read as

$$Q = \frac{V^D}{D! \Lambda^{3D}}.$$

For the same system, the grand canonical partition function is given by (cf. (Hill, 1960))

$$\begin{aligned} \Xi(\mu, V, T) = \sum_{D=0}^{\infty} e^{\beta\mu D} Q &= \sum_{D=0}^{\infty} \frac{1}{D! h^{3D}} \int_V \int_{-\infty}^{\infty} e^{\beta\mu D - \beta H(\mathbf{q},\mathbf{p})} \mathrm{d}\mathbf{q}\, \mathrm{d}\mathbf{p} \\ &= \sum_{D=0}^{\infty} \frac{1}{D! \Lambda^{3D}} \int_V e^{\beta\mu D - \beta U(\mathbf{q})} \mathrm{d}\mathbf{q}. \end{aligned} \tag{5.10}$$

Thus, the probability of observing the system in a state Γ_D that has D particles with positions $\mathbf{q}$ and momenta $\mathbf{p}$ is given by

$$\pi(\Gamma_D) = \frac{1}{\Xi} \frac{1}{D! h^{3D}} \exp\left(\beta\mu D - \beta H(\mathbf{q}, \mathbf{p})\right) \tag{5.11}$$

and the average of an observable F in the grand canonical ensemble is computed as (cf. (Hill, 1956))

$$\langle F(D, \mathbf{q}, \mathbf{p}) \rangle_{\mu VT} = \frac{1}{\Xi} \sum_{D=0}^{\infty} \frac{e^{\beta\mu D}}{D! h^{3D}} \int_V \int_{-\infty}^{\infty} F(D, \mathbf{q}) e^{-\beta H(\mathbf{q},\mathbf{p})} \mathrm{d}\mathbf{q}\, \mathrm{d}\mathbf{p}.$$

If one wants to study an observable $F(D, \mathbf{q})$ which is a function of the dimension D and the positions only, its ensemble average becomes

$$\langle F(D, \mathbf{q}) \rangle_{\mu VT} = \frac{1}{\Xi} \sum_{D=0}^{\infty} \frac{e^{\beta\mu D}}{D! \Lambda^{3D}} \int_V F(D, \mathbf{q}) e^{-\beta U(\mathbf{q})} \mathrm{d}\mathbf{q}. \tag{5.12}$$

If volume V is subdivided into a large number of K identical elementary cells, the configuration integral in (5.12) can be replaced by a sum, as $\mathrm{d}\mathbf{q}$ approaches the size of such cells (Rowley, Nicholson, and Parsonage, 1975; Yao, Greenkorn, and Chao, 1982).

Let us assume that the volume V' is much larger than a simulation box of volume V, i.e., $V' \gg V$ (Figure 5.4). Let D_{tot} be the total number of particles considered for the simulation. Only D of them are real and placed in the box of volume V. A number D_{tot} can and should be chosen big enough to ensure that there will never be lack of particles during a simulation

and that the sum in the partition function in (5.10) can be properly defined. We propose the change of variables $r_i = q_i/V_0$ for any particle i in a volume $V_0 = \{V, V'\}$. Then, it is convenient to switch to these reduced coordinates $\mathbf{r} = (r_1, r_2, \ldots, r_D, \ldots, r_{D_{\text{tot}}})$, such that the positions of the particles in the boxes of volume V and volume V' range from 0 to 1:

$$r_i = q_i/V \quad \forall i \in [1, D], \qquad r_j = q_j/V' \quad \forall j \in [D+1, D_{\text{tot}}]. \tag{5.13}$$

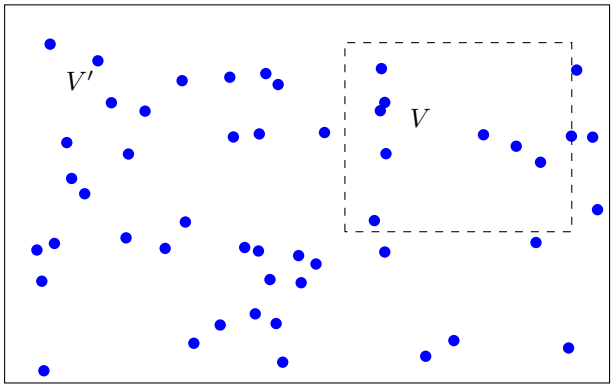


FIGURE 5.4: Real particles in volume V are surrounded by reservoir particles.

The $\mathbf{r}$ coordinates are convenient for simulation, whereas $\mathbf{q}$ are used for keeping track of the positions of the particles migrating between V' and V. Thus, using the new notations, the average (5.12) can be further simplified

$$\langle F(D, \mathbf{r}) \rangle_{\mu VT} = \sum_{D=0}^{K} F(D, \mathbf{r}) \nu(D, \mathbf{r}), \tag{5.14}$$

where

$$\nu(D, \mathbf{r}) = \frac{u(D, \mathbf{r})}{\sum_{D=0}^{K} u(D, \mathbf{r})} \tag{5.15}$$

and

$$u(D, \mathbf{r}) = \frac{\exp(\beta\mu D) V^D}{D! \Lambda^{3D}} \exp(-\beta U(\mathbf{r})).$$

The volume in this last expression appears due to the change of variables in (5.13). Note that the averages of the observable F in (5.14) are written in terms of $\mathbf{r}$.

It is easy to see that ν in (5.15) is the probability that defines the different states in a Markov chain of a Monte Carlo method in the grand canonical ensemble (Norman and Filinov, 1969; Adams, 1974; Adams, 1975; Rowley, Nicholson, and Parsonage, 1975; Yao, Greenkorn, and Chao, 1982). Thus, as in the conventional Metropolis-Hastings algorithm (Hastings, 1970), the acceptance or rejection criterion of the proposals can be written in terms of the probabilities ν. Therefore, the acceptance probability P_A of changing from a state Γ to a state Γ' is defined as

$$P_A(\Gamma \to \Gamma') = \min\left\{1, \frac{\nu_{\Gamma'}}{\nu_\Gamma}\right\}.$$

This probability can be calculated for two possible scenarios: either the number of particles changes or it remains constant. The latter is the usual NVT MC, but the former is more

complex and it has to be also split into two events: an insertion or a deletion of a particle. Both cases have to be studied separately. More specifically,

- in case of a deletion of a real particle, the proposed state Γ' corresponds to a decrease in the number of particles with respect to the current state Γ. Then, the new configuration would be accepted if

$$\frac{\nu_{\Gamma'}(D-1)}{\nu_{\Gamma}(D)} = \frac{D\Lambda^3}{\exp(\beta\mu)V}\exp\left(-\beta(U_{\Gamma'}-U_{\Gamma})\right) \geq \eta_1, \tag{5.16}$$

  where η_1 is a random number between $(0,1)$;

- in case of an insertion of a particle, the proposed state Γ' would be accepted if

$$\frac{\nu_{\Gamma'}(D+1)}{\nu_{\Gamma}(D)} = \frac{\exp(\beta\mu)V}{(D+1)\Lambda^3}\exp\left(-\beta(U_{\Gamma'}-U_{\Gamma})\right) > \eta_2, \tag{5.17}$$

  where η_2 is a random number between $(0,1)$.

Obviously, in both cases the potential energy always changes, even if the particles that are not affected by the insertion/deletion step are not moved (Yao, Greenkorn, and Chao, 1982). That is why the difference $U_{\Gamma'} - U_{\Gamma}$ appears in both acceptance rules (5.16) and (5.17).

In the situation when the particles are only allowed to move while their number is maintained constant, the proposed state Γ' is accepted if

$$\frac{\nu_{\Gamma'}(D)}{\nu_{\Gamma}(D)} = \exp\left(-\beta(U_{\Gamma'}-U_{\Gamma})\right) \geq \eta,$$

where η is a random number between $(0,1)$. This part is equivalent to the Metropolis test used in the regular Monte Carlo (MC) in the NVT ensemble (Allen and Tildesley, 1989), where particles are only allowed to move while their number D does not change.

If a proposal is not accepted, the system is maintained in the state Γ, as in a standard NVT MC.

5.6.1 Free energy estimation from the chemical potential

The essential thermal properties such as the Helmholtz free energy or the entropy can be estimated using the chemical potential. The free energy is related to the logarithm of the partition function. Thus, it is the generator through which other thermodynamic quantities are obtained via differentiation. Usually, we are interested in the free energy difference between two thermodynamic states, rather than in the absolute free energy. For instance, free energy differences tell whether a chemical reaction occurs spontaneously or requires an input of work, or whether a given solute is hydrophobic or hydrophilic. Free energy differences are directly related to equilibrium constants for chemical processes. Thus, from free energy differences, acid or base ionization constants can be computed. One might think that GC Monte Carlo methods are sampling absolute free energies, but it is not exactly the case. However, this type of algorithms allows for a direct estimation of the relative Helmholtz free energy A (see for more details (Adams, 1974; Adams, 1975; Barker and Henderson, 1976)).

As in GC simulations the chemical potential is maintained constant, one can equate the chemical potential of a molecule in an ideal gas at density ρ and the chemical potential of

the same species in an interacting system at density ρ'. This yields the relation between the chemical potential and Helmholtz free energy A (cf. (Hill, 1956)):

$$\frac{A}{D} = \mu - \frac{pV}{D}.$$

Here p is the pressure, which can be calculated using the virial theorem (cf. (Adams, 1975))

$$p = \rho k_B T + \frac{\text{vir}}{V}. \tag{5.18}$$

Therefore, it is clear that the relative Helmholtz free energy A is related to the chemical potential, the volume, the pressure and the number of particles, which are quantities calculated during a GC simulation.

5.7 Grand Canonical Hybrid Monte Carlo methods

5.7.1 The proposed moves

In the Hybrid Monte Carlo methods, the MD trajectories generate the proposed states that are accepted or rejected with a Metropolis test. Such trajectories are obtained by numerically solving the equations of motion using symplectic integrators, such as the standard velocity Verlet integrator. In the Grand Canonical HMC algorithms that we propose here, short MD trajectories of length L are run as in the original HMC (Duane et al., 1987). However, since the number of particles might change between two consecutive Monte Carlo steps, it is not trivial to model such trajectories. The possible solutions are shown in this section.

Following (Stern, 2007), we consider two main scenarios: an insertion/deletion occurs, or it does not. Thus, it leads to three possible *moves*:

1. An MD trajectory of D real particles while one of them is removed.
2. An MD trajectory of D real particles while a new particle is inserted making a reservoir particle become real.
3. An MD trajectory of D real particles.

(5.19)

The three moves are decided with probabilities α_r, α_i and α_m respectively. According to (Nicholson and Parsonage, 1982), the probabilities of insertion and removal α_i and α_r have to be equal to satisfy microscopic reversibility. In (Norman and Filinov, 1969) the authors advise to assign the same probability $\alpha_r = \alpha_i = \alpha_m = 1/3$ for the three possible moves. However, since no rationale was found behind such a choice but simply the empirical findings, a user may prefer to impose the only constraint $\alpha_r = \alpha_i$, and consider the value of α_m to be a tunable parameter of the algorithm. Obviously, a choice of this parameter may affect the sampling efficiency of the resulting algorithm.

If D were a continuous variable, the grand canonical ensemble could be understood as the isobaric ensemble (Barker and Henderson, 1976). However, in the current formulation of our algorithms D changes by at least ± 1. In fact, it seems to be an optimal number of changes in D, as, except for low densities and/or high temperatures, the probability of even such small changes is very low (Barker and Henderson, 1976). In (Norman and Filinov, 1969) the authors tried the moves in which more than one molecule was added or removed and found the most

probable value of ΔD to be ± 1. Thus, the attempts of moving multiple molecules turned to be the wasted labor (Barker and Henderson, 1976).

Whereas the move 3 in (5.19) is a typical proposal generated in the NVT HMC, i.e., an NVE MD trajectory, the moves 1 and 2 require more explanations. In the proposed algorithms, the positions for the insertion/deletion are chosen randomly. In case of insertion, a particle is selected randomly from the reservoir and then its insertion position in the simulation box is calculated as $r_i = q_i/V'$, where q_i is its position in the reservoir. In case of deletion, a particle is selected randomly from the simulation box and its position in the reservoir is calculated as $r_i = q_i/V$, where q_i is its current position in the box (see (5.13) and the discussion in Section 5.6 for details)[3]. If the inserted particle appears in a very populated area, the huge instabilities in the potential energies are expected and can lead to the explosion of the simulation. To avoid such a scenario, we suggest using a slow growth procedure in which a particle, once its position is decided, starts to grow progressively. If a particle is deleted, a progressive shrinking is proposed.

Let us consider a transition which generates a state $\hat{\Gamma}$ from an initial state Γ that has a different number of real particles. The slow-growth trajectory is run for a time τ, which is equivalent to l steps of the integration. During the slow-growth, the potential energy of the initial state Γ evolves to the potential energy of the state $\hat{\Gamma}$ with the different number of particles. Thus, we get a time-dependent Hamiltonian (cf. (Stern, 2007))

$$H_{\Gamma\to\hat{\Gamma}}(\lambda, \mathbf{r}, \mathbf{p}) = K(\mathbf{p}) + U_{\Gamma\to\hat{\Gamma}}(\lambda, \mathbf{r}),$$

where

$$\begin{aligned} U_{\Gamma\to\hat{\Gamma}}(\lambda = 0, \mathbf{r}) &= \tilde{U}(\Gamma, \mathbf{r}), \\ U_{\Gamma\to\hat{\Gamma}}(\lambda = \tau, \mathbf{r}) &= \tilde{U}(\hat{\Gamma}, \mathbf{r}), \\ U_{\Gamma\to\hat{\Gamma}}(\lambda, \mathbf{r}) &= U_{\hat{\Gamma}\to\Gamma}(\tau - \lambda, \mathbf{r}). \end{aligned}$$

$\tilde{U}(\Gamma, \mathbf{r})$ is the potential energy of the system at the state Γ, whereas $U_{\Gamma\to\hat{\Gamma}}(\lambda, \mathbf{r})$ is a corresponding time-dependent potential energy. The parameter λ is an integer in the interval $[0, \tau]$ which controls the time dependency of the Hamiltonian. The simplest possible definition for the time-dependent potential energy is a linear interpolation, such as

$$U_{\Gamma\to\hat{\Gamma}}(\lambda, \mathbf{r}) = \left(1 - \frac{\lambda}{\tau}\right)\tilde{U}(\Gamma, \mathbf{r}) + \left(\frac{\lambda}{\tau}\right)\tilde{U}(\hat{\Gamma}, \mathbf{r}), \tag{5.20}$$

though more sophisticated time-dependent potential energies could be considered. We consider here the one in (5.20).

Due to the reversibility of the definition of the time-dependent potential energy in (5.20) the insertion and deletion processes are equivalent. If the Velocity integrator is used then the

[3]The particles in the reservoir could be also moved after some steps, it might have a positive effect on the sampling.

positions and momenta at any time $t + \Delta t$, can be found as

$$\begin{aligned}
\mathbf{p}(t + \Delta t/2) &= \mathbf{p}(t) - \frac{\Delta t}{2}\nabla\left[\left(1 - \frac{t}{\tau}\right)\tilde{U}(\Gamma, \mathbf{r}(t)) + \left(\frac{t}{\tau}\right)\tilde{U}(\hat{\Gamma}, \mathbf{r}(t))\right] \\
\mathbf{r}(t + \Delta t) &= \mathbf{r}(t) + \Delta t\ \mathbf{p}(t + \Delta t/2) \\
\mathbf{p}(t + \Delta t) &= \mathbf{p}(t + \Delta t/2) - \frac{\Delta t}{2}\nabla\left[\left(1 - \frac{t + \Delta t}{\tau}\right)\tilde{U}(\Gamma, \mathbf{r}(t + \Delta t))\right. \\
&\qquad \left. + \left(\frac{t + \Delta t}{\tau}\right)\tilde{U}(\hat{\Gamma}, \mathbf{r}(t + \Delta t))\right].
\end{aligned}$$

Here Δt is the time step and the time-dependent energy is chosen according to (5.20). Once $t + \Delta t$ is equal to τ, the state $\hat{\Gamma}$ is generated. The last stage of the integration of the position and the momenta reads as

$$\begin{aligned}
\mathbf{r}(t + \Delta t) &= \mathbf{r}(\tau) = \mathbf{r}(t) + \Delta t\ \mathbf{p}(t + \Delta t/2) \\
\mathbf{p}(t + \Delta t) &= \mathbf{p}(\tau) = \mathbf{p}(t + \Delta t/2) - \frac{\Delta t}{2}\nabla\tilde{U}(\hat{\Gamma}, \mathbf{r}(t + \Delta t)).
\end{aligned}$$

After the time τ, $L - l$ steps are run with the regular velocity Verlet numerical integrator (cf. (3.5)) to generate a new state Γ'. The same time step Δt is used. There is no time-dependent potential energy at this point.

It is advised to take L significantly larger than l. It helps in the relaxation of the system and thus in the sampling, since more configurations will be accepted. Both L and l are tunable parameters of the algorithms.

The moves suggested here ensure that the system evolves globally and avoids the "memory" effect (Barker and Henderson, 1976), in which particles are always inserted and removed in the same areas.

Similarly to (2.4) in Section 2.1, the integrator described above can be generally written as

$$\begin{aligned}
\Psi_{\Delta t, L, l} &: \mathbb{R}^{6D} \to \mathbb{R}^{6D}, \\
&(\mathbf{r}, \mathbf{p}) \mapsto (\mathbf{r}', \mathbf{p}').
\end{aligned}$$

It is easy to see that the cases with no insertion or deletion imply $l = 0$.

5.7.2 GC-HMC: Metropolis tests

The Grand Canonical Hybrid Monte Carlo algorithms have to be equipped with proper Metropolis tests to ensure the sampling in the desired ensemble. Three Metropolis tests, corresponding to the moves described in (5.19), are derived in this section.

As in Section 2.1, in the HMC algorithms, the main difference with respect to pure MC is that the particles follow dynamical trajectories (cf. Section 5.7.1 for details), which are accepted or rejected. Thus, instead of only allowing local changes in the system, one performs a global move by integrating the equations of motion (Mehlig, Heermann, and Forrest, 1992). The conditional probability of a proposed configuration Γ' started at Γ is given by (cf. (2.5))

$$\rho_S(\Gamma \to \Gamma') = \rho_S(\mathbf{p}),$$

where the initial momenta are drawn from the Maxwell-Boltzmann distribution (2.6). The proposal probability depends only on the momenta.

The Metropolis-Hastings test for a proposed configuration Γ' initialized from Γ is (cf. Hastings, 1970)

$$P_A(\Gamma \to \Gamma') = \min\left\{1, \frac{\pi(\Gamma')\rho_S(\Gamma' \to \Gamma)}{\pi(\Gamma)\rho_S(\Gamma \to \Gamma')}\right\},$$

where π is the probability of observing the system in a state (5.11). Then, the three acceptance probabilities P_A are obtained:

- Move 1:
$$P_A(\Gamma_D \to \Gamma'_{D-1}) = \min\left\{1, \frac{Dh^3}{\exp(\beta\mu)V}\exp\left(-\beta(H_{\Gamma'} - H_\Gamma)\right)\right\}; \tag{5.21}$$
- Move 2:
$$P_A(\Gamma_D \to \Gamma'_{D+1}) = \min\left\{1, \frac{\exp(\beta\mu)V}{(D+1)h^3}\exp\left(-\beta(H_{\Gamma'} - H_\Gamma)\right)\right\}; \tag{5.22}$$
- Move 3:
$$P_A(\Gamma_D \to \Gamma'_D) = \min\left\{1, \exp\left(-\beta(H_{\Gamma'} - H_\Gamma)\right)\right\}. \tag{5.23}$$

It is easy to see that this acceptance rule is equivalent to the one used, for instance, by Boinepalli and Attard, 2003. As in Section 2.1, we will denote the acceptance probabilities in (5.21), (5.22) and (5.23) as α.

5.7.3 Grand Canonical Hybrid Monte Carlo

Though our ultimate goal is to develop a GSHMC algorithm for simulation in the grand canonical ensemble, we start with a formulation of Grand Canonical Hybrid Monte Carlo (GC-HMC). The algorithm is based on the canonical HMC (Duane et al., 1987), which has been explained in detail in Section 2.1. Two other algorithms presented in the following sections, GC-GHMC and GC-GSHMC, rely on this basic algorithm.

To keep track of changes from one state to other, we propose to assign to each particle i, of D_{tot} particles, an index g_i that can take the value 0 or 1 if the particle is in the reservoir or it is real, respectively. This index can be used to describe the changes from one state to the other. This is similar to the idea of occupancy by Rowley, Nicholson, and Parsonage, 1975. The sum of all of the indexes g_i equates to the number D of real particles that has to be always smaller or equal than D_{tot}.

The main steps of the algorithm are described below:

Algorithm 5 Grand Canonical Hybrid Monte Carlo

Input: Δt: time step
L: number of integration steps
l: number of slow-growth steps
α_r, α_i, α_m: probabilities of the moves
Ψ: discretization scheme
D: initial dimension of the system
D_{tot}: maximum number of particles used in the simulation
N: number of MC iterations
T: temperature

1: initialize $\mathbf{r}^0$, $D^0 = D$, $\mathbf{g}^0$, $l_0 = l$
2: **for** $n = 1, \ldots, N$ **do**

3: $l = l_0$
4: $\mathbf{r} = \mathbf{r}^{n-1}$
5: $D = D^{n-1}$, $\mathbf{g} = \mathbf{g}^{n-1}$
6: draw momenta $\mathbf{p}$ for ghost and real particles from Maxwell-Boltzmann distribution (2.6)
7: pick randomly a move from (5.19)
  draw $x \sim \mathcal{U}(0,1)$
  if $x < \alpha_r$
    pick randomly a particle i and $D' = D - 1$
  else if $x \geq \alpha_r$ and $x < \alpha_i + \alpha_r$
    pick randomly a particle i and $D' = D + 1$
  else
    $l = 0$, $D' = D$
  end if
8: generate a proposal by integrating Hamiltonian dynamics

$$(\mathbf{r}', \mathbf{p}') = \Psi_{\Delta t, L, l}(\mathbf{r}, \mathbf{p})$$

9: calculate the acceptance probability α using (5.21), (5.22) or (5.23)
10: Metropolis test
  draw $u \sim \mathcal{U}(0,1)$
  if $u < \alpha$
    accept: $\mathbf{r}^n = \mathbf{r}'$, $D^n = D'$
    if the move is insertion/deletion
      $g_i^n = 1 - g_i^{n-1}$
    end if
  else
    reject: $\mathbf{r}^n = \mathbf{r}$, $D^n = D$
  end if
11: discard momenta $\mathbf{p}'$, $\mathbf{p}$
12: **end for**

The step 6 in the algorithm above ensures that, in case of insertion, the new particle is initially drawn from the right distribution.

5.7.4 Grand Canonical Generalized Hybrid Monte Carlo

As discussed in Chapter 2, in the HMC algorithm the momenta are always completely resampled after the Metropolis test (step 4 in Algorithm 1). This also applies to the grand canonical ensemble version of HMC presented in Section 5.7.3. The partial momentum update as introduced in the Generalized Hybrid Monte Carlo method (Section 2.2) often helps to improve sampling efficiency of HMC. In order to take advantage of this feature in the HMC simulation in the grand canonical ensemble, we adapt the Generalized Hybrid Monte Carlo method to GC simulation.

The Grand Canonical GHMC (GC-GHMC) can be summarized as follows:

Algorithm 6 Grand Canonical Generalized Hybrid Monte Carlo

Input: M: mass matrix
  Δt: time step

L: number of integration steps
l: number of slow-growth steps
α_r, α_i, α_m: probabilities of the moves
Ψ: discretization scheme
D: initial dimension of the system
D_{tot}: maximum number of particles used in the simulation
N: number of MC iterations
T: temperature
$\varphi \in (0, \pi/2]$: noise angle

initialize $(\mathbf{r}^0, \mathbf{p}^0)$, $D^0 = D$, $\mathbf{g}^0$, $l_0 = l$
for $n = 1, \dots, N$ **do**
  $l = l_0$
  $(\mathbf{r}, \mathbf{p}) = (\mathbf{r}^{n-1}, \mathbf{p}^{n-1})$
  $D = D^{n-1}$, $\mathbf{g} = \mathbf{g}^{n-1}$
  partial momentum update for ghost and real particles

$$\begin{aligned} \mathbf{p}^* = &\quad \cos\varphi\ \mathbf{p} + \sin\varphi\ \mathbf{u} \\ \mathbf{u}^* = &-\sin\varphi\ \mathbf{p} + \cos\varphi\ \mathbf{u} \end{aligned}$$

  where $\mathbf{u} \sim \mathcal{N}(0, \beta^{-1}M)$
  pick randomly a move from (5.19)
    draw $x \sim \mathcal{U}(0,1)$
    if $x < \alpha_r$
      pick randomly a particle i and $D' = D - 1$
    else if $x \geq \alpha_r$ and $x < \alpha_i + \alpha_r$
      pick randomly a particle i and $D' = D + 1$
    else
      $l = 0$, $D' = D$
    end if
  generate a proposal by integrating Hamiltonian dynamics

$$(\mathbf{r}', \mathbf{p}') = \Psi_{\Delta t, L, l}(\mathbf{r}, \mathbf{p}^*)$$

  calculate the acceptance probability α using (5.21), (5.22) or (5.23)
  Metropolis test
    draw $u \sim \mathcal{U}(0,1)$
    if $u < \alpha$
      accept: $(\mathbf{r}^n, \mathbf{p}^n) = (\mathbf{r}', \mathbf{p}')$, $D^n = D'$
      if the move is insertion/deletion
        $g_i^n = 1 - g_i^{n-1}$
      end if
    else
      reject and flip momenta: $(\mathbf{r}^n, \mathbf{p}^n) = \mathcal{F}(\mathbf{r}, \mathbf{p}^*)$, $D^n = D$
    end if
end for

5.7.5 Grand Canonical Generalized Shadow Hybrid Monte Carlo

The objective of the Generalized Shadow Hybrid Monte Carlo (GSHMC) described in Section 4.3 is to maintain a high acceptance rate in the simulations. It is achieved by combining a partial momentum Monte Carlo step with the importance sampling with respect to a modified density such as (4.2). As it has repeatedly been said, the modified Hamiltonian is better preserved by a numerical integrator than the true Hamiltonian (see equations (4.4)-(4.5)). In GSHMC, the shadow Hamiltonians $\tilde{H}^{[k]}$ are used in the Metropolis tests instead of the true Hamiltonian (see step 10 in Algorithm 3). Therefore, the better conservation of the modified Hamiltonians leads to an improvement of the acceptance rate, which can be very beneficial in the grand canonical ensemble, where many insertion/deletion proposals are rejected. The Metropolis tests for GC-GSHMC can be derived by merely modifying (5.21)-(5.23) to replace Hamiltonians with modified Hamiltonians. Thus, depending on the type of move, the acceptance probability P_A for GC-GSHMC is defined as:

- Move 1:

$$P_A(\Gamma_{D-1} \to \Gamma'_D) = \min\left\{1, \frac{Dh^3}{\exp(\beta\mu)V} \exp\left(-\beta\left(\tilde{H}^{[k]}_{\Gamma'} - \tilde{H}^{[k]}_{\Gamma}\right)\right)\right\}; \tag{5.24}$$

- Move 2:

$$P_A(\Gamma_D \to \Gamma'_{D+1}) = \min\left\{1, \frac{\exp(\beta\mu)V}{(D+1)h^3} \exp\left(-\beta\left(\tilde{H}^{[k]}_{\Gamma'} - \tilde{H}^{[k]}_{\Gamma}\right)\right)\right\}; \tag{5.25}$$

- Move 3:

$$P_A(\Gamma_D \to \Gamma'_D) = \min\left\{1, \exp\left(-\beta\left(\tilde{H}^{[k]}_{\Gamma'} - \tilde{H}^{[k]}_{\Gamma}\right)\right)\right\}. \tag{5.26}$$

Again, for the sake of simplicity we will denote the acceptance probabilities as α.

We can summarize the Grand Canonical GSHMC (GC-GSHMC) algorithm as follows:

Algorithm 7 Grand Canonical Generalized Shadow Hybrid Monte Carlo

Input: M: mass matrix
Δt: time step
L: number of integration steps
l: number of slow-growth steps
α_r, α_i, α_m: probabilities of the moves
Ψ: discretization scheme
D: initial dimension of the system
D_{tot}: maximum number of particles used in the simulation
N: number of MC iterations
T: temperature
$\varphi \in (0, \pi/2]$: noise angle
k: order of the shadow Hamiltonian

1: initialize $(\mathbf{r}^0, \mathbf{p}^0)$, $D^0 = D$, $\mathbf{g}^0$, $l_0 = l$
2: **for** $n = 1, \ldots, N$ **do**
3: $l = l_0$
4: calculate the shadow Hamiltonian at $(\mathbf{r}, \mathbf{p}) = (\mathbf{r}^{n-1}, \mathbf{p}^{n-1})$
5: $D = D^{n-1}$, $\mathbf{g} = \mathbf{g}^{n-1}$
PMMC step

6: generate a proposal for ghost and real particles by the partial momentum update

$$\begin{aligned}\mathbf{p}^* = &\quad \cos\varphi\ \mathbf{p} + \sin\varphi\ \mathbf{u}\\ \mathbf{u}^* = &-\sin\varphi\ \mathbf{p} + \cos\varphi\ \mathbf{u}\end{aligned}$$

where $\mathbf{u} \sim \mathcal{N}(0, \beta^{-1}M)$

7: calculate the shadow Hamiltonian at $(\mathbf{r}, \mathbf{p}^*)$

8: calculate the acceptance probability

$$\alpha_p = \min\left\{1, \frac{\exp\left(-\beta\left(\tilde{H}^{[k]}(\mathbf{r},\mathbf{p}^*) + \frac{1}{2}(\mathbf{u}^*)^T M^{-1}\mathbf{u}^*\right)\right)}{\exp\left(-\beta\left(\tilde{H}^{[k]}(\mathbf{r},\mathbf{p}) + \frac{1}{2}\mathbf{u}^T M^{-1}\mathbf{u}\right)\right)}\right\}$$

9: Modified Metropolis test

$$(\mathbf{r}, \bar{\mathbf{p}}) = \begin{cases}(\mathbf{r}, \mathbf{p}^*) & \text{with probability } \alpha_p\\ (\mathbf{r}, \mathbf{p}) & \text{otherwise}\end{cases}$$

MDMC step

10: pick randomly a move from (5.19)

draw $x \sim \mathcal{U}(0,1)$
if $x < \alpha_r$
pick randomly a particle i and $D' = D - 1$
else if $x \geq \alpha_r$ and $x < \alpha_i + \alpha_r$
pick randomly a particle i and $D' = D + 1$
else
$l = 0$, $D' = D$
end if

11: generate a proposal by integrating Hamiltonian dynamics

$$(\mathbf{r}', \mathbf{p}') = \Psi_{\Delta t, L, l}(\mathbf{r}, \bar{\mathbf{p}})$$

12: calculate the shadow Hamiltonian at $(\mathbf{r}', \mathbf{p}')$

13: calculate the acceptance probability α using (5.24), (5.25) or (5.26)

14: Metropolis test

draw $u \sim \mathcal{U}(0,1)$
if $u < \alpha$
accept: $(\mathbf{r}^n, \mathbf{p}^n) = (\mathbf{r}', \mathbf{p}')$, $D^n = D'$
if the move is insertion/deletion
$g_i^n = 1 - g_i^{n-1}$
end if
else
reject and flip momenta: $(\mathbf{r}^n, \mathbf{p}^n) = \mathcal{F}(\mathbf{r}, \bar{\mathbf{p}})$, $D^n = D$
end if

15: compute the weight

$$w_n = \exp\left(-\beta\left(H(\mathbf{r}^n, \mathbf{p}^n) - \tilde{H}^{[k]}(\mathbf{r}^n, \mathbf{p}^n)\right)\right)$$

16: **end for**
17: calculate the average of an observable $\Omega(\mathbf{r}, \mathbf{p})$

$$\langle \Omega \rangle = \frac{\sum_{n=1}^{N} w_n \Omega_n}{\sum_{n=1}^{N} w_n}$$

We remark that a modified Metropolis test in step 8 ensures that momenta are drawn with respect to the correct distribution. The chemical potential does not play any role since the number of particles does not change during the momentum update. Also, in the same step, for the computation of the extended Hamiltonians $\hat{H}(\mathbf{r}, \mathbf{p}, \mathbf{u}) = \tilde{H}(\mathbf{r}, \mathbf{p}) + \frac{1}{2}\mathbf{u}^T M^{-1}\mathbf{u}$ only the noise $\mathbf{u}$ corresponding to the real particles is considered. The partial momentum update is applied to both real and ghost particles, so in case of insertion, the new particle would follow the right distribution. However, since the modified Hamiltonian $\tilde{H}(\mathbf{r}, \mathbf{p})$ is computed only for real particles, the "kinetic" term $\frac{1}{2}\mathbf{u}^T M^{-1}\mathbf{u}$ is calculated in the same dimension, i.e., D. Obviously, the same applies to $\hat{H}(\mathbf{r}, \mathbf{p}^*, \mathbf{u}^*)$.

5.8 Results

We have tested the algorithms proposed in Section 5.7 with Lennard-Jones fluids, aiming to compare the obtained results with the data previously presented in the literature. We consider the potential energy between two Lennard-Jones molecules at a center-to-center distance r as

$$U(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right], \tag{5.27}$$

where ϵ and σ are the parameters characterizing the fluid. All the simulations in this section have been performed with a truncated and shifted potential (Smit, 1992), and the results are reported in reduced units (details can be found in Appendix C.1). The cutoff distance was set to 2.5σ and periodic boundary conditions were imposed in all three dimensions.

The thermodynamic properties of interest are density ρ and pressure p. While ρ is calculated as D/V, p is computed using (5.18) and the details of its computation are explained in Appendix C.2.) for different temperatures T and chemical potentials μ. Similar experiments have been previously performed in (Adams, 1979; Yao, Greenkorn, and Chao, 1982; Lo and Palmer, 1995) using Monte Carlo or molecular dynamics.

Following (Lo and Palmer, 1995), we have investigated two temperatures, $T^* = 0.769$ and $T^* = 1.0$. For different chemical potentials and temperatures, the densities and pressures, measured during simulations with three new algorithms, were compared with the values obtained from the Nicolas equation of state (EOS) (Nicolas et al., 1979; Johnson, Zollweg, and Gubbins, 1993). Since studying the vapor-liquid equilibria is of interest in many applications, we chose the simulation points on both sides of the vapor-liquid coexistence curve (Lin, Blanco, and Goddard III, 2003). The investigated points are shown in Figure 5.5.

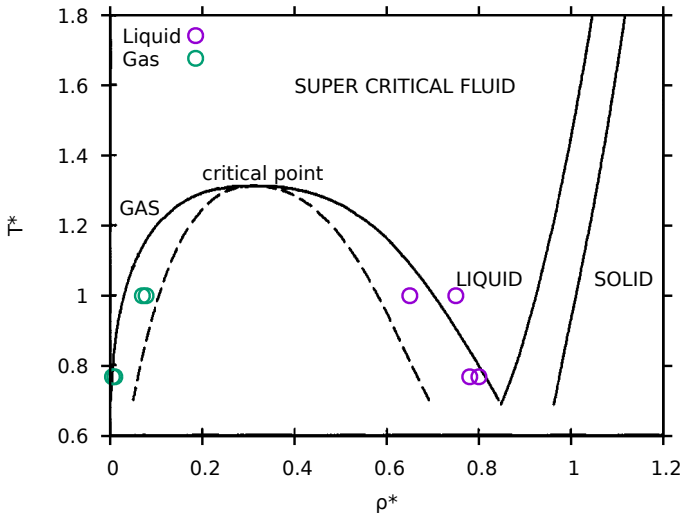


FIGURE 5.5: Phase diagram of Lennard-Jones systems. The investigated thermodynamic points are plotted for both liquid and gas phases. The plot was taken from (Lin, Blanco, and Goddard III, 2003) and adapted to this study.

All results presented in this section were calculated from the simulations of length $t^* = 50000$. Our aim was (i) to validate the novel algorithms through comparison with the EOS predicted data; and (ii) to compare accuracy and performance of the novel algorithms between themselves. In addition, we also compare the simulations with experiments performed with the GC-MC method by Yao, Greenkorn, and Chao, 1982.

At each thermodynamic point in the liquid branch, the same integration time steps were used for GC-HMC, GC-GHMC and GC-GSHMC. For the thermodynamic points in which $T^* = 0.769$ and for the case with $T^* = 1.00$ and $\mu^* = -2.852$, the reduced time step was $\Delta t^* = 0.01$. For $T^* = 1.00$ and $\mu^* = -1.757$, a bigger time step was possible, namely $\Delta t^* = 0.2$. In all cases, the trajectory lengths were set to $L = 500$ and $l = 100$. In the case of GC-GSHMC and GC-GHMC, the angle φ in the partial momentum update (cf. Step 6 in Algorithm 6 and step 6 in Algorithm 7) was assigned to 0.3. The fourth order shadow Hamiltonians (4.10) were used with GC-GSHMC. For all considered methods, the probabilities of the three different moves where chosen to be $1/3$, as suggested in Section 5.7.1.

The results obtained using GC-MC, GC-HMC, GC-GHMC and GC-GSHMC along the liquid branch are shown in Table 5.5. For clarity, the chosen thermodynamic points (temperatures and chemical potentials) are presented in the dashed column whereas the reference values obtained from the EOS are shown right next to their equivalent values obtained in the simulations.

method	conditions		properties			
	T^*	μ^*	ρ^*	ρ^*(EOS)	p^*	p^*(EOS)
GC-GSHMC	0.769	-2.999	0.780 ± 0.000	0.780	0.343 ± 0.022	0.324
	0.769	-2.727	0.800 ± 0.002	0.800	0.533 ± 0.015	0.539
	1.00	-2.852	0.650 ± 0.001	0.650	0.272 ± 0.055	0.226
	1.00	-1.757	0.750 ± 0.001	0.750	0.976 ± 0.025	0.999
GC-GHMC	0.769	-2.999	0.780 ± 0.002	0.780	0.291 ± 0.045	0.324
	0.769	-2.727	0.800 ± 0.002	0.800	0.466 ± 0.075	0.539
	1.00	-2.852	0.651 ± 0.019	0.650	0.244 ± 0.032	0.226
	1.00	-1.757	0.750 ± 0.001	0.750	0.939 ± 0.052	0.999
GC-HMC	0.769	-2.999	0.780 ± 0.003	0.780	0.222 ± 0.122	0.324
	0.769	-2.727	0.800 ± 0.002	0.800	0.453 ± 0.081	0.539
	1.00	-2.852	0.654 ± 0.009	0.650	0.236 ± 0.011	0.226
	1.00	-1.757	0.751 ± 0.005	0.750	0.927 ± 0.68	0.999
GC-MC	0.769	-2.999	0.780 ± 0.005	0.780	0.231 ± 0.113	0.324
	0.769	-2.727	0.803 ± 0.002	0.800	0.532 ± 0.015	0.539
	1.00	-2.852	0.640 ± 0.012	0.650	0.204 ± 0.032	0.226
	1.00	-1.757	0.754 ± 0.005	0.750	1.015 ± 0.106	0.999

TABLE 5.5: Liquid branch densities ρ^* and pressures p^* at given temperatures T^* and chemical potentials μ^* calculated using GC-MC, GC-HMC, GC-GHMC and GC-GSHMC. The densities and pressures obtained from the Nicolas equation of state (EOS) are also reported.

All methods produce results that are close to the expected reduced densities and pressures. The best agreement with the EOS values is found, in general, with the GC-GSHMC algorithm.

The sampling performance is also compared for the four methods. The time-normalized effective sample size (ESS) (Geyer, 1992) and the acceptance rates are shown in Table 5.6. From now on we call a time-normalized ESS simply ESS. The ESS are calculated for the potential energy. The relative ESS with respect to the GC-MC ESS are reported. The aim is to demonstrate the improvement in sampling obtained with the new methods presented in this chapter. For simplicity, only a few of the states considered above in Table 5.5 are presented now. However, the similar trends were observed in the other thermodynamics states.

method	conditions T^*	conditions μ^*	properties α (%)	properties ESS
GC-GSHMC	0.769	-2.999	46.60	9.92
	1.00	-1.757	51.28	12.08
GC-GHMC	0.769	-2.999	32.70	7.96
	1.00	-1.757	31.47	10.04
GC-HMC	0.769	-2.999	32.20	4.84
	1.00	-1.757	31.24	7.96
GC-MC	0.769	-2.999	32.33	1
	1.00	-1.757	28.54	1

TABLE 5.6: Liquid branch acceptance rates (α) and effective sample sizes (ESS) observed in GC-MC, GC-HMC, GC-GHMC and GC-GSHMC simulations. ESS was normalized with respect to the data obtained with GC-MC for given T^* and chemical potentials μ^*.

The low acceptance rates in all presented simulations are due to the number of rejections of the insertion/deletion moves. As expected, GC-GSHMC demonstrates the highest acceptance rates due to the better conservation of the shadow Hamiltonians included in the Metropolis test, even in the cases of insertion/deletion. Higher order shadow Hamiltonians could be used to improve the acceptance rates. The fact that GC-MC produces the poorest sampling is not surprising. It is well known that HMC reduces the correlation between successive sampled states with respect to MC by using the Hamiltonian dynamics for better exploration of the phase space. This leads to a quicker convergence to the desired distribution. Finer tuning of the parameters φ, L or l could improve sampling performance of all HMC-based methods.

We performed the similar tests at different thermodynamic points in the gas branch. Again, at each thermodynamic point, the same integration time steps were used for GC-HMC, GC-GHMC and GC-GSHMC. The reduced time step was $\Delta t^* = 0.01$ for the cases with $T^* = 0.769$ and $\Delta t^* = 0.02$ otherwise. In all cases, the trajectory lengths where set to $L = 500$ and $l = 100$. In the case of GC-GSHMC and GC-GHMC, the angle φ was 0.3. As in the liquid case, the probabilities of the three different moves were always assigned to 1/3, and the fourth order shadow Hamiltonians (4.10) were used with GC-GSHMC. The results observed with GC-MC, GC-HMC, GC-GHMC and GC-GSHMC along the gas branch are shown in Table 5.7.

method	conditions		properties			
	T^*	μ^*	ρ^*	ρ^*(EOS)	p^*	p^*(EOS)
GC-GSHMC	0.769	-4.127	0.0050 ± 0.0000	0.0050	0.0035 ± 0.0002	0.0037
	0.769	-3.646	0.0100 ± 0.0000	0.0100	0.0069 ± 0.0001	0.0071
	1.00	-3.200	0.0700 ± 0.0009	0.0700	0.0504 ± 0.0002	0.0515
	1.00	-3.150	0.0744 ± 0.0037	0.0780	0.0559 ± 0.0005	0.0552
GC-GHMC	0.769	-4.127	0.0050 ± 0.0000	0.0050	0.0035 ± 0.0002	0.0037
	0.769	-3.646	0.0100 ± 0.0000	0.0100	0.0067 ± 0.0001	0.0071
	1.00	-3.200	0.0697 ± 0.0012	0.0700	0.0501 ± 0.0002	0.0515
	1.00	-3.150	0.0741 ± 0.0064	0.0780	0.0561 ± 0.0029	0.0552
GC-HMC	0.769	-4.127	0.0052 ± 0.0001	0.0050	0.0033 ± 0.0005	0.0037
	0.769	-3.646	0.0102 ± 0.0004	0.0100	0.0065 ± 0.0016	0.0071
	1.00	-3.200	0.0698 ± 0.0011	0.0700	0.0499 ± 0.0032	0.0515
	1.00	-3.150	0.0740 ± 0.0037	0.0780	0.0562 ± 0.0153	0.0552
GC-MC	0.769	-4.127	0.0052 ± 0.0005	0.0050	0.0035 ± 0.0002	0.0037
	0.769	-3.646	0.0097 ± 0.0053	0.0100	0.0067 ± 0.0003	0.0071
	1.00	-3.200	0.0694 ± 0.0012	0.0700	0.0494 ± 0.0127	0.0515
	1.00	-3.150	0.0741 ± 0.0075	0.0780	0.0560 ± 0.0121	0.0552

TABLE 5.7: Gas branch densities ρ^* and pressures p^* at given temperatures T^* and chemical potentials μ^* calculated using GC-MC, GC-HMC, GC-GHMC and GC-GSHMC. The densities and pressures obtained from the Nicolas equation of state (EOS) are also reported.

One important observation is that, in general, the errors are much smaller than in the liquid case. Also, the agreement with the EOS data is better for all tested methods than in Table 5.5. While all methods produce accurate results, the GSHMC simulations reproduce the EOS data the most accurately.

As in the liquid case, we inspected the sampling performance of the tested methods. The relative time-normalized effective sample size (ESS) and the acceptance rates are shown in Table 5.8. Again, for simplicity, only a few of the states considered above in Table 5.7 are presented here.

method	conditions T^*	conditions μ^*	properties α (%)	properties ESS
GC-GSHMC	0.769	-4.127	57.17	12.05
	1.00	-3.150	57.42	16.15
GC-GHMC	0.769	-4.127	50.01	12.95
	1.00	-3.150	47.42	13.05
GC-HMC	0.769	-4.127	48.05	10.95
	1.00	-3.150	47.28	9.90
GC-MC	0.769	-4.127	41.45	1
	1.00	-3.150	42.16	1

TABLE 5.8: Gas branch acceptance rates (α) and effective sample sizes (ESS) observed in GC-MC, GC-HMC, GC-GHMC and GC-GSHMC simulations. ESS was normalized with respect to the data obtained with GC-MC at given T^* and chemical potentials μ^*.

As expected, higher acceptance rates than in the liquid case are found. The trend of the highest acceptance rates for GC-GSHMC is reproduced again. However, for the gas branch simulations, the bigger difference between the acceptance rates of GC-MC, GC-HMC and GC-GHMC is observed. As in the liquid case, even without a detailed tunning of the algorithms' parameters, the superiority over GC-MC of the newly developed GC-HMC, GC-GHMC and GC-GSHMC methods in the sampling efficiency has been demonstrated. The best performance was achieved with GC-GSHMC and GC-GHMC.

5.9 Conclusions and future work

The HMC, GHMC and GSHMC algorithms have been extended for the first time to the grand canonical ensemble. Their validity has been proved in simulations of Lennard-Jones fluids at different conditions. All three new methods reproduce well the predicted data (Nicolas et al., 1979; Johnson, Zollweg, and Gubbins, 1993). Also, the new algorithms sample up to 16 times better than the MC algorithm by Yao, Greenkorn, and Chao, 1982. Among three new methods, GC-GSHMC shows the best accuracy and sampling efficiency.

The proposed algorithms are only valid for homogeneous systems. Our future goal is to extend them to simple inhomogeneous systems and implement and test with rigid water models for the potential use in simulation of proteins in water.

Another future direction is to improve acceptance rates of the new methods. In very dense systems, the placement of a new particle can play a fundamental role in sampling efficiency of Monte Carlo based methods, since a completely random placement can lead to dramatic changes of the energy and thus rejections in the Metropolis tests. One possible way for improving the placement of inserted particles and increasing the acceptance rates is combining the current algorithms with the cavity-based methods (Mezei, 1980; Mezei, 1987; Deitrick, Scriven, and Davis, 1989). Another possible extension to the algorithms presented here would be to allow continuous changes in D (Cağin and Pettitt, 1991; Lo and Palmer, 1995; Boinepalli and Attard, 2003) and investigate the effect of such changes on the overall efficiency of simulations.

5.10 Published paper

1. **M. Fernández-Pendás**, B. Escribano, T. Radivojević, and E. Akhmatskaya (2014). "Constant pressure hybrid Monte Carlo simulations in GROMACS". In: *Journal of Molecular Modelling* 20.12, p. 2487. URL: http://dx.doi.org/10.1007/s00894-014-2487-y

Chapter 6

Enhancing Performance and Accuracy of MHMC for Simulation of Complex Systems: Numerical Integrators

6.1 Introduction

Replacing the standard Verlet integrator with a splitting integrator specified by a suitable value of a parameter may significantly improve, for a range of time steps, the conservation of the Hamiltonian and thus the acceptance rate of the proposals in the Hybrid Monte Carlo (HMC) method (see (McLachlan, 1995; Blanes, Casas, and Sanz-Serna, 2014) and Chapter 3 for details). Such integrators, however, possess shorter stability limits than the Verlet algorithm, as explained in Section 3.2.2.3. Also, the user is confronted with the problem of how best to choose the value of the parameter. The drawbacks of the use of splitting integrators, more sophisticated than Verlet, may be alleviated by resorting to the Adaptive Integration Approach (AIA), proposed in Section 3.3. For a user-chosen time step, this approach automatically identifies an optimal, system-specific integrator, by using information on the highest frequencies of the harmonic interactions present in the system; this information is typically extracted from the input data intended for a molecular dynamics package. The term "optimal" refers to the fact that the selected integrator minimizes, within a family of two-stage integrators, the expectation of the energy error for harmonic forces. When stability is an issue, AIA automatically chooses the Verlet integrator and, as the time step is reduced below the Verlet limit, AIA moves to more accurate integrators.

The family of modified Hamiltonian Monte Carlo (MHMC) methods introduced in Section 4.2 consists of HMC algorithms which, instead of sampling from the target canonical distribution (4.1), sample from an auxiliary importance density (4.2). Verlet/leapfrog has been the integrator of choice for MHMC methods and until recently such a decision has never been challenged. However, in (Radivojević, 2016; Radivojević and Akhmatskaya, 2017) it has been shown that replacing Verlet with optimized two-stage splitting integrators in MHMC may improve the observed sampling efficiency by a factor of up to 4 in high-dimensional statistical problems. In those references, though, there is no recipe for the rational choice of the integration scheme or/and the time step for a given system.

In this chapter, we present and analyze the novel multi-stage integrators, which were specifically derived for MHMC methods.

In Section 6.2 we introduce the modified multi-stage integrators with fixed parameters and compare their performance with that of the integrators proposed for HMC. In Section 6.3 we extend the ideas of AIA to derive a Modified Adaptive Integration Approach (MAIA) for

MHMC, in order to automatically select, for a given system and time step, the two-stage integrator with optimal conservation of the modified Hamiltonian, leading to the highest acceptance rates in MHMC. Extended MAIA (e-MAIA) offers the extra feature of controlling the stochasticity introduced in the momentum refreshment step in MHMC. Implementation of MAIA and e-MAIA in MultiHMC-GROMACS is explained in Section 6.4. Numerical results for proving the efficiency of MAIA and e-MAIA are provided in Section 6.5.

6.2 Modified multi-stage integrators

We focus on multi-stage integrators belonging to families (3.25) and (3.27). There are two reasons for an interest in these integrators in the context of MHMC. One is their potential to achieve, at a given computational cost, higher accuracy than Verlet. More accurate integrations imply higher acceptance rates in Hybrid Monte Carlo methods and thus better space exploration. A second possible benefit for the MHMC algorithms from the integrators of this class is that, due to the extra accuracy, they may avoid the need for computationally expensive, higher order modified Hamiltonians.

Our goal is to derive new multi-stage integrators to be used in the methods which sample with modified Hamiltonians, i.e., MHMC, and compare their impact on the performance of such methods with the efficiency of advanced integrators for HMC (McLachlan, 1995; Blanes, Casas, and Sanz-Serna, 2014) and the Verlet integrator.

In MHMC methods, the Hamiltonian dynamics equations are the same as in HMC methods. However, MHMC are based on different Metropolis tests where the acceptance rate depends on the capability of the integrator to conserve the value of a modified Hamiltonian. Indeed, the sampling performance of MHMC is controlled not by the energy error with respect to the true Hamiltonian as in HMC, but by the energy error with respect to the modified Hamiltonian. Thus, in order to enhance the performance of MHMC, the authors of (Radivojević and Akhmatskaya, 2017), inspired by the ideas of (McLachlan, 1995) and (Blanes, Casas, and Sanz-Serna, 2014) for improving HMC performance, designed the new integrators for MHMC by minimizing (expected) error in the modified Hamiltonians (4.3). To distinguish the new minimum error and minimum expected error integrators for sampling with modified (M) Hamiltonians from the corresponding ones designed for the HMC method, the authors of (Radivojević and Akhmatskaya, 2017) use the prefix M-; for instance, M-ME will denote minimum error integrator for sampling with modified Hamiltonians. We will follow these notations from now on.

In this section, we briefly review the major ideas behind the derivation of such integrators and present the resulting parameters. Then, we introduce yet another member of this group belonging to the three-stage family (3.27), which did not appear in (Radivojević and Akhmatskaya, 2017). We conclude the section with the comparison between modified multi-stage integrators and their counterparts developed for HMC. Performance of Verlet is also assessed.

The error metric for the derivation of the minimum error integrator proposed in (McLachlan, 1995) for sampling with a true Hamiltonian in the HMC method has been adapted to replace a true Hamiltonian with a modified Hamiltonian. This resulted in a modified minimum error integrator of two-stages, M-ME2 (see (Radivojević, 2016; Radivojević et al., 2018) and Table 6.1). Additionally, for problems with quadratic potential and kinetic function, the analysis of (Campos and Sanz-Serna, 2017) provides the condition for the highest stability

limit for three-stage integrators. In particular, the integrators that lie on the hyperbola

$$6ab - 2a - b + \frac{1}{2} = 0, \tag{6.1}$$

have considerably longer stability limit than others. As it has been explained in Section 3.2.2.2, the relation (6.1) reduces the three-stage integrators to a one-parameter family. The resulting parameter value for M-ME3 (modified minimum error of three-stages) has been obtained in (Radivojević et al., 2018) and it is presented in Table 6.1.

In order to derive integrators with an optimal expected modified energy error, the strategy similar to the one proposed in (Blanes, Casas, and Sanz-Serna, 2014) is adopted. The idea is to find such parameters of integrators that minimize the expected value of the modified energy error. Thus, in this case, the energy error resulting from numerical integration is in terms of the modified Hamiltonian and the expected value is taken with respect to the modified density (4.2).

Similar to the case of (Blanes, Casas, and Sanz-Serna, 2014), one may prove that the expected error in the modified Hamiltonian $\mathbb{E}[\Delta\tilde{H}^{[4]}]$ is positive. The objective then is to find a function $\rho(h, b)$ that upperbounds $\mathbb{E}[\Delta\tilde{H}^{[4]}]$, i.e.,

$$0 \leq \mathbb{E}[\Delta\tilde{H}^{[4]}] \leq \frac{1}{\beta}\rho(h, b), \tag{6.2}$$

where b is the parameter of a multi-stage integrator family. For the analysis, the one-dimensional harmonic oscillator is considered as in Section 3.2.2.3 (cf. (Akhmatskaya et al., 2017)). To find the error in the modified Hamiltonian after L integration steps with a time step h, one first finds the numerical solution for a single time step $(q(t+h), p(t+h)) = \psi_h(q(t), p(t))$. In matrix form, this is given by (3.30), with coefficients A_h, B_h, C_h depending on the integrator (cf. (3.31)). For the two-stage family of integrators, the resulting coefficients of S_h are (3.32) and for the three-stage integrators are (3.33). With the A_h, B_h, C_h coefficients, for a shadow Hamiltonian (4.10), one can define the function ρ in (6.2) as (cf. (Radivojević, 2016))

$$\rho(h, b) = \frac{(MB_h + C_h)^2}{2M(1 - A_h^2)}, \tag{6.3}$$

where

$$M = \frac{1 + 2h^2\mu}{1 + 2h^2\lambda}$$

depends on the parameters λ and μ of the shadow Hamiltonian, and the time step h. Note that the true Hamiltonian can be recovered by setting coefficients λ, μ to zero. Doing so, we obtain exactly (3.38), i.e., the same function as derived in (Blanes, Casas, and Sanz-Serna, 2014):

$$\rho_{\text{HMC}}(h, \xi) = \frac{(B_h + C_h)^2}{2(1 - A_h^2)}.$$

Finally, in the two-stage case, by substituting (3.32) into (6.3) one obtains the expression

$$\rho(h, b) = \frac{h^8\left(b\left(12 + 4b(6b - 5) + b(1 + 4b(3b - 2))h^2\right) - 2\right)^2}{4(2 - bh^2)(4 + (2b - 1)h^2)(2 + b(2b - 1)h^2)(12 + (6b - 1)h^2)(6 + (1 + 6(b - 1)b)h^2)}, \tag{6.4}$$

which bounds the expected error in the modified Hamiltonian. This function is then used within an optimization routine to find the value b that provides the optimal conservation

of the modified Hamiltonian for a specific system. The resulting integrator was named M-BCSS2, and the details of its derivation can be found in (Radivojević, 2016). The value of the parameter of M-BCSS2 is provided in Table 6.1.

Finally, one can construct the ρ function (6.3) for three-stage integrators. Using again the stability analysis from (Campos and Sanz-Serna, 2017), namely enforcing the condition (6.1), doing a minimization as in (Blanes, Casas, and Sanz-Serna, 2014; Radivojević, 2016) we obtain the parameter of the M-BCSS3 integrator for sampling with MHMC.

All integrators and coefficients that have been presented in Chapter 3[1] and this section are summarized in Table 6.1. The integrators derived in this study are highlighted.

integrator	**application**	**number of stages**	**coefficients**	$h_{\max}$
Verlet	HMC, MHMC	1	–	6.000
BCSS2	HMC	2	$b = 0.21178$	3.951
M-BCSS2	MHMC	2	$b = 0.238016$	4.144
ME	HMC	2	$b = 0.193183$	3.830
M-ME2	MHMC	2	$b = 0.230907$	4.089
BCSS3	HMC	3	$a = (1-2b)/4(1-3b)$ $b = 0.11888$	4.662
M-BCSS3	MHMC	3	$a = (1-2b)/4(1-3b)$ $b = 0.1441153$	4.902
M-ME3	MHMC	3	$a = (1-2b)/4(1-3b)$ $b = 0.142757$	4.887

TABLE 6.1: The splitting integrators for sampling with the true or 4th order modified Hamiltonians developed or tested in this study. Stability limit $h_{\max}$ is computed for problems with a quadratic potential and here presented in terms of the three-stage family.

In Figure 6.1, $\max_{0<h<\bar{h}} \rho_{\text{HMC}}(h,b)$ from (3.49) (dashed lines) and $\max_{0<h<\bar{h}} \rho(h,b)$ from (6.4) (solid lines) are plotted as functions of the maximal time step $\bar{h}$ (here normalized to the three-stage schemes, i.e., $\bar{h}_{r\text{-stage}} = r \cdot \bar{h}/3, r = 1,2,3$). While $\rho_{\text{HMC}}(h,b)$ is shown for two- and three-stage HMC integrators, $\rho(h,b)$ is depicted for two- and three-stage MHMC integrators. The corresponding functions for the Verlet integrator are also plotted. We note that the upper bound of the expected error in Hamiltonian, or modified Hamiltonian, and thus the error of the method, is lower for integrators derived for MHMC than in the case of the HMC specific integrators, which confirms the better conservation of modified Hamiltonians than true Hamiltonians by symplectic integrators. As follows from Figure 6.1, the multi-stage integrators derived for HMC and MHMC provide better accuracy than Verlet for time steps smaller or equal to a half stability limit of Verlet, i.e., $\bar{h} = 3$, with three-stage integrators being superior to the two-stage class[2]. The integrators derived for MHMC guarantee a better accuracy than other integrators for $\bar{h}$ even bigger than 3, which implies their efficiency for

[1] The parameters for BCSS2 and BCSS3, and ME have been taken from the original papers (Blanes, Casas, and Sanz-Serna, 2014) and (McLachlan, 1995), respectively.

[2] One should notice that $\bar{h}$ in Figure 6.1 refers to a time step for a three-stage integrator. If Verlet is viewed as a single stage integrator, its half stability limit corresponds to $\bar{h} = 1$.

bigger time steps compared with Verlet and multi-stage integrators for HMC. A logarithmic scale version of the left-hand graph, shown in the right-hand graph, gives a better insight into the behavior of the functions.

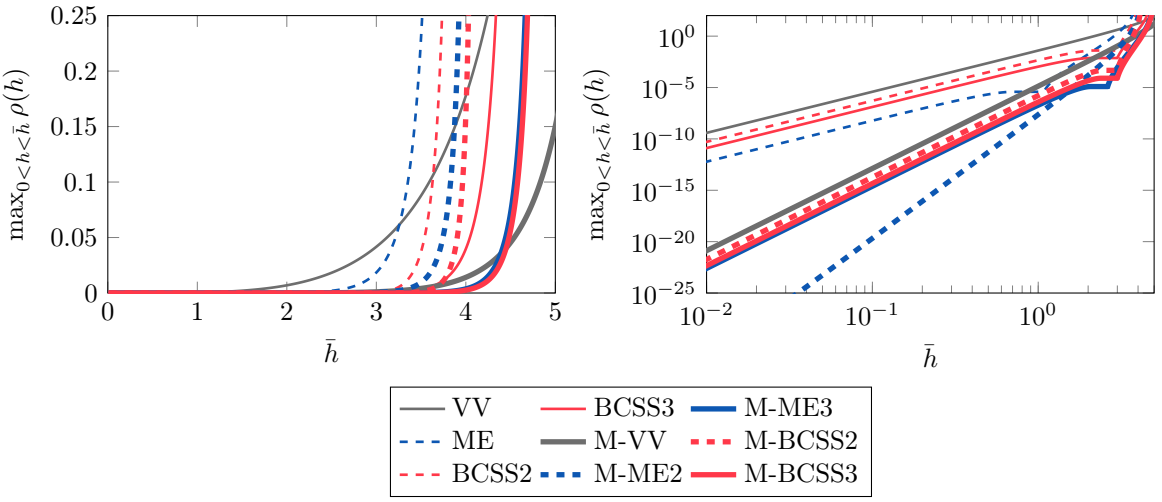


FIGURE 6.1: Upper bound for the expected energy error for the two- and three-stage (M-)BCSS, (M-)ME and Verlet integrators for sampling with the true Hamiltonian (dashed) and 4th order modified Hamiltonian (solid). Right-hand graph shows the same functions on a logarithmic scale.

It is important to note that the Verlet integrator has the largest stability interval among other splitting integrators, and due to this, care should be taken of the choice of the time step when using multi-stage integrators. The stability intervals $(0, h_{\max})$ computed for each of the examined integrators are given in Table 6.1 in terms of the three-stage family. We note that the trends of the stability limit $h_{\max}$ for each integrator are in agreement with the corresponding upper bound functions. Nevertheless, as Figure 6.1 suggests, the accuracy is degrading with $\bar{h}$ approaching the stability limit. It is the characteristics of the simulation problem (such as the dimension of the system, number of observations, nature of the physical system) that determine the optimal time step and therefore the integrator which would provide the best performance.

The implementation of the modified integrators presented in this section is explained in Section 7.4. However, since the different integrators are specified through a parameter b, their implementation does not differ from that used for the integration schemes introduced in Section 3.2.2.

We investigated the performance of the multi-stage integrators discussed in this chapter using the GSHMC method described in Section 4.3 and belonging to the MHMC class.

As a benchmark, we chose the toxin system introduced in Section 3.5.2. The simulations were performed for a range of time steps: 10, 15, 20, 22.5 and 25 fs (in one-stage dimensions). Different lengths of MD trajectories L were also tested, but for the sake of clarity, in all tests presented here the length of MD trajectories was fixed to 4000 steps for Verlet and scaled correspondigly for two- and three-stage integrators. The angle φ used for the momentum refreshment was set to 0.2 and the modified Hamiltonian (4.10) was used for all tests.

We start by measuring the acceptance rates in the GSHMC simulations with different multi-stage integration schemes. A fundamental feature of the GSHMC method is that it maintains very high acceptance rates. It is confirmed in Figure 6.2 (left), where the effect of

various multi-stage integrators and the standard Verlet on the acceptance rates in GSHMC simulations is presented. For small time steps, all integrators guarantee high acceptance rates, but the situation changes as the time step increases and the shorter stability intervals of the different multi-stage methods (cf. Table 6.1) result in acceptance rates well below those achieved with Verlet. We observe that the integrators derived specifically for sampling with modified Hamiltonians (solid lines) in general show better acceptance rates than their non-modified counterparts (dashed lines). Moreover, the M-BCSS3 integrator provides the best conservation of the modified Hamiltonians and thus the highest acceptance rates. The exception occurs for the biggest time steps tested, where Verlet leads to the highest acceptance rates due to its better stability. All the trends presented in Figure 6.2 (left) are in a good agreement with the theoretical predictions shown in Figure 6.1.

The averages of simulated temperatures T calculated in GSHMC simulations were used for evaluating the accuracy provided by the tested integration schemes. Figure 6.2 (right) confirms that all the methods are capable of producing the desired averaged temperature. The only exceptions are the two-stage methods derived for HMC, which, for the biggest time steps, obtained unrealistically high temperatures as a result of the very low acceptance rates observed during the simulations in these cases (Figure 6.2 (left)).

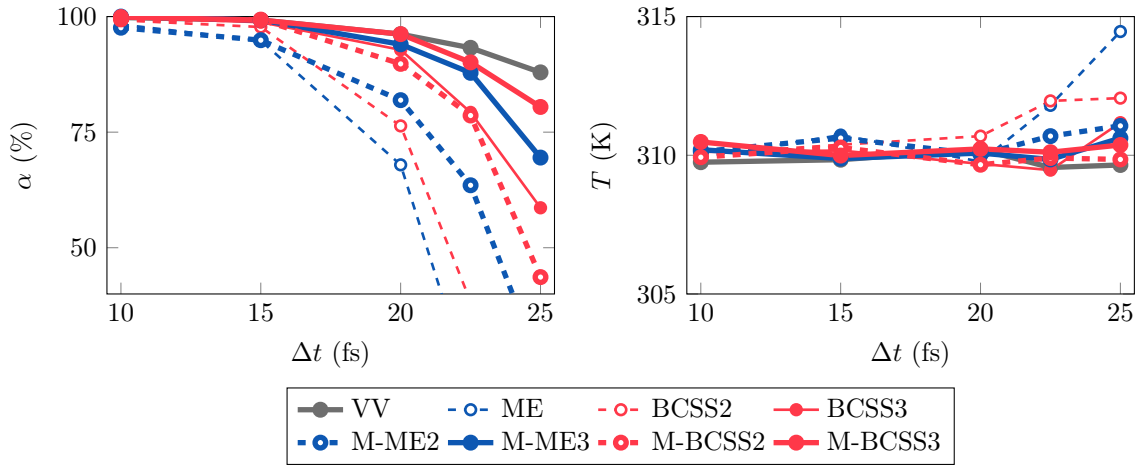


FIGURE 6.2: Toxin. Acceptance rates (left) and average temperatures (right) as functions of the time step Δt. Comparison of the two-stage (M-)BCSS2, (M-)ME(2), three-stage (M-)BCSS3, (M-)ME(3), and Verlet integrators.

We shall see next how the integrators impact the sampling efficiency of GSHMC, measured in terms of ESS of the toxin drift to the preferred interfacial location over the equilibration and production periods. We notice that no time normalization for ESS is required as the simulation parameters (Δt, L, overall length) are chosen in the way to maintain the same computational cost for all tests. Figure 6.3 presents the relative ESS (i.e., ESS normalized with respect to the values obtained with Verlet) calculated from GSHMC simulations using different integrators and time steps. In the left-hand graph, ESS is calculated for equilibration period, during which the toxin is moving towards its desired position in the bilayer membrane. Clearly, M-BCSS3 provided the highest values of ESS and thus the best sampling for all choices of time steps but the last one, for which the acceptance rates decay due to a lower stability limit than for Verlet. In the right-hand graph, ESS is calculated for the production phase of the simulations, i.e., once the toxin has reached its equilibrium position. As it is the case for

equilibration, M-BCSS3 provided the highest values of ESS and thus the best sampling for all choices of time steps but the last one. Importantly, the highest absolute ESS over the range of studied time steps has also been observed with the M-BCSS3 integrator.

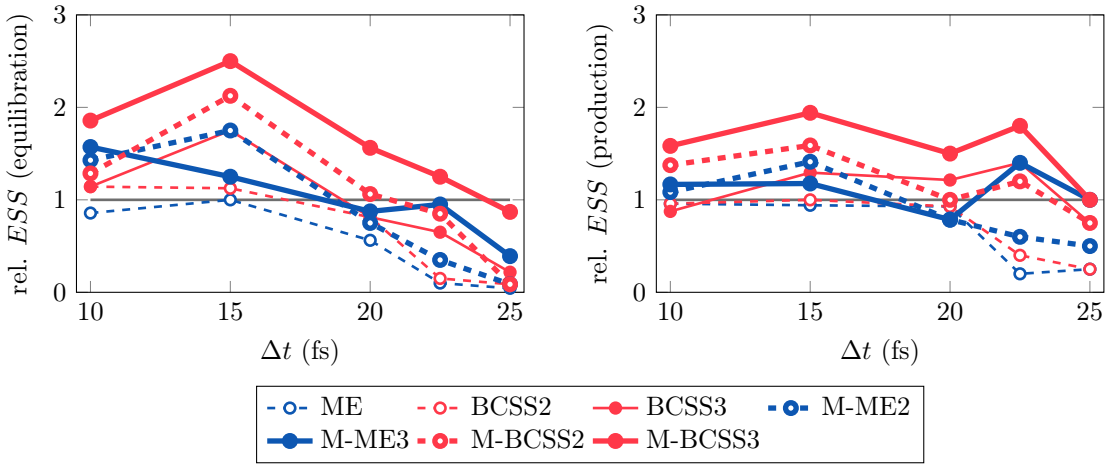


FIGURE 6.3: Toxin. Relative ESS (with respect to Verlet) for the equilibration (left) and production (right) phases of the simulations. Comparison of the two-stage (M-)BCSS2, (M-)ME(2), three-stage (M-)BCSS3, M-ME3, and Verlet integrators.

In summary, we demonstrate that the multi-stage integrators specifically developed for MHMC methods can outperform in accuracy and sampling efficiency the traditionally used integrators, including Verlet, for a range of time steps. However, their performance drops with increasing a time step and can become poorer than that achieved with Verlet. Thus, given a time step, the question "which integrator to choose?" remains unclear. In the next section, we will propose the solution to this problem by introducing the adaptive integration approach.

6.3 Adaptive algorithms

We present two novel two-stage adaptive algorithms: the Modified Adaptive Integration Approach (MAIA) and the extended MAIA (e-MAIA).

6.3.1 MAIA

MAIA is an algorithm which adapts the parameter b in the two-stage integrators (3.43) to the problem being solved and the value of Δt chosen by the user so as to maximize the expected acceptance rate α of the proposal $(\mathbf{q}', \mathbf{p}')$ in (4.8) or, equivalently, to minimize the expectation of the modified energy error

$$\Delta\tilde{H}^{[4]}(\mathbf{q}, \mathbf{p}^*) = \tilde{H}^{[4]}(\mathbf{q}', \mathbf{p}') - \tilde{H}^{[4]}(\mathbf{q}, \mathbf{p}^*),$$

with respect to the modified density (4.2).

The analysis is based on a study of the one-dimensional harmonic oscillator (cf. Sections 3.2.2.3 and 6.2) for two-stage integrators (cf. Akhmatskaya et al., 2017). For a method

of the family (3.43), the modified Hamiltonian in (4.10) takes the form

$$\tilde{H}^{[4]}(q,p) = \frac{1}{2}\frac{p^2}{M} + \frac{1}{2}kq^2 + \Delta t^2\lambda\frac{k}{M^2}p^2 + \Delta t^2\mu\frac{k^2}{M}q^2. \tag{6.5}$$

If $\omega = \sqrt{k/M}$ is the angular frequency of the harmonic oscillator and h denotes the nondimensional time step defined as $h = \omega\Delta t$, then for the expected $\Delta\tilde{H}^{[4]}$ it holds (6.2), with the function ρ presented in (6.4). Note that the expectation $\mathbb{E}[\Delta\tilde{H}^{[4]}]$ is taken with respect to the probability $\tilde{\pi}$ (4.2) sampled by the algorithm.

For a model consisting of D, possibly coupled, harmonic oscillators with angular frequencies ω_i, $i = 1, \ldots, D$, the bound becomes

$$\mathbb{E}[\Delta\tilde{H}^{[4]}] \leq \frac{1}{\beta}\sum_{i=1}^{D}\rho(h_i, b),$$

with $h_i = \omega_i\Delta t$. Minimization of the right-hand side will, therefore, ensure optimal conservation of the modified Hamiltonian in the harmonic model.

In MAIA, given a physical problem which includes nonharmonic forces and a value of Δt, we estimate the fastest of the angular frequencies, $\tilde{\omega}$, of the two-body interactions and compute the nondimensional quantity

$$\tilde{h} = \sqrt{3}\tilde{\omega}\Delta t, \tag{6.6}$$

with $\sqrt{3}$ being a safety factor to be discussed presently. We then find the value of b that minimizes

$$\max_{0<h<\tilde{h}} \rho(h, b). \tag{6.7}$$

Note that $(0, \tilde{h})$ is the shortest interval that contains all the values $h_i = \sqrt{3}\,\omega_i\Delta t$, where ω_i are the frequencies in the problem. In contrast to AIA, where the factor of $\sqrt{2}$ had to be used to avoid resonances of up to fourth order, in MAIA, the factor $\sqrt{3}$, covering resonances of up to fifth order, was found to be appropriate (see Table 3.1).

The MAIA algorithm can be summarized as follows:

Given a physical system and a value of Δt, the MAIA algorithm determines the value of the parameter b to be used in (3.43) in the following way:

1. Find the periods or frequencies of all two-body interactions in the system. Determine the minimum period $\tilde{T} = 2\pi/\tilde{\omega}$, with the fastest frequency $\tilde{\omega}$, and compute the nondimensional quantity $\tilde{h}$ in (6.6).

2. Check whether $\tilde{h} < 2\sqrt{2}$, which is the usual stability limit in molecular simulation for Verlet integrators (see for instance (Mazur, 1997)). If not, there is no value of b for which the scheme (3.43) is stable for the attempted time step Δt and the integration is aborted.

3. Find the optimal value of the parameter b by minimizing (6.7) with the help of an optimization routine.

When Δt is "large" for the problem at hand, in the sense that stability is the primary concern, MAIA will choose $b = 1/4$, i.e., the Verlet integrator. Smaller values of Δt allow MAIA to reduce b and increase accuracy in the conservation of the modified Hamiltonian (see

Figure 6.4). Figure 6.4 also shows the advantage of MAIA when compared with the older algorithm AIA, developed for the HMC method, which does not use modified Hamiltonians and samples with respect to the target canonical density. This is also in agreement with the expectations in (4.4) and in (4.5).

The right panel of Figure 6.4 also confirms the two different expectations of the Hamiltonian error and the modified Hamiltonian error in equations (4.4)-(4.5).

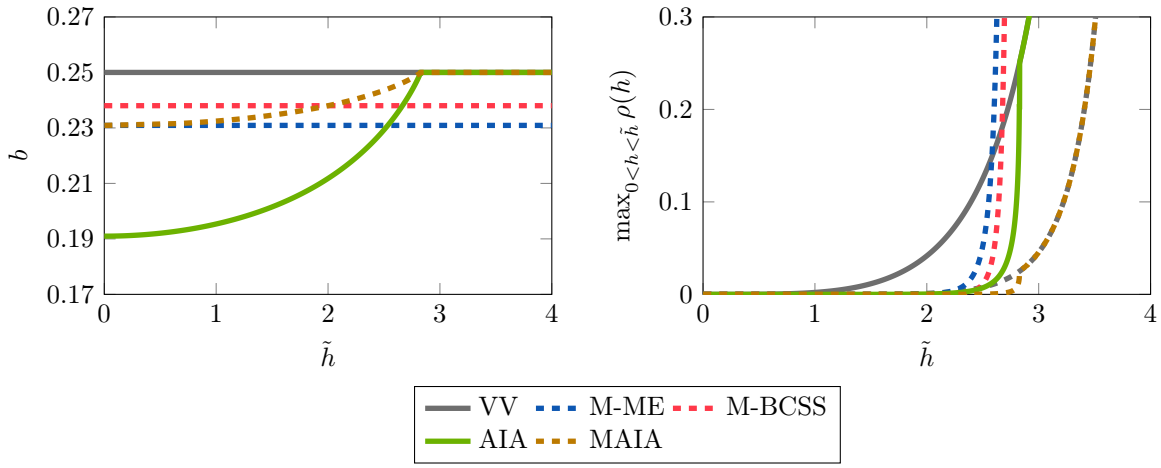


FIGURE 6.4: Parameter b for different integrators as a function of $\tilde{h}$ (left) and bounds of the expected energy error measured with respect to the true, in solid lines, or modified Hamiltonian, in dashed lines (right). There are two lines for VV, as it may be used to sample from the true (HMC) or the importance density (GSHMC). AIA operates with respect to the true energy and MAIA with respect to its modified counterpart. The algorithms that operate with modified Hamiltonians possess smaller expected errors. This explains why, in general, VV GSHMC has higher acceptance rates than VV HMC and MAIA improves on AIA. Since in this section only two-stage integrators are discussed, from now on we drop the index 2 introduced in Section 6.2 for two-stage integrators, i.e., M-ME2, M-BCSS2.

Figure 6.5 justifies why AIA and MAIA can be useful: the integrators with fixed parameter minimum error, BCSS, ME and VV (and their modified counterparts) have the smallest expected (modified) energy error for different choices of a time step. Thus, the methodology that automatically tunes the integrator parameter is useful.

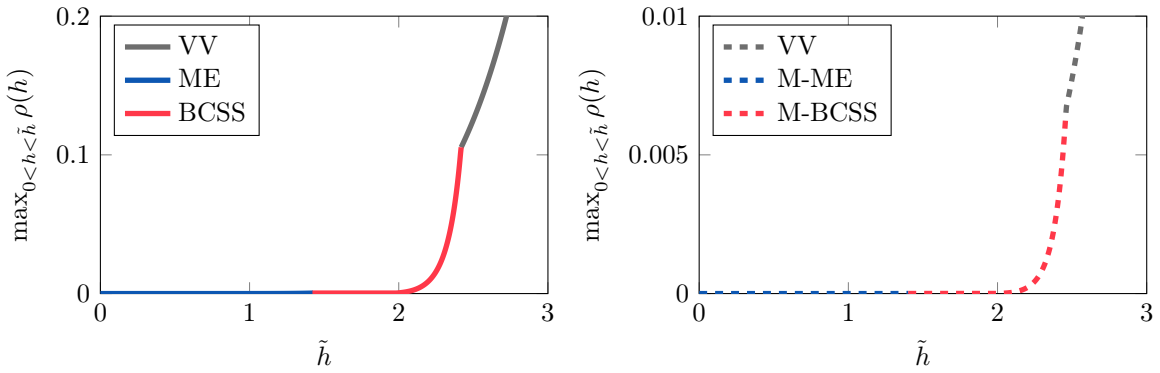


FIGURE 6.5: Minimum bounds of the expected energy error among the three two-stage integrators VV, ME and BCSS (left) and expected modified energy error among the three two-stage integrators VV, M-ME and M-BCSS (right). The time steps studied are all smaller than 3 since the loss of stability for the biggest time steps is not interesting in this comparison.

6.3.2 e-MAIA

The overall performance of an MHMC method depends not only on the acceptance rate α of the proposal made after each MD integration leg (see (4.8)) but also on the momentum update acceptance rate α_p in (4.12). The latter may play an important role in the quality of sampling (previous investigations on this issue can be found in (Akhmatskaya and Reich, 2008; Wee et al., 2008)) since α_p determines the frequency of the momenta resamplings. So far, we have looked for the integrator that maximizes α and our next objective is to find a way to control α_p.

As we did above, we build the analysis on the use of a harmonic oscillator model. For the scalar harmonic potential, the stationary marginal p.d.f.'s of the (stochastically independent variables) p and u (see (4.11)) are

$$\pi(p) \propto \exp\left(-\beta\left(\frac{1}{2}\frac{p^2}{M} + \Delta t^2 \lambda \frac{k}{M^2} p^2\right)\right), \qquad \pi(u) \propto \exp\left(-\frac{\beta}{2}\frac{u^2}{M}\right), \tag{6.8}$$

respectively, and the extended Hamiltonian in (4.13) reads

$$\hat{H}(q,p,u) = \tilde{H}^{[4]}(q,p) + \frac{1}{2}\frac{u^2}{M},$$

with $\tilde{H}^{[4]}$ given in (6.5). As it was shown in (Radivojević, 2016), the difference in extended Hamiltonian satisfies

$$\begin{aligned}\Delta\hat{H} &= \hat{H}(q,p^{\text{trial}},u^{\text{trial}}) - \hat{H}(q,p,u) \\ &= \Delta t^2 \lambda \left(\sin^2\varphi\left(\frac{k}{M^2}u^2 - \frac{k}{M^2}p^2\right) + 2\cos\varphi\sin\varphi u\frac{k}{M^2}p\right),\end{aligned}$$

and from here it is found that

$$\mathbb{E}[\beta\Delta\hat{H}] = \Delta t^2 \beta\lambda \sin^2\varphi \frac{\omega^2}{M}\left(\mathbb{E}[u^2] - \mathbb{E}[p^2]\right).$$

From (6.8) we have

$$\mathbb{E}[p^2] = \beta^{-1} M \left(1 + 2\Delta t^2 \lambda \omega^2\right)^{-1}, \quad \mathbb{E}[u^2] = \beta^{-1} M,$$

and then

$$\begin{aligned}\mathbb{E}[\beta \Delta \hat{H}] &= \Delta t^2 \lambda \sin^2 \varphi \omega^2 \left(1 - \left(1 + 2\Delta t^2 \lambda \omega^2\right)^{-1}\right) \\ &= \frac{2\Delta t^4 \lambda^2 \sin^2 \varphi \, \omega^4}{1 + 2\Delta t^2 \lambda \omega^2}.\end{aligned}$$

In terms of the dimensionless time step $h = \omega \Delta t$, one obtains

$$\mathbb{E}[\beta \Delta \hat{H}] = \frac{2h^4 \lambda^2 \sin^2 \varphi}{1 + 2h^2 \lambda}. \tag{6.9}$$

For the model consisting of D harmonic oscillators with angular frequencies ω_i, $i = 1, \ldots, D$, the equivalent of (6.9) is

$$\mathbb{E}[\beta \Delta \hat{H}] = \sum_{i=1}^{D} \frac{2h_i^4 \lambda^2 \sin^2 \varphi}{1 + 2h_i^2 \lambda} \geq D \frac{2\bar{h}^4 \lambda^2 \sin^2 \varphi}{1 + 2\bar{h}^2 \lambda}, \tag{6.10}$$

where $h_i = \omega_i \Delta t$ are the dimensionless time steps and $\bar{h} = \bar{\omega} \Delta t$ with $\bar{\omega}$ equal to the slowest angular frequency among all the oscillators.

Using $\alpha_p \leq \exp\left(-\beta \Delta \hat{H}\right)$ (see (4.12)), from the inequality (6.10) and for a concrete choice of the angle φ_p, we can find the approximation

$$-\frac{\log \mathbb{E}[\alpha_p]}{D} \approx \frac{2\bar{h}^4 \lambda^2 \sin^2 \varphi_p}{1 + 2\bar{h}^2 \lambda}. \tag{6.11}$$

It has to be remarked that the fastest oscillation frequency features in the analyses of MAIA and its predecessor AIA, but the *slowest* frequency is used in (6.11).

From (6.11), the expected acceptance rate in the momentum update may be controlled by three parameters: the parameter $\lambda = \lambda(b)$ that depends on the specific integrator being used, the parameter $\bar{h}$, which for a given problem is a function of Δt, and the angle φ. This fact motivates the algorithm that we call extended MAIA or e-MAIA. For a user-chosen Δt, e-MAIA first finds an integrator within the family of two-stage schemes that maintains the smallest expected modified energy error in the molecular dynamics part of the MHMC algorithm and then adjusts the value of φ to achieve a desired acceptance rate for the momentum update step. As explained above, the acceptance rates in the momentum update step depend on the choice of angle φ, whereas the MAIA analysis does not depend on φ. This means that, for some fixed values of φ and Δt, the integrator nominated by MAIA may not be favorable for maintaining an appropriate acceptance rate in the momenta. The goal of e-MAIA is to provide an adaptive choice of the angle φ to achieve a target, user-specified acceptance rate in the momentum update step while keeping the highest acceptance rate for positions.

While a high acceptance rate in the MD part has a positive effect on sampling with modified Hamiltonians, a too-frequent acceptance of momentum (close to 100 %) could lead to two undesired scenarios: (i) an accuracy deteriorating thermalization of the simulation, if the high acceptance rate is caused by a value of the angle φ very close to zero (cf. (Wee et al., 2008; Akhmatskaya, Bou-Rabee, and Reich, 2009)); or (ii) a disruption of the dynamical trajectories if the momenta are always resampled while φ is significantly bigger than zero

(cf. (Akhmatskaya and Reich, 2008)). In the first scenario, the simulation will mimic an MD behavior in the NVE ensemble. The rationale for introducing e-MAIA is the possibility of simultaneously adapting the parameters b and φ to control both the acceptance probabilities α and α_p of the MD integration legs and the momentum updates.

The algorithm e-MAIA is as follows:

1. For a given physical problem, choose a time step Δt for the integration of the equations of motion, a target acceptance rate AR_p for the momentum update, and an initial value φ_0 of the angle φ.

2. Find the slowest and the fastest angular frequencies in the harmonic interactions, $\bar{\omega}$ and $\tilde{\omega}$, respectively.

3. The integrator parameter b^* is obtained as in MAIA by optimization of the function ρ in (6.4). This choice of b^* guarantees the highest possible acceptance rate for harmonic interactions in the MD step.

4. The function which bounds the expected extended Hamiltonian error is given by (see (6.11))
$$\tau(\bar{h}, b^*, \varphi) = \frac{2\bar{h}^4 \lambda^{*2} \sin^2 \varphi}{1 + 2\bar{h}^2 \lambda^*},$$
where λ^* is the value of λ when $b = b^*$ and
$$\bar{h} = \bar{\omega} \Delta t. \tag{6.12}$$
The angle φ^* is chosen as
$$\varphi^* = \arg \min_{\varphi \in (0, \pi/2]} \theta(\varphi), \tag{6.13}$$
with
$$\theta(\varphi) = \left| -\frac{\log (AR_p)}{D} - \tau(\bar{h}, b^*, \varphi) \right|.$$

5. If the selected φ^* is smaller than φ_0, then either decrease the target AR_p and go to step 4 or, alternatively, define the function
$$\sigma(h, b, \varphi_0) = \rho(h, b) + \tau(h, b, \varphi_0) \tag{6.14}$$
and choose b^{**} that minimizes $\max_{0<h<\tilde{h}} \sigma(h, b, \varphi_0)$. The fastest oscillation is used again for the momentum update part, since in this case we are constructing an upper bound of the expected energy error.

We stress that for very small values of φ, an MHMC method loses its extra sampling abilities and behaves similarly to standard molecular dynamics. In e-MAIA, this possibility is eliminated in step 5 of the algorithm in two ways. One option is to keep decreasing the target AR_p until φ^* rises above φ_0. Another option is to optimize the joint bound function constructed for both expected errors, $\mathbb{E}[\beta \Delta \tilde{H}]$ and $\mathbb{E}[\beta \Delta \hat{H}]$. Though this sacrifices the position acceptance rates, the expected loss is small provided that $\varphi_0 \ll \pi/2$.

The reader should notice that, whereas MAIA in principle works for any method that samples with respect to modified Hamiltonians, e-MAIA only works for those MHMC methods which perform the momentum update step in the way described in (4.11).

6.4 Implementation

Similarly to AIA, MAIA and e-MAIA have been implemented in the GROMACS preprocessing module `grompp`. The preprocessing module is run only once before any simulation and, thus, it does not introduce computational overheads in the simulation.

In addition to the `grompp` standard functionalities, the more advanced analysis of the harmonic interactions is included in this module in MultiHMC-GROMACS. As has been explained in Section 3.3, the fastest harmonic interaction predetermines a maximal time step allowed for the stable numerical integration of the equations of motion. On the other hand, the slowest harmonic interactions are used in the e-MAIA algorithm to identify the best choice of the parameter φ. In MultiHMC-GROMACS, `grompp` searches for the periods corresponding to the fastest and slowest oscillations, $\tilde{T}$ and $\bar{T}$, respectively. The value $\tilde{T}$ is used to define the upper limit of the dimensionless time step, $\tilde{h} = \sqrt{3}(2\pi/\tilde{T})\Delta t$, following MAIA algorithm. The optimal value of the parameter b for a MAIA or e-MAIA integrator is then found as the argument that minimizes the maximum of ρ (6.4) for the range of dimensionless time steps from zero to $\tilde{h}$. As in Section 3.3, the minimization is performed with a particle swarm optimization algorithm driven by a golden section search. The value $\bar{T}$ is used to determine the angle φ, as explained in the e-MAIA algorithm.

Both b and φ are stored in the *input record* structure introduced by GROMACS for keeping all the input data during the whole simulation. Thus, b and φ can be accessed from every routine in the package.

The integrators resulting from the Modified Adaptive Integration Approach described above belong to the family (3.43) and thus are naturally included in the list of integrators implemented (see Section 7.4 for details). The parameter used in the *.mdp* file is `maia`. In case Extended MAIA is used, the parameter `maia` is selected, but two more parameters have to be added to the *.mdp* file: a boolean variable that decides if e-MAIA is used or not and the target acceptance rate AR_p. The angle of the GSHMC method will be used as the initial value φ_0 (more details on the GSHMC parameters can be found in Section 7.3). The specific parameters in the *.mdp* file are summarized below:

```
extended_maia           = yes;      yes / no
target_ar               = 0.9;      any positive rational
parameter_phi           = 0.2;      0<phi<pi/2
```

Obviously, to run e-MAIA, both MAIA and GSHMC methods have to be selected.

The flowchart in Figure 6.6 summarizes MAIA and e-MAIA algorithms.

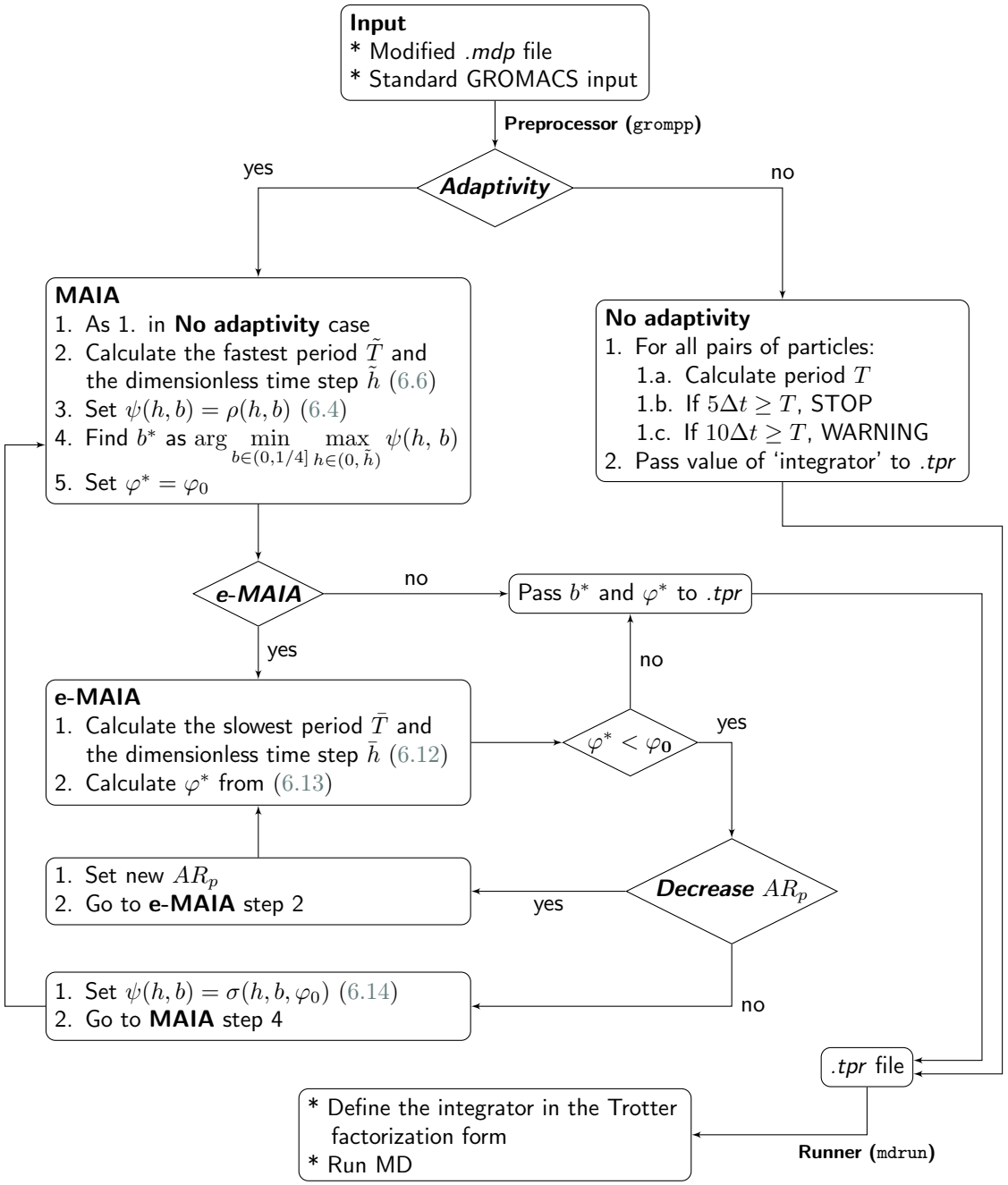


FIGURE 6.6: Flowchart of the Modified Adaptive Integration Approach (MAIA) and the extended MAIA (e-MAIA) as implemented in MultiHMC-GROMACS.

6.5 Numerical experiments

In order to evaluate the efficiency of the proposed (e-)MAIA algorithm, we first compared its performance with that of several integration schemes which potentially can compete with it. Then we estimated the performance of GSHMC combined with (e-)MAIA in comparison with other popular sampling methods. More precisely:

- e-MAIA was compared with fixed parameters integrators specifically derived for MHMC methods. The counterpart of BCSS for modified Hamiltonians (M-BCSS) and the equivalent to the scheme of McLachlan that minimizes the errors of modified Hamiltonians (M-ME) were included in the comparison. Both M-BCSS and M-ME have been recently derived (cf. (Radivojević, 2016) and Section 6.2) and implemented in MultiHMC-GROMACS. All three integrators were combined with the GSHMC method. Also, e-MAIA was compared with integrators successfully used for molecular simulation in MD, HMC and GSHMC. The velocity Verlet and AIA combined with GSHMC were selected in this case.

- e-MAIA was compared with MAIA when both were implemented within GSHMC.

- GSHMC was compared with HMC and MD. For each tested sampling method, the most efficient integrator was used: e-MAIA was chosen for GSHMC and AIA was employed in MD and HMC.

To provide a fair comparison, the following issues have been taken into account while producing the numerical results. To equalize the time spent on force calculations using Verlet and two-stage integrators, Verlet was always run with half a time step and twice the number of steps. Also, in the simulations with HMC and GSHMC, the number of Metropolis tests was kept constant regardless of the acceptance/rejection output. The computational overhead due to the evaluation of modified Hamiltonians in GSHMC was taken into account by normalizing calculated integrated autocorrelation functions with respect to computational times. We notice that this overhead is, on average, of 1-2 % with respect to MD with the v-rescale thermostat or with respect to HMC, since both MD and HMC have the same computational cost. We also notice that the overheads of GSHMC tend to decrease when the trajectory lengths increase.

The tests were performed using two benchmark systems previously introduced for testing AIA (see Section 3.5.2). Both benchmarks, toxin and villin, were run over a range of time steps Δt. The aim was to monitor the evolution of the parameters b and φ (4.11) automatically chosen for each Δt in (e-)MAIA, and estimate their effect on the overall sampling performance of GSHMC. In all plots in this section, values of time steps correspond to two-stage integrators and assume twice smaller time steps for velocity Verlet.

Different lengths of MD trajectories L in GSHMC simulations were also tested. This parameter may play an important role in the sampling efficiency of GSHMC simulations when the chosen values are either too small or too large, as it has been observed in (Wee et al., 2008). However, for the sake of clarity, in all tests presented in this work, the length of MD trajectories was fixed to 2000 steps when two-stage integrators were used and to 4000 otherwise. These values were found to be good choices for both GSHMC and HMC with different integration schemes and this is also confirmed by findings in Section 3.6. Also, as discussed above, for this trajectory length L the computational overheads of GSHMC with respect to MD are smaller than 1%.

With the obvious exception of e-MAIA, the angle used for the momentum refreshment (4.11) was set to 0.2 for all tests unless stated otherwise.

Each test has been repeated 10 times and every result reported here was obtained by averaging over the multiple runs to reduce statistical errors.

The numerical experiments were performed using the two benchmark systems from Section 3.5.2: toxin and villin. The same system setups were also considered.

6.5.1 Toxin

We start by measuring the acceptance rates of positions and momenta in the GSHMC simulations with different integration schemes. For the sake of clarity, we excluded from the plots the results for the MAIA algorithm, leaving only the data for e-MAIA. This makes sense because the position acceptance rates for MAIA and e-MAIA are always very similar (see step 3 of the e-MAIA algorithm), while e-MAIA has a clear advantage over MAIA as far as the acceptance rates for momenta are concerned. We shall provide more details on this issue later.

The primary objective of the MAIA algorithm is to maximize the acceptance of position proposals in an MHMC method by minimizing the expected errors in modified Hamiltonians. Then, the first natural test for MAIA is to check whether the position acceptance rates observed in GSHMC simulations combined with MAIA are not below those observed with other two-stage integrators. In Figure 6.7, the effect of various integrators such as e-MAIA, the modified versions of BCSS (M-BCSS) and ME (M-ME), the standard VV, and AIA on the acceptance rates in GSHMC simulations is investigated. The trends presented in the plot in the left are in good agreement with the theoretical prediction in Figure 6.4 (right panel). Indeed, the acceptance rates obtained with the modified adaptive approach e-MAIA, over the range of time steps considered, are never lower than the ones provided by the other integrators tested. For small time steps, all integrators, except AIA, guarantee high acceptance rates, but the situation changes as the time step increases and the shorter stability intervals of M-BCSS and M-ME result in acceptance rates well below those achieved with e-MAIA and VV. The low acceptance rates for AIA are not surprising since this method was developed for sampling with respect to the true Hamiltonian and provides the lowest expected errors in Hamiltonian rather than in modified Hamiltonian. However, for the largest time step of 50 fs, the parameter b in AIA becomes equal to $1/4$ and thus AIA is equivalent to VV (see Figure 6.4, left). The same applies to MAIA/e-MAIA for the longest time step, as can also be seen in Figure 6.4 (left). It merely reflects the fact that the velocity Verlet integrator possesses the longest stability interval among the two-stage integrators and the adaptive methods AIA and MAIA select velocity Verlet when the time step goes beyond the stability limit of other two-stage integrators.

The acceptance rates for momenta are shown in the right panel of Figure 6.7. For e-MAIA, we fixed the target acceptance AR_p to 90 % bearing in mind that too high (near 100 %) acceptance rates may degrade accuracy, whereas low acceptance rates usually reduce the sampling efficiency of GSHMC. With this target set, e-MAIA chose an appropriate value of φ for each time step being tested. The simulations with other integrators were run with the fixed value $\varphi = 0.2$, which was selected to achieve good performance for the longest time steps. Naturally, with every integrator, the parameter φ can be adapted, by trial and error, to each simulation and time step, but we have to stress that, in practice, blindly tuning the value of φ is somewhat time-consuming and not necessarily results in the optimal choice of φ. That is why the ability of e-MAIA to automatically optimize such a choice is very welcome. As follows from Figure 6.7 (right), for all tested time steps, e-MAIA maintained well the target AR_p by varying φ. The other integrators being combined with GSHMC led to very high, unwelcome acceptance rates for most time steps tested.

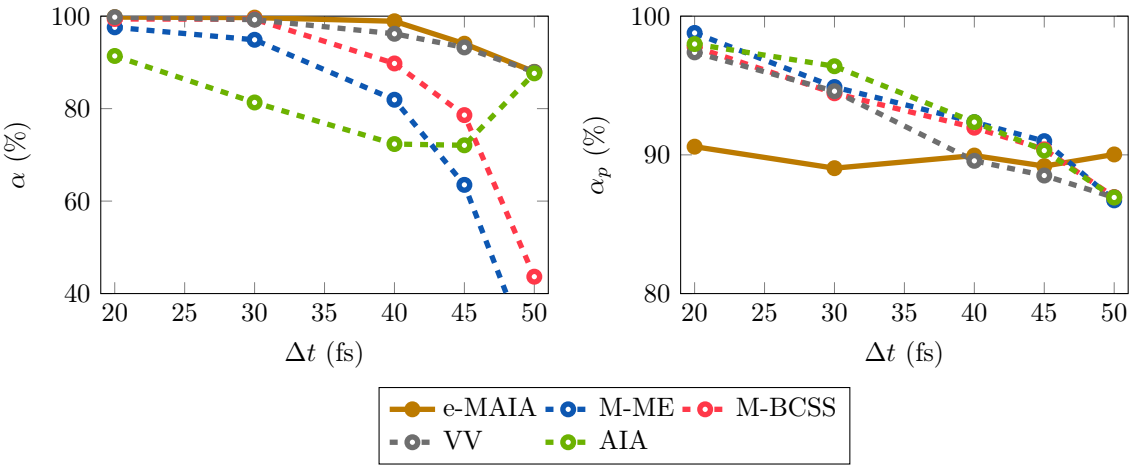


FIGURE 6.7: Toxin. Acceptance rates for positions (left) and momenta (right) observed in GSHMC simulations when using M-BCSS, M-ME, VV, AIA (all dashed lines), and e-MAIA (solid line). e-MAIA maintains the target AR_p of 90 % for each value of Δt (right).

We shall see next how the trends observed above for the acceptance rates impact the sampling efficiency of GSHMC. In the case of toxin, this efficiency was measured in terms of the integrated autocorrelation function IACF of the toxin drift d to the preferred interfacial location over the "convergence period." The IACF is defined as

$$\text{IACF}^{\Omega} = \sum_{l=0}^{K'} \text{ACF}(\tau_l), \tag{6.15}$$

where $\text{ACF}(\tau_l)$, $l = 0, \ldots, K' < K$ is the standard autocorrelation function for the time series Ω_k of K samples, $k = 1, \ldots, K$ (see (Kennedy and Pendleton, 2001; Allen and Tildesley, 1989) for details). For GSHMC, the ACF's are calculated taking into account the weights collected during simulations as suggested in (Radivojević, 2016). We notice that in all simulations performed the normalized weights are close to 1 due to small differences between modified and true Hamiltonians observed in the simulations as well as the choice of temperatures (common for molecular simulations of biological systems) leading to $\beta < 1$. This means that the metrics designed for weighted and nonweighted methods would not generate data that are too different. This, however, is not expected in a general case and is not common in statistical applications (see (Radivojević and Akhmatskaya, 2017) for a detailed discussion). The IACF in (6.15) gives a quantitative measure of the time required, on average, to generate an uncorrelated sample. Low values of measured IACFs imply low correlations between samples and thus more efficient sampling.

Figure 6.8 (left) presents the IACFs (normalized with respect to computational time) obtained from GSHMC simulations using different integrators and time steps. The simulations with e-MAIA provided the lowest values of IACFs and thus the best sampling for all choices of time step. All methods showed the good performance at $\Delta t = 40$ fs and, for this time step, the simulations with e-MAIA resulted in efficiency (as measured by IACF) from 5 (vs. M-BCSS, VV) to 9 (vs. AIA) times higher than the simulations with other integration schemes. For the largest time step, $\Delta t = 50$ fs, the performance achieved using e-MAIA was 12 times better than in the simulations with M-BCSS and M-ME. However, it did not differ any more

from those observed in the simulations with VV and AIA since for this long time step both AIA and e-MAIA chose velocity Verlet as an integrator.

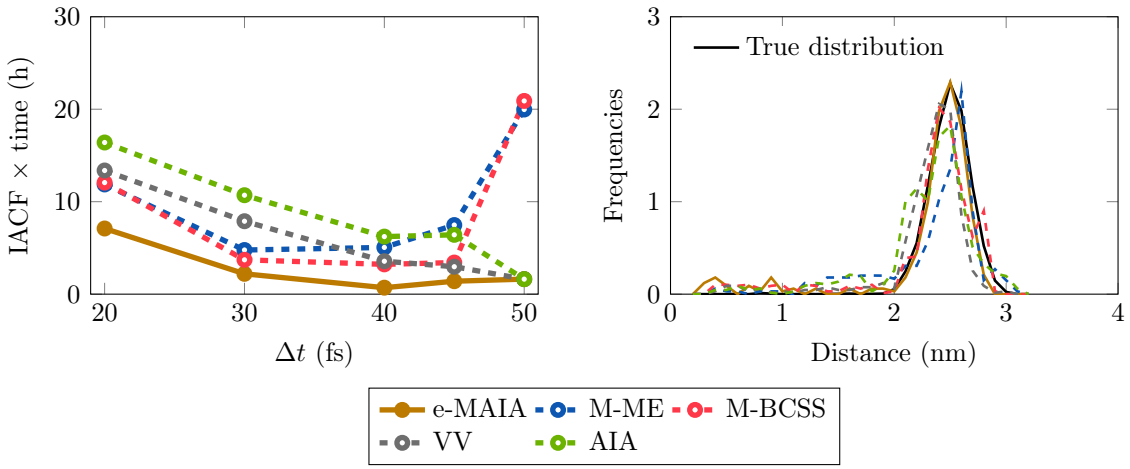


FIGURE 6.8: Toxin. Sampling efficiency of GSHMC combined with the integrators used in Figure 6.7. On the left, IACF of the drift, d, of the toxin to the preferred interfacial location evaluated as a function of Δt in GSHMC tests. On the right, the distribution of d observed in GSHMC simulations with various integrators using a time step of 30 fs. The solid black line (right) presents the "true" distribution produced with a ten times longer simulation (200 ns).

The right panel of Figure 6.8 compares the distributions of the distance d between the c.o.m. of the toxin and the c.o.m. of the bilayer, collected from simulations with $\Delta t = 30$ fs with different integrators, against a "true" distribution. Such distribution was obtained from an MD simulation with velocity Verlet, over a time interval of length 200 ns, i.e., ten times longer. As for all tests in this section, the plots have results averaged over 10 repetitive runs. The curve corresponding to the simulation with e-MAIA shows the best match with the "true" distribution.

The performances of e-MAIA and MAIA are compared in Figure 6.9. We chose the target AR_p in e-MAIA to be 90 % and the angle φ in MAIA to be equal to 1.1, which was the value found by e-MAIA for achieving the target $AR_p = 90$ % in GSHMC simulations at the smallest time step tested, $\Delta t = 20$ fs. Figure 6.9 reveals that, even though both e-MAIA and MAIA find the same integrator parameter b, leading to similar acceptance rates for positions, a good choice of the angle φ may visibly improve the sampling performance of GSHMC. The improvement is by factors of 8 and 2 for $\Delta t = 40$ fs and $\Delta t = 50$ fs, respectively. The evolution, as the time step increases, of the optimal parameter φ as calculated by e-MAIA is also shown in Figure 6.9 (right).

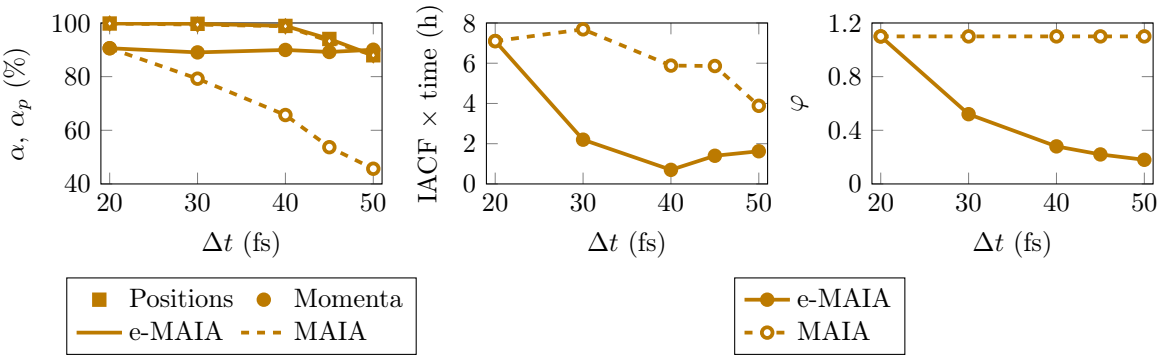


FIGURE 6.9: Toxin. e-MAIA (solid) vs. MAIA (dashed). Acceptance rates for positions and momenta (left), IACFs (center) and the angle φ (right) found by e-MAIA as a function of the time step (right) observed in GSHMC simulations. The angle φ used in MAIA was 1.1 and the target AR_p for e-MAIA was 90 %.

To finalize the numerical experiments on the toxin benchmark, we compared, using the normalized IACF metrics, the performance of three sampling methods, MD, HMC, and GSHMC. For each method, the best performing integrator was selected. Thus, GSHMC was combined with e-MAIA, based on the findings discussed above, whereas the AIA integrator was used for HMC and MD, according to the recommendations in Section 3.6. Figure 6.10 (left) demonstrates the superiority of GSHMC over the other two methods, regardless the choice of time step. For the optimal choice of time step for this system, namely, $\Delta t = 40$ fs, the sampling efficiency of GSHMC is 4 times higher than that of HMC and 11 times better than that of MD. For the longest time step, $\Delta t = 50$ fs, the difference is even more dramatic and expressed in improvement factors of 17 and 30 over HMC and MD, respectively. Plotted in Figure 6.10 (right) are the distributions of the distance d between the c.o.m. of the toxin and the c.o.m. of the bilayer produced by GSHMC, HMC, and MD simulations using a time step of 30 fs. They also confirm the better convergence of the GSHMC results to the "true" distribution.

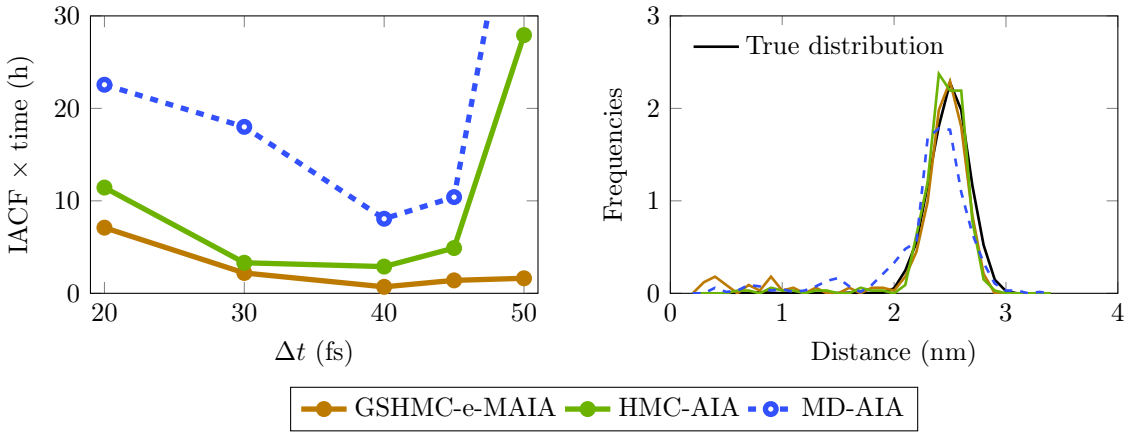


FIGURE 6.10: Toxin. Sampling efficiency: GSHMC (e-MAIA) vs. HMC (AIA) vs. MD (AIA). The best integrator for each sampling method was employed. Sampling efficiency was measured by means of IACFs (left) and the distribution of the distance between the toxin and the membrane bilayer (right). The solid black line (right) presents the "true" distribution produced with a ten times longer simulation (200 ns).

6.5.2 Villin

As in the toxin case, we first inspected the acceptance rates for positions and momenta in GSHMC simulations with different integrators and found that the e-MAIA method worked as expected, i.e., provided the best position acceptance rates (Figure 6.11, left) and maintained the target momenta acceptance rate of 90 % (Figure 6.11, right) for all choices of time steps.

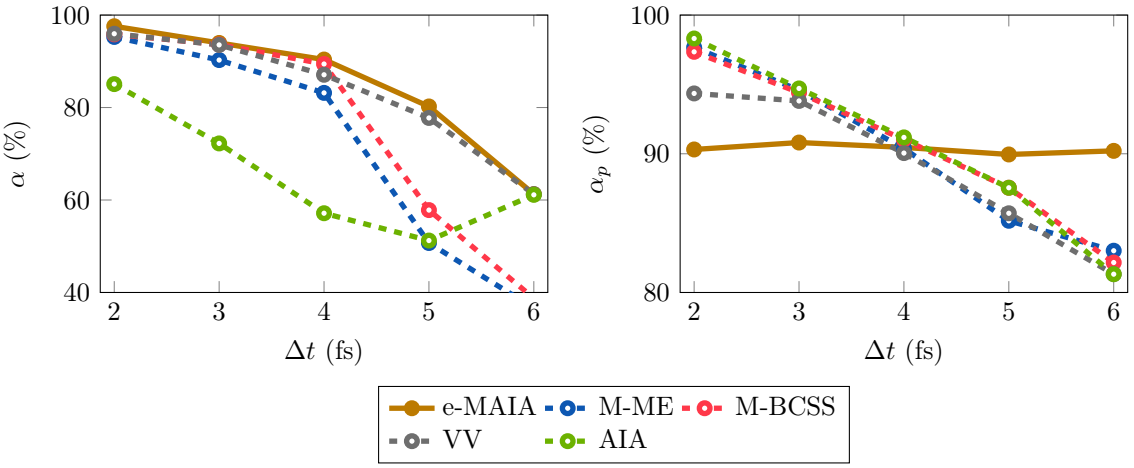


FIGURE 6.11: Villin. Acceptance rates for positions (left) and momenta (right) observed in GSHMC simulations when using M-BCSS, M-ME, VV, AIA (all dashed lines) and e-MAIA (solid line). e-MAIA maintains the target AR_p of 90 % for each value of Δt (right).

In contrast to the coarse-grained toxin benchmark, a quantitative analysis of the MAIA's contribution to the GSHMC performance gain is not feasible with the atomistic villin benchmark. Such an analysis would require a long, computationally demanding series of simulations, for a range of time steps, integrators, and sampling methods. It is, however, possible to find evidence of the positive impact of MAIA on the sampling efficiency of GSHMC by using comparatively short simulations of 5 ns and metrics directly related to the quality of sampling.

One of such metrics is the radius of gyration (RG), which provides an estimation of the compactness of the desired structure, and is computed as

$$\mathrm{RG} = \left(\frac{\sum_{i=1}^{n} \|r_i\|^2 M_i}{\sum_{i=1}^{n} M_i} \right)^{1/2},$$

where n is the number of atoms in the structure, r_i the distance between atom i and the center of mass of the structure, and M_i the mass of atom i. As in the study by van der Spoel and Lindahl, 2003, we considered the experimental value of 0.94 nm as a target value and investigated the level of convergence to this value in short simulations when using different time steps, numerical integrators, and simulation methods.

Another metric used in this study relates to the positional root-mean-squared deviation (RMSD). The RMSD of a group of atoms in a molecule with respect to a reference structure can be calculated as

$$\mathrm{RMSD} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \delta_i^2},$$

where δ_i is the distance between the positions of atom i in the two structures being compared.

Following the ideas from van der Spoel and Lindahl, 2003, we calculated the maximal RMSD of the α-carbon between any two visited structures in each simulation in order to judge the level of exploration of conformational space during the simulation.

In Figure 6.12 we plot, as functions of the time step, the radii of gyration and maximal RMSDs of the α-carbon calculated from the data collected in GSHMC simulations using e-MAIA, M-BCSS, M-ME, VV and AIA integrators. The simulations with e-MAIA (solid line) produced the best approximations to the experimental data (left plot), the highest values of maximal RMSD (right plot) (implying better sampling) and the smallest performance degradation at the longest time steps.

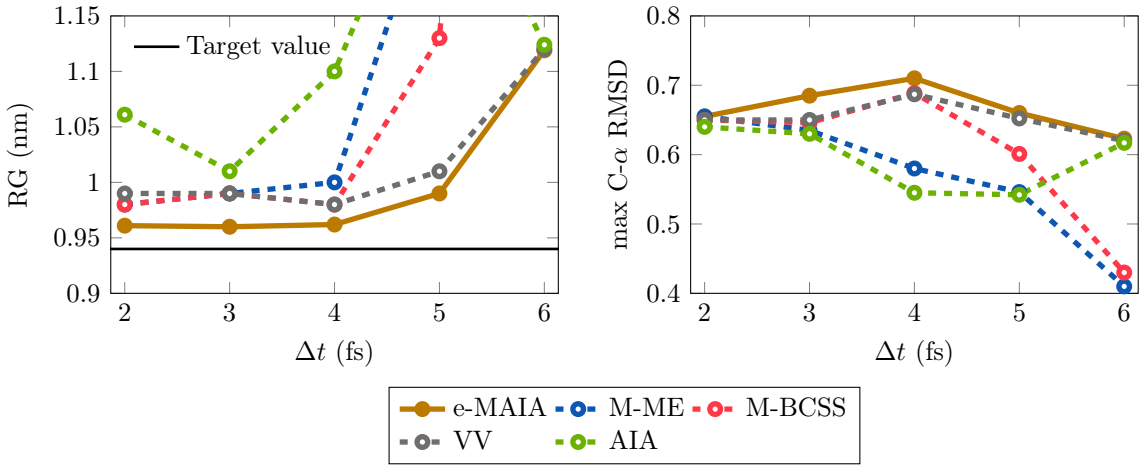


FIGURE 6.12: Villin. Sampling efficiency of GSHMC combined with the integrators used in Figure 6.11: radius of gyration (left) and maximum RMSD of the α-carbon of the protein (right). The solid black line (left) represents the target experimental value of 0.94 nm.

The comparison of the results obtained using MAIA and e-MAIA in GSHMC simulations of villin confirmed the trends observed earlier in the toxin tests. Both methods achieved almost the same position acceptance rates, whereas the momenta acceptance rates were significantly higher in the simulations with e-MAIA (Figure 6.13, left). The latter was possible due to the automatic tuning of the parameter φ provided by e-MAIA for maintaining the target $AR_p = 90\%$ (Figure 6.13, right); its positive effect can be noticed in Figure 6.13, center.

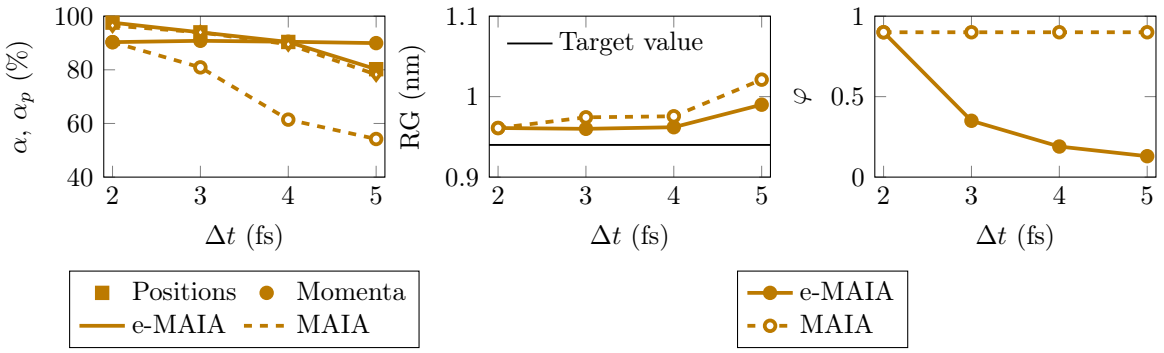


FIGURE 6.13: Villin. e-MAIA (solid) vs. MAIA (dashed). Acceptance rates for positions and momenta (left), radii of gyration (center) and the angle φ found by e-MAIA as a function of the time step (right) observed in GSHMC simulations. The angle φ used in MAIA was 0.9 and the target AR_p for e-MAIA was 90 %.

Figure 6.14 compares the radii of gyration (left) and maximal RMSDs of the α-carbon (right) obtained from the simulations of villin using three different sampling methods, GSHMC, HMC, and MD. As in the toxin case, the best performing integrator was used for each sampler, i.e., e-MAIA was selected for GSHMC and AIA was combined with HMC and MD. For both metrics, GSHMC demonstrated the best results over the range of time steps. Its advantage over HMC and MD is most visible at longer time steps when both HMC and MD lose accuracy and sampling efficiency.

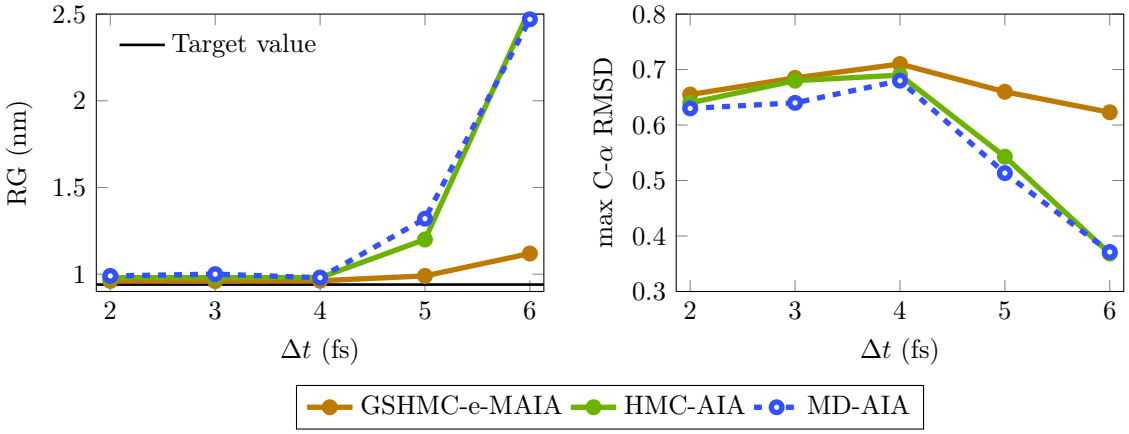


FIGURE 6.14: Villin. Sampling efficiency: GSHMC (e-MAIA) vs. HMC (AIA) vs. MD (AIA). The best integrator for each sampling method was employed. Sampling efficiency was measured through the radius of gyration (left) and the maximum RMSD of the α-carbon of the protein (right). The solid black line (left) represents the target experimental value of 0.94 nm.

Some extra explanations can be found in Figure 6.15, where the evolution with time of the relative radii of gyration observed for each simulation method with respect to the results obtained in MD simulations. We have calculated the radii of gyration for all three methods using two different simulations lengths, the whole simulation and half of it. This plot demonstrates that the difference in performance between GSHMC with e-MAIA and MD

with AIA increases with increasing simulation time. This suggests that, for more realistic simulation times, at least as good, but likely the bigger improvements, of GSHMC over MD can be expected. The same conclusions can be made when comparing GSHMC with HMC or HMC with MD.

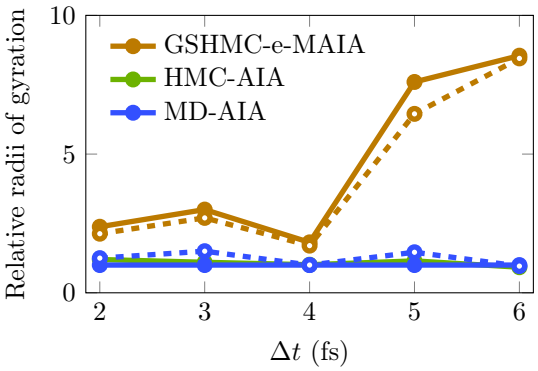
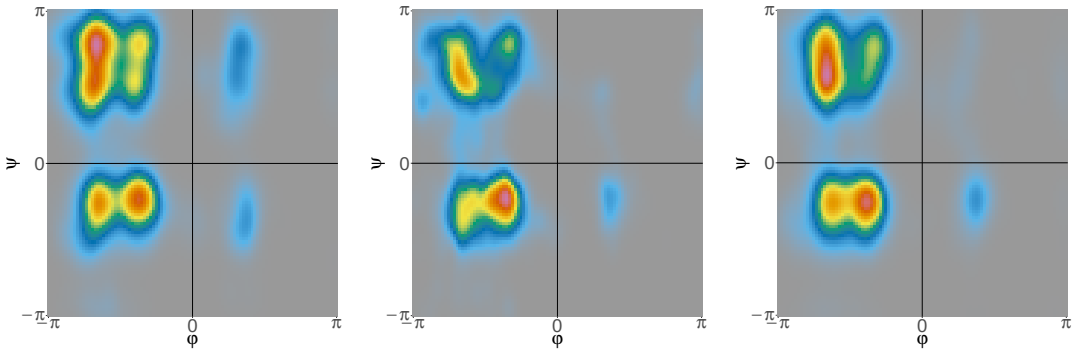


FIGURE 6.15: Villin. Evolution with time of the relative radii of gyration (RG) observed for each simulation method with respect to the RG found in MD simulations. The dashed lines represent the RG at half of the simulation time whereas the solid lines are used for the full simulations.

Additionally, we have generated Ramachandran plots considering all residues of the protein except for glycine. In Figure 6.16 the Ramachandran plots, obtained for the largest time step $\Delta t = 6$ fs, are presented as two-dimensional joint distributions of φ and Ψ angles. Figure 6.16 confirms the advantages of GSHMC over other tested methods. Indeed, GSHMC combined with e-MAIA is the only method capable of sampling all regions including the less populated basins in the $\varphi, \Psi > 0$ region, which were out of reach for HMC and MD sampling.



(A) GSHMC with e-MAIA.

(B) HMC with AIA.

(C) MD with AIA.

FIGURE 6.16: Villin. Sampling efficiency: GSHMC (e-MAIA) vs. HMC (AIA) vs. MD (AIA). Ramachandran plots for all residues of the protein except for glycine with φ torsion on the horizontal axis and Ψ on the vertical axis. The best integrator for each sampling method was employed. The time step was 6 fs, the largest in these tests.

Obviously, a deep atomistic study of the villin folding requires significantly longer runs than those presented here, as well as the incorporation of additional sampling techniques, such as, for example, parallel tempering, to the simulations. The latter can be implemented similarly, with a similar cost for all three methodologies considered in our study. However, the simulations will undoubtedly be more efficient if the underlying sampling method provides higher sampling efficiency, which is the case for GSHMC with e-MAIA.

6.6 Conclusions

We have introduced the new multi-stage integrators for modified Hamiltonian Monte Carlo (MHMC) methods. The proposed two- and three-stage integration methods provide better conservation of modified Hamiltonians than does the Verlet integrator, commonly used in MHMC. Each of the derived methods is characterized by its coefficients, which are obtained from the minimization of the (expected) error in modified Hamiltonians introduced by numerical integration. The new methods were tested and compared with Verlet and also with the sophisticated splitting integrators previously suggested for sampling with HMC.

For two-stage modified integrators, we have also proposed an adaptive integration approach ultimately leading to enhancing the accuracy and sampling efficiency of modified Hamiltonian Monte Carlo (MHMC) methods. Given a simulation system and a user-chosen time step, the Modified Adaptive Integration Approach (MAIA) identifies the two-stage numerical integrator which, when used in the Hamiltonian dynamics step of an MHMC method, provides the best conservation of the relevant modified Hamiltonian and thus the highest acceptance of the proposed trajectories. An enhanced variant of MAIA, e-MAIA, tailored to Generalized Shadow Hybrid Monte Carlo (GSHMC) methods, additionally supplies a value of the parameter φ that, for the problem under consideration, keeps the momentum acceptance at a user-desired level. The MAIA algorithm has been implemented, with no computational overhead during simulations, in MultiHMC-GROMACS, the modified version of the popular software package GROMACS. The effect of the use of MAIA on the sampling efficiency of GSHMC has been demonstrated by using constrained atomistic and unconstrained coarse-grained benchmarks and compared with the performance of other suitable integration schemes, including the popular velocity Verlet integrator. The tests revealed that the replacement in GSHMC of any fixed two-stage integrator with e-MAIA leads systematically to improvements in sampling efficiency of up to an order of magnitude. The performance comparison of GSHMC, HMC, and MD combined with their best choices of numerical integrators (e-MAIA, AIA, AIA, respectively) confirmed the efficiency and robustness of the GSHMC-MAIA combination, whose advantages are especially noticeable when using the longest possible simulation time steps. For such cases, GSHMC, while maintaining good accuracy in simulation, provided a sampling efficiency (as measured with IACF) up to 30 times higher than the efficiency that may be achieved with MD.

6.7 Published papers

1. B. Escribano, A. Lozano, T. Radivojević, **M. Fernández-Pendás**, J. Carrasco, and E. Akhmatskaya (2017). "Enhancing sampling in atomistic simulations of solid-state materials for batteries: a focus on olivine NaFePO4". In: *Theoretical Chemistry Accounts* 136.4, p. 43. URL: http://dx.doi.org/10.1007/s00214-017-2064-4

2. E. Akhmatskaya, **M. Fernández-Pendás**, T. Radivojević, and J. M. Sanz-Serna (2017). "Adaptive splitting integrators for enhancing sampling efficiency of modified Hamiltonian Monte Carlo methods in molecular simulations". In: *Langmuir* 33.42, pp. 11530–11542. URL: https://doi.org/10.1021/acs.langmuir.7b01372

3. T. Radivojević, **M. Fernández-Pendás**, J. M. Sanz-Serna, and E. Akhmatskaya (2018). "Multi-stage splitting integrators for sampling with modified Hamiltonian Monte Carlo methods". In: *submitted*

Chapter 7

Implementation. The MultiHMC-GROMACS Package

7.1 Introduction

Typical systems studied with molecular simulations consist of a vast number of atoms. Thus, high-speed computers are essential for performing the computationally-intensive simulations (cf. (Board et al., 1994; Klepeis et al., 2009)). Despite the fact that the first simulations of homogeneous systems had been performed in the 50s, macromolecular applications only started to become feasible in the mid-80s with the advent of high-speed parallel computing. On the other hand, the progress in macromolecular simulations could not be possible without advanced numerical algorithms and their efficient implementation on high-performance computers. Some essential algorithms proposed for molecular simulation have been reviewed in Chapter 1. While the whole range of methods developed until now is not covered there, the mentioned algorithms are the most relevant to the topic of this dissertation. All these methods are commonly used in MD simulations and available in all popular modern MD software packages, such as GROMACS (Berendsen, van der Spoel, and van Drunen, 1995; Hess et al., 2008), Amber (Salomon-Ferrer, Case, and Walker, 2013), LAMPPS (Plimpton, 1995), Desmond (Bowers et al., 2006), CHARMM (Brooks et al., 2009), NAMD (Nelson et al., 1996), etc. Each algorithm and software package have their limitations, and it is a user's responsibility to choose (and tune if necessary) the most appropriate method/package for the problem of interest.

The new algorithms presented in this dissertation have been implemented in the modified version of GROMACS, developed in BCAM and called MultiHMC-GROMACS. This package has been used to produce all numerical results presented in this study. GROMACS is a popular MD software package available under the GNU Lesser General Public License. It is written in the C programming language, highly optimized for maximal computational efficiency and fully parallelized using the MPI protocol. The package is used primarily for performing molecular dynamics simulations. It supports most important algorithms expected from a modern molecular dynamics implementation. In the following sections, the MultiHMC-GROMACS package, partially developed during the Ph.D., will be explained in detail.

7.2 MultiHMC-GROMACS: Overview

GROMACS is one of the most computationally efficient and versatile molecular dynamics packages available today. Its open-source nature makes it an excellent choice for implementing and benchmarking new methods. GROMACS supports state-of-the-art molecular dynamics algorithms and offers an extremely fast calculation of non-bonded atomic interactions, which usually are the dominant part of molecular dynamics simulations. Its main structure is summarized in Figure 7.1.

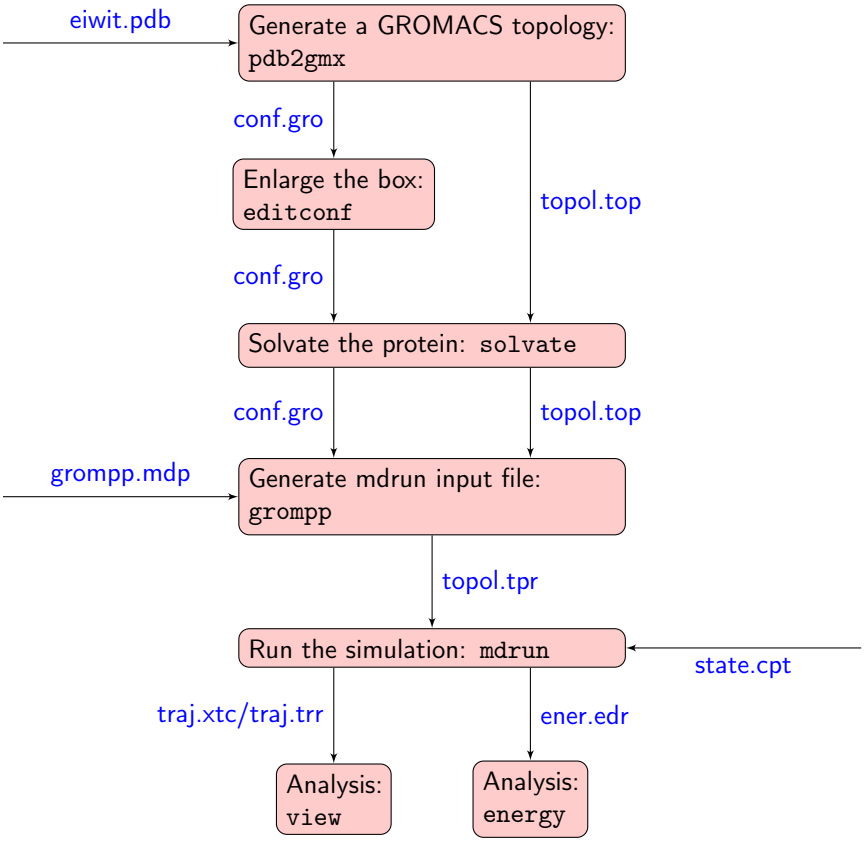


FIGURE 7.1: Main structure of the GROMACS package. The core GROMACS modules are shown in red, whereas the important files are highlighted in blue.

MultiHMC-GROMACS has been developed to achieve better accuracy and sampling performance in GROMACS through the use of Hybrid Monte Carlo methods and multi-stage numerical integrators (the details of those techniques can be found in the previous chapters). Currently, MultiHMC-GROMACS is based on GROMACS 4.5.4 (Pronk et al., 2013). However, its migration to later versions of GROMACS, to take advantage of CUDA-based GPU acceleration (Páll et al., 2015; Abraham et al., 2015), is underway. The first implementation of MultiHMC-GROMCAS has been described in detail in (Escribano, Akhmatskaya, and Mujika, 2013). During the elaboration of this dissertation, the following new features have been introduced to MultiHMC-GROMACS:

- Hybrid Monte Carlo (HMC) and Generalized Hybrid Monte Carlo (GHMC). More details will be provided in this chapter.

- The reduced-flipping method by Wagoner and Pande, 2012. It is a straightforward extension of the GSHMC implementation, which introduces an optional flipping in case of a proposal rejection, in step 11 of the Algorithm 3 (Section 4.3).
- Andersen barostat and its combination with Generalized Shadow Hybrid Monte Carlo (GSHMC). The implementation has been discussed in detail in Section 5.2.2.
- New shadow Hamiltonians for the GSHMC method. Such Hamiltonians have been explained in Section 4.3.1 and more details will be provided later in Section 7.3.
- Multi-stage integrators derived for HMC methods sampling with true and modified Hamiltonians. The introduction of multi-stage integrators in the code is discussed in this chapter.
- Extension of two-stage integrators to constrained dynamics using the RATTLE algorithm. It is explained in Section 3.4.
- Combination of two-stage integrators with MTTK, Nosé-Hoover and v-rescale thermostats. The details will be provided in Section 7.3.
- The adaptive integration approaches AIA, MAIA and e-MAIA. The details of the implementations can be found in Section 3.4 for AIA, and in Section 6.4 for MAIA and e-MAIA.

As stated above, the implementations of the novel algorithms developed in this dissertation have been already explained in the corresponding chapters. In this chapter, we present a general structure of MultiHMC-GROMACS and emphasize the important concepts and approaches proposed for the implementation of the key algorithms. We will mainly focus on the `mdrun` module unless otherwise specified since the majority of the algorithms implemented in MultiHMC-GROMACS have been introduced there. The new functionalities, as implemented in MultiHMC-GROMACS, do not interfere with the original GROMACS routines, aiming to maintain its performance and parallelization, since those are the strongest points of the package.

The chapter is structured as follows. The implementation of HMC methods is discussed in Section 7.3. The integration framework proposed in MultiHMC-GROMACS is explained in Section 7.4. We feature the main differences between MultiHMC-GROMACS and original GROMACS package and provide the conclusions in Sections 7.5 and 7.6, respectively.

7.3 HMC and GHMC as particular cases of GSHMC

The GSHMC method (cf. Section 4.3) had been previously implemented in MultiHMC-GROMACS as explained in detail in (Escribano, Akhmatskaya, and Mujika, 2013). We have extended this implementation to make it more general and introduced HMC and GHMC (cf. Chapter 2) as particular cases of GSHMC. It is clear that GHMC is just GSHMC without the modified Hamiltonians and with a less complicated partial momentum update (no Metropolis test required). Also, as explained in Table 2.1, HMC is a particular case of GHMC. To illustrate this in detail, we provide the structure of the GSHMC algorithm in Figure 7.2, where the notations of Section 4.3 are used.

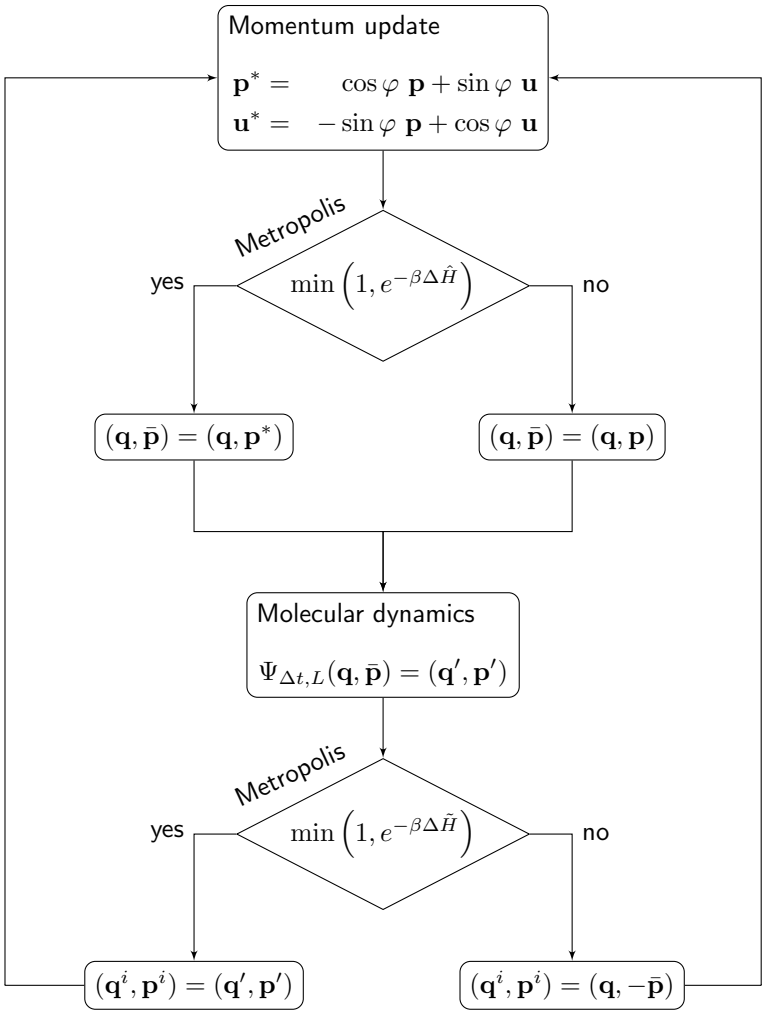


FIGURE 7.2: Structure of the GSHMC algorithm.

Figure 7.3 presents the GHMC algorithm as a particular case of GSHMC illustrated in Figure 7.2. The features specific to GHMC are highlighted in red. More precisely, there is no Metropolis test after the momentum update (which is equivalent to considering a test that is always accepted) and the shadow Hamiltonians now are substituted by the true Hamiltonians. The latter can be viewed as a shadow Hamiltonian whose expansion is of an order of two.

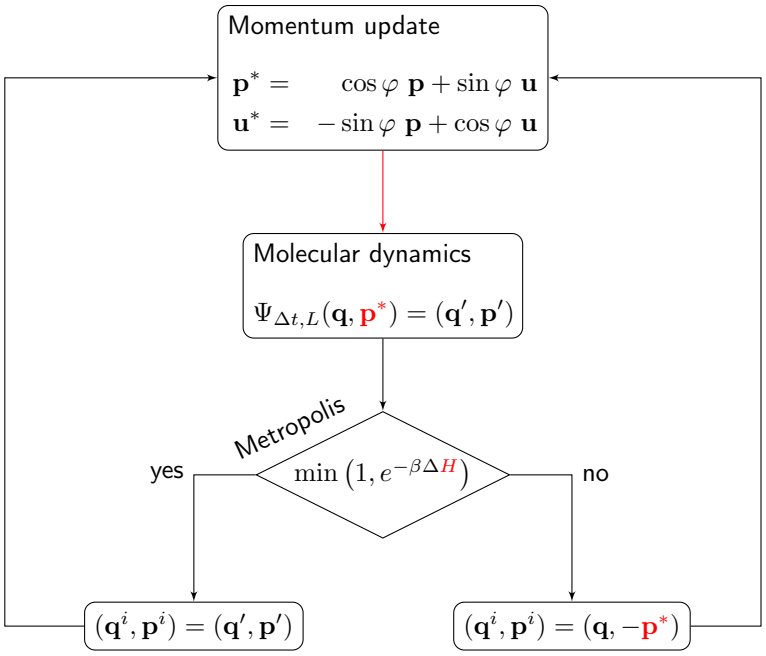


FIGURE 7.3: Structure of the GHMC algorithm as a special case of GSHMC.

The similar procedure as above can be applied to HMC. In Figure 7.4 the changes with respect to GHMC (Figure 7.3) are represented in red again. In this case, there is no partial momentum update; the momenta are always fully resampled with respect to the Maxwell-Boltzmann distribution. Thus, since the momenta are always completely discarded, after the Metropolis test, only the positions are stored.

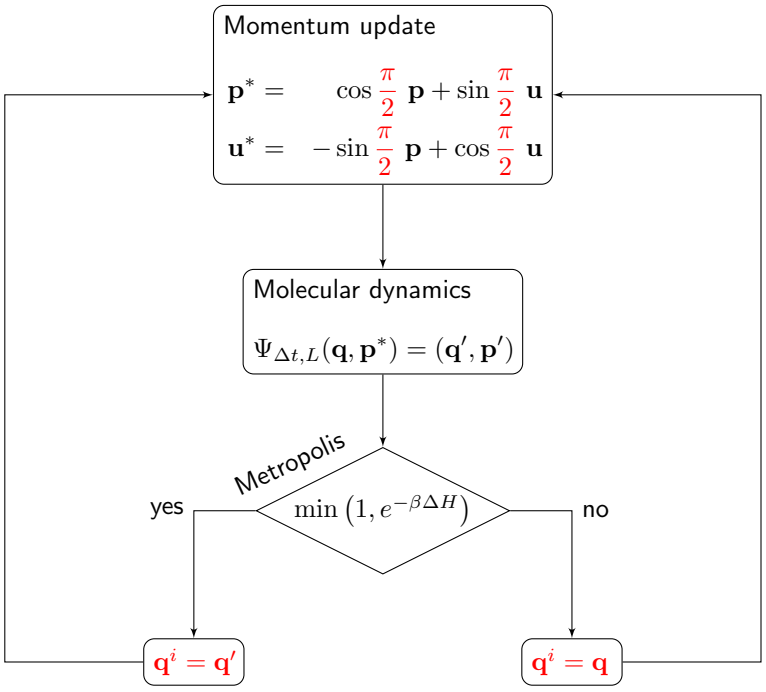


FIGURE 7.4: Structure of the HMC algorithm as a special case of GHMC.

Figure 7.5 shows how the GSHCM implementation (and then that of GHMC/HMC) fits into the structure of `do_md()`, the routine performing MD steps in GROMACS. We call this part of the code `gshmc()`, to emphasize the fact that the GSHMC method is the most sophisticated among the Hybrid Monte Carlo family. As HMC methods are not a part of the released GROMACS version, all functions inside of the `gshmc()` module do not belong to the original GROMACS library. The shadow Hamiltonians are implemented in the subroutine `shadow()` and Metropolis test is performed in the function `metropolis()`. The calculation of shadow Hamiltonians will be discussed in more detail later. It is clear that the `metropolis()` function can be easily used for the three methods HMC, GHMC and GSHMC. One only needs to pass to the subroutine the appropriate information. The function `momentum_update()` can be used in GSHMC, GHMC and HMC for partial momentum update. In the latter, φ is fixed to $\pi/2$. The `monte_carlo()` routine generates the noise u for the momentum update from the temperature T (see Section 4.11 for details).

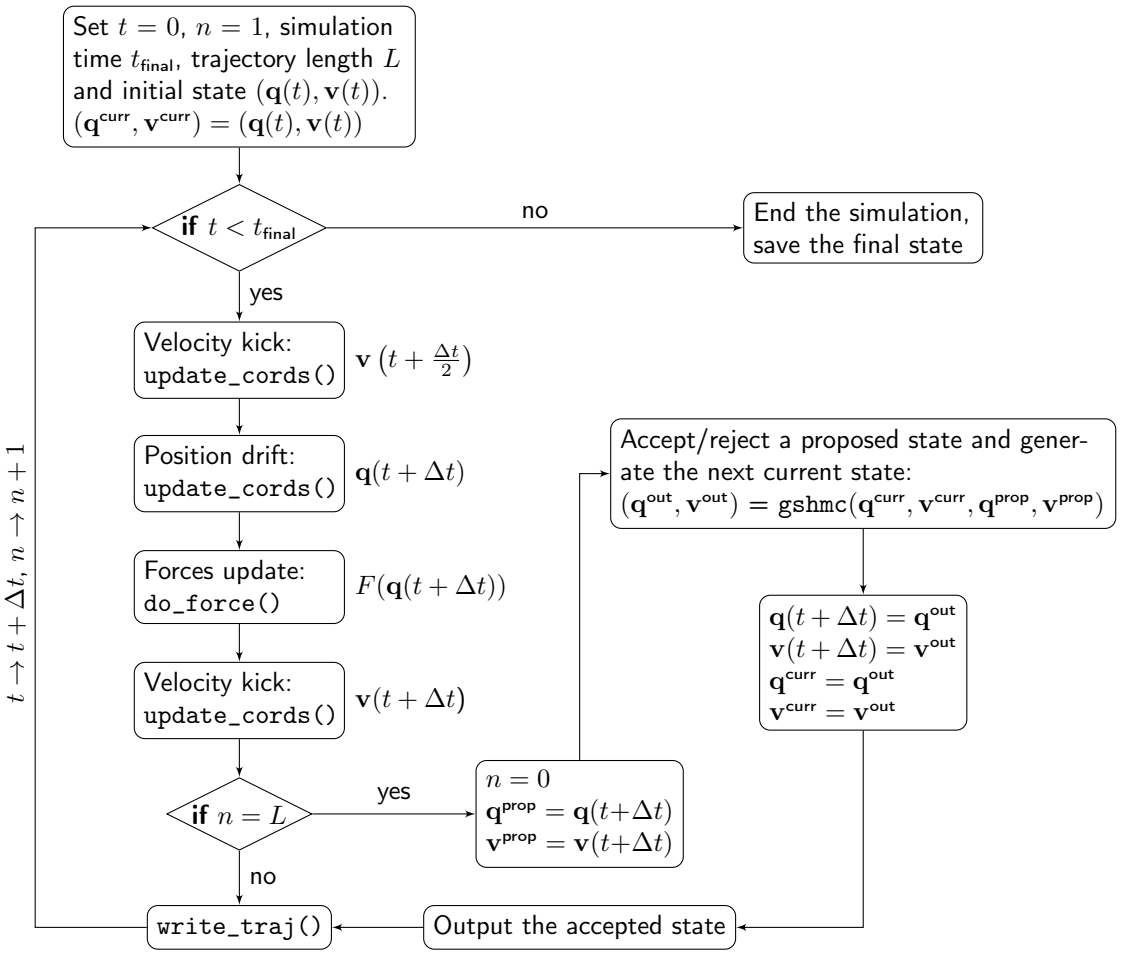


FIGURE 7.5: Update of configurations in MultiHMC-GROMACS.

The parameter file *.mdp* for GSHMC, GHMC and HMC has the following additional variables:

```
; Hybrid Monte Carlo methods =
method                   = GSHMC;      HMC / GHMC / GSHMC / MD
parameter_phi            = 0.2;        0<phi<pi/2
```

```
nr_mom_updates          = 1;        any positive integer
variable_change         = no;       yes / no
nr_MD_steps             = 1000;     any positive integer
hamiltonian_order       = 4;        4 / 6
shadow                  = BCH;      BCH / Legendre
canonical_temperature   = 310;      any positive rational
momentum_flip           = yes;      yes / no
reduced_flip            = no;       yes / no
```

The variable `method` decides whether to run HMC, GHMC, GSHMC or MD. The variables `parameter_phi` and `nr_MD_steps` correspond to φ and L in the notation of Section 4.3, respectively. Both `nr_mom_updates` and `variable_change` are features that were suggested in the original formulation of the algorithm (Akhmatskaya and Reich, 2008) but are not considered in the experiments of this dissertation. It is clear that `hamiltonian_order` alludes to the order of the expansion of the shadow Hamiltonian (4.9). Currently, there are two options implemented, fourth and sixth order. The variable `shadow` chooses the formulation of the shadow Hamiltonians. Now there are two options available, `Legendre` and `BCH`. `Legendre` is the one considered in the original GSHMC paper (Akhmatskaya and Reich, 2008) and derived discretizing the Lagrangian (cf. (4.9)). `BCH` is obtained using the BCH formula (Radivojević, 2016) (cf. (4.14)). The temperature used in the momentum update step is fixed to `canonical_temperature`. The reader should notice that the HMC methods work as thermostats and thus this is the reference temperature. It has to be remarked also that the `momentum_flip` decides between the original algorithm, which flips momenta upon rejection (cf. Section 4.3), and a version without momentum flip (Akhmatskaya, Bou-Rabee, and Reich, 2009). A reduction of the flips upon rejection, as suggested by Wagoner and Pande, 2012, can also be selected with a `reduced_flip` parameter.

Since HMC and GHMC are implemented as particular cases of GSHMC, only part of the parameters above have to be used for those methods. Namely, the length of the trajectory, `nr_MD_steps`, and the canonical temperature, `canonical_temperature`, in both cases; the angle φ, `parameter_phi`, and the choice of momentum flip[1], `momentum_flip`, in the GHMC case.

7.3.1 Calculation of shadow Hamiltonians

The implementation of two types of shadow Hamiltonians has been discussed in Section 4.3.1. Here we remark some practical features of those implementations.

The original GSHMC method (Akhmatskaya and Reich, 2008) considers the modified Hamiltonians (4.9) (in this dissertation we limit ourselves to the 4th order case). In MultiHMC-GROMACS, these shadow Hamiltonians are selected for a simulation with the parameter `Legendre`. The derivatives in (4.9) are prohibitively expensive to compute analytically. Thus, they are approximated by the central differences method. For instance, the third order derivative of the position at time t_n can be calculated as

$$\mathbf{Q}^{(3)}(t_n) = \frac{-\frac{1}{2}\mathbf{Q}(t_{n-2}) + \mathbf{Q}(t_{n-1}) - \mathbf{Q}(t_{n+1}) + \frac{1}{2}\mathbf{Q}(t_{n+2})}{\Delta t^3} + \mathcal{O}(\Delta t^2).$$

[1]The reader should note that, since the momenta are completely refreshed in the HMC method, the momentum flip does not play any role.

Therefore, in order to calculate a shadow Hamiltonian at time t_n, two short trajectories have to be run, one forward (t_{n+1}, t_{n+2}) and another backward (t_{n-1}, t_{n-2}), and the generated positions have to be stored. The positions and momenta at time t_n also have to be stored during the shadow Hamiltonian calculation. The forward trajectory is followed by the backward trajectory initialized at the recovered state at t_n. Therefore, the calculation of the shadow Hamiltonians requires extra force calculations which leads to an increase in the computational costs. However, the trajectories run for the computation of the shadow Hamiltonians might be reused for other purposes. For instance, after a momentum update, the calculation of a new shadow Hamiltonian is required for the Modified Metropolis test (step 6 in the Algorithm 3 (Section 4.3)). If the new momentum is accepted, the two forward steps can be viewed as the two first steps of the MD trajectory used to generate the next proposal for the MDMC step.

In addition, we implemented another formulation of the shadow Hamiltonians as proposed in (4.10) and it can be selected for its use in a simulation with the parameter `BCH` in the *.mdp* file. The implementation of such shadow Hamiltonians is done in terms of derivatives of the momenta (4.14). Thus, the derivatives orders can be reduced by one with respect to (4.9). While in (4.9) the highest order derivative was the third, in (4.14) only first and second derivatives are required. As an illustration, the second order derivative of the momenta at time t_n can be calculated as

$$\ddot{\mathbf{P}}(t_n) = \frac{\mathbf{P}(t_{n-1}) - 2\mathbf{P}(t_n) + \mathbf{P}(t_{n+1})}{\Delta t^2} + \mathcal{O}(\Delta t^2).$$

Therefore, in this case, the shorter forward and backward trajectories are required for evaluation of a shadow Hamiltonian at time t_n; only t_{n+1} and t_{n-1} are needed. It means that the `BCH` shadow Hamiltonians in their current implementation are less expensive than the `Legendre` ones.

7.4 Integrators in MultiHMC-GROMACS

7.4.1 General integration framework

The integration of the equations of motion, in the presence of thermostat, barostat and constraints using the velocity Verlet integrator (VV), as implemented in GROMACS, is illustrated in Figure 7.6. The reader should note that the implementation is done in terms of velocities instead of momenta.

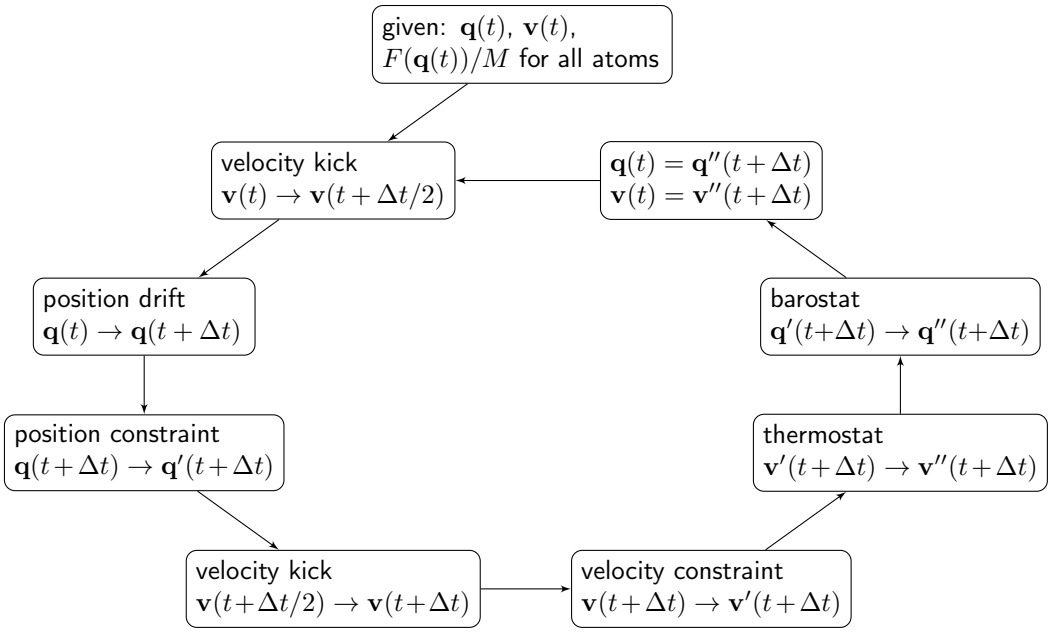


FIGURE 7.6: Flowchart of one integration step with velocity Verlet and a time step Δt in the presence of thermostat, barostat and constraints, as implemented in GROMACS.

Keeping in mind the scheme in Figure 7.6, multi-stage (two-, three- and four-stage) integrators proposed in Sections 3.2.2 and 6.2 have been implemented in MultiHMC-GROMACS.

The implementation of multi-stage integrators in MultiHMC-GROMACS is general enough to allow the use of all members of the families introduced previously for HMC and in this thesis for MHMC. The appropriate integrator can be selected with the variable `integrator` in the *.mdp* file. The values of the variable `integrator` corresponding to each available integrator are summarized in Table 7.1.

integrator	**parameter**	**reference**
Leapfrog	`md`	Feynman, Leighton, and Sands, 1964
Velocity Verlet	`md-vv`	Swope et al., 1982
BCSS2	`two-s`	Blanes, Casas, and Sanz-Serna, 2014
M-BCSS2	`two-s-m`	Radivojević et al., 2018
HOH	`two-s-HOH`	Predescu et al., 2012
ME	`two-s-minE`	McLachlan, 1995
M-ME2	`two-s-mme`	Radivojević et al., 2018
AIA	`aia`	Fernández-Pendás, Akhmatskaya, and Sanz-Serna, 2016
MAIA	`maia`	Akhmatskaya et al., 2017
BCSS3	`three-s`	Blanes, Casas, and Sanz-Serna, 2014
M-BCSS3	`three-s-m`	Radivojević et al., 2018
M-ME3	`three-s-mme`	Radivojević et al., 2018
BCSS4	`four-s`	Blanes, Casas, and Sanz-Serna, 2014

TABLE 7.1: Parameters used in the *.mdp* file to select an integrator in MultiHMC-GROMACS.

We notice that, except for `md` and `md-vv`, all numerical integrators have been implemented within this thesis.

Before we go to the detailed explanation of the implementation of integrators in MultiHMC-GROMACS, we would like to stress that after many years of development, the constant addition of new features, methods and bug-fixes by many different contributors has made the upper layers of the released version of GROMACS very convoluted. In particular, the main flow of the MD method resides in the function `do_md()`, which necessarily includes options for using all different integrators, constraints algorithms, temperature and pressure coupling methods, parallelization schemes, output writing... The implementation of a new algorithm by a GROMACS user will most likely require introducing changes to this function. Re-writing `do_md()` from scratch would demand a tremendous amount of work, both in terms of programming and testing, with a high probability that some functionalities would be broken in the process. Therefore, here we propose a more light-handed approach, needing only a partial revision of the function `do_md()` in order to make it cleaner and easier to follow. These changes helped us with the implementation of new integrators and Hybrid Monte Carlo methods. The better structuring of the code would also allow in the future for more obvious modularity and will make subsequent clean-ups more feasible than a complete re-writing of the whole function.

In GROMACS, the integration of the equations of motion is done in the `do_md()` function. Apart from initialization of all necessary structures, it mainly consists of a loop over the number of steps in which the function `update_coords()` is repeatedly evaluated. This loop allows for some flexibility. However, its structure can be changed aiming to have it simpler and more versatile. One of the input variables of the `update_coords()` function is a flag that indicates if the update of either velocities or positions is performed. For simplicity we will use the notations of `update_velocity()` and `update_position()` to refer to the two possible functionalities of the `update_coords()` function. In the original implementation of GSHMC (Escribano, Akhmatskaya, and Mujika, 2013), the structure of the integration of positions and velocities is as follows:

Algorithm 8 General integration framework

```
1: while number of integration steps do
2:     update_velocity()
3:     gshmc()
4:     update_velocity()
5:     update_position()
6:     update_forces()
7: end while
```

Here, `gshmc()` denotes the piece of the code which performs all the functions related to the GSHMC, GHMC and HMC methods.

Due to the ordering of the function calls, the logic in the algorithm above is not really intuitive, but it can be better understood in the following pseudo-code:

Algorithm 9 General integration framework rewritten

```
while number of integration steps do
    update_velocity()
    update_position()
    update_forces()
    update_velocity()
    gshmc()
end while
```

The structure in Algorithm 9 is the one we have adopted for the implementation of the multi-stage integrators. In the following sections, we discuss the details of the implementation of two- and three-stage integrators.

7.4.2 Two-stage integrators

Splitting schemes can be easily implemented directly from their Trotter expansions (cf. Section 3.2.3). The main *while* loop in `do_md()` is slightly modified with respect to Algorithm 9, but the idea of alternative updates of velocities and positions remains. The Liouville operator (cf. (3.39)) can be written as

$$i\hat{L} = i\hat{L}_{\mathbf{v}} + i\hat{L}_{\mathbf{q}},$$

where $i\hat{L}_{\mathbf{v}}$ and $i\hat{L}_{\mathbf{q}}$ are the deterministic Newtonian evolution of velocity and positions, respectively

$$i\hat{L}_{\mathbf{v}} = \frac{F(\mathbf{q})}{M}\frac{\partial}{\partial \mathbf{v}}, \qquad i\hat{L}_{\mathbf{q}} = \mathbf{v}\frac{\partial}{\partial \mathbf{q}}.$$

Thus, the two-stage splitting schemes from Section 3.2.2.1 can exploit the flexibility of the general Trotter formulation:

$$e^{i\hat{L}\Delta t} \approx e^{i\hat{L}_{\mathbf{v}} b\Delta t} \cdot e^{i\hat{L}_{\mathbf{q}}\frac{\Delta t}{2}} e^{i\hat{L}_{\mathbf{v}}(1-2b)\Delta t} \cdot e^{i\hat{L}_{\mathbf{q}}\frac{\Delta t}{2}} \cdot e^{i\hat{L}_{\mathbf{v}} b\Delta t}, \tag{7.1}$$

with $0 \leq b \leq 1/4$. As explained in Section 3.2.2.1 (cf. (3.26)), equation (7.1) can be written as

$$e^{i\hat{L}\Delta t} \approx \left(e^{i\hat{L}_{\mathbf{v}} b\Delta t} \cdot e^{i\hat{L}_{\mathbf{q}}\frac{\Delta t}{2}} \cdot e^{i\hat{L}_{\mathbf{v}}(1/2-b)\Delta t}\right) \cdot \left(e^{i\hat{L}_{\mathbf{v}}(1/2-b)\Delta t} \cdot e^{i\hat{L}_{\mathbf{q}}\frac{\Delta t}{2}} \cdot e^{i\hat{L}_{\mathbf{v}} b\Delta t}\right). \tag{7.2}$$

From the expression above, it is straightforward to suggest an appropriate modification for the *while* loop in Algorithm 9, since the two parentheses in (7.2) can be viewed as two "steps" of an asymmetric velocity Verlet integrator. Thus, one step of (7.1) corresponds to two laps of Algorithm 9 where the parameter b is an input of function `update_velocity()`. Therefore, it is not necessary to add extra function evaluations in Algorithm 9. The current implementation is summarized in Algorithm 10, emphasizing the dependence on b of `update_velocity()`.

Algorithm 10 Two-stage integrators implementation

```
while 2 × number of integration steps do
    update_velocity(b)
    update_position()
    update_forces()
    update_velocity(b)
    gshmc()
end while
```

As it has repeatedly been remarked, the fact that the number of integration steps is doubled in the *while* loop in Algorithm 10 does not increase the computational cost with respect to the standard Verlet for a correct choice of a time step vs. a length of an MD trajectory (cf. Section 3.2.2.3).

With two-stage integrators, the kinetic energies are calculated as in the velocity Verlet case. The averages are done after the positions and velocities are updated half step, and after they are updated the whole step. Thus, positions and velocities are synchronized in time.

If one uses the neighbor list parameter n as a frequency for updating the long-range forces, the two-stage formulation can be directly combined with a multi-step formulation such as the RESPA algorithm by Tuckerman, Berne, and Martyna, 1991. If one splits the forces $F(\mathbf{q})$ into short and long-range forces as

$$F(\mathbf{q}) = F_s(\mathbf{q}) + F_l(\mathbf{q}),$$

one can also split the Liouville operator $i\hat{L}_{\mathbf{v}}$ in

$$i\hat{L}_{\mathbf{v}s} = \frac{F_s(\mathbf{q})}{M}\frac{\partial}{\partial \mathbf{v}}, \qquad i\hat{L}_{\mathbf{v}l} = \frac{F_l(\mathbf{q})}{M}\frac{\partial}{\partial \mathbf{v}}.$$

Then, (7.1) reads as

$$e^{i\hat{L}\Delta t} \approx e^{i\hat{L}_{\mathbf{v}l}\frac{\Delta t}{2}} \cdot \left(e^{i\hat{L}_{\mathbf{v}s}b\Delta t} \cdot e^{i\hat{L}_{\mathbf{q}}\frac{\Delta t}{2}} \cdot e^{i\hat{L}_{\mathbf{v}s}(1-2b)\Delta t} \cdot e^{i\hat{L}_{\mathbf{q}}\frac{\Delta t}{2}} \cdot e^{i\hat{L}_{\mathbf{v}s}b\Delta t}\right)^n \cdot e^{i\hat{L}_{\mathbf{v}l}\frac{\Delta t}{2}}.$$

This algorithm is not implemented yet, but the flexibility of the formulation of Algorithm 10 would allow its easy introduction.

7.4.2.1 Combining two-stage integrators with thermostats and barostats

In the case of using a thermostat, such as Nosé-Hoover (Nosé, 1984b; Hoover, 1985), we can also express the time evolution of a system in terms of Liouville operators. In GROMACS, for ensuring the ergodicity of the sampling, the Nosé-Hoover chain approach is used (Martyna, Klein, and Tuckerman, 1992). For the sake of simplicity, we take the chains of length 1. Thus, one can define the following Liouville operator (cf. (Martyna et al., 1996))

$$i\hat{L}_{\text{NHC}} = -\frac{p_\xi}{Q}\mathbf{v}\frac{\partial}{\partial \mathbf{v}} + \frac{p_\xi}{Q}\frac{\partial}{\partial \xi} + (T - T_0)\frac{\partial}{\partial p_\xi},$$

where ξ is the heat bath variable (that has its own momenta p_ξ), Q is a constant that acts as a mass, and T is the temperature and T_0 is the reference temperature. Thus,

$$\begin{aligned} e^{i\hat{L}\Delta t} \approx & e^{i\hat{L}_{\text{NHC}} b\Delta t} \cdot \left(e^{i\hat{L}_{\mathbf{v}} b\Delta t} \cdot e^{i\hat{L}_{\mathbf{q}} \frac{\Delta t}{2}} \cdot e^{i\hat{L}_{\mathbf{v}} (1/2-b)\Delta t} \right) \cdot e^{i\hat{L}_{\text{NHC}} (1/2-b)\Delta t} \cdot \\ & e^{i\hat{L}_{\text{NHC}} (1/2-b)\Delta t} \cdot \left(e^{i\hat{L}_{\mathbf{v}} (1/2-b)\Delta t} \cdot e^{i\hat{L}_{\mathbf{q}} \frac{\Delta t}{2}} \cdot e^{i\hat{L}_{\mathbf{v}} b\Delta t} \right) \cdot e^{i\hat{L}_{\text{NHC}} b\Delta t}. \end{aligned} \tag{7.3}$$

The v-rescale thermostat (Bussi, Donadio, and Parrinello, 2007) can also follow the similar scheme and the rescaling can be done in the same points in which the NHC's are evaluated. Actually, in the original formulation in (Bussi, Donadio, and Parrinello, 2007), the Nosé-Hoover thermostat is recast in a way that mimics the v-rescale approach.

The MTTK barostat has been formulated by Martyna et al., 1996. The idea is very similar to the formulation of the Nosé-Hoover terms of the same authors. In this case, they define the following Liouville operator associated to the barostat

$$i\hat{L}_{\text{NHC-baro}} = -\left(1 + \frac{d}{D}\right) \frac{p_\varepsilon}{W} \mathbf{v} \frac{\partial}{\partial \mathbf{v}} + \left(\frac{(dP_{int} - dP_{ext})V}{W} - \frac{p_\varepsilon}{W} \frac{p_\xi}{Q} \right) \frac{\partial}{\partial p_\varepsilon},$$

where p_ϵ is the momentum associated with the logarithm of the volume V ($\epsilon = \log V/d$) and W is the mass of the barostat. The pressure is calculated through the virial theorem. Thus, (7.3) reads as

$$\begin{aligned} e^{i\hat{L}\Delta t} \approx & e^{i\tilde{L}_{\text{NHC}} b\Delta t} \cdot \left(e^{i\hat{L}_{\mathbf{v}} b\Delta t} \cdot e^{i\hat{L}_{\mathbf{q}} \frac{\Delta t}{2}} \cdot e^{i\hat{L}_{\mathbf{v}} (1/2-b)\Delta t} \right) \cdot e^{i\tilde{L}_{\text{NHC}} (1/2-b)\Delta t} \cdot \\ & e^{i\tilde{L}_{\text{NHC}} (1/2-b)\Delta t} \cdot \left(e^{i\hat{L}_{\mathbf{v}} (1/2-b)\Delta t} \cdot e^{i\hat{L}_{\mathbf{q}} \frac{\Delta t}{2}} \cdot e^{i\hat{L}_{\mathbf{v}} b\Delta t} \right) \cdot e^{i\tilde{L}_{\text{NHC}} b\Delta t}, \end{aligned}$$

where $i\tilde{L}_{\text{NHC}} = i\hat{L}_{\text{NHC}} + i\hat{L}_{\text{NHC-baro}}$.

The different Trotter expansions above suggest how the Nosé-Hoover, v-rescale and MTTK algorithms have been implemented in MultiHMC-GROMACS for two-stage integrators. The updates associated with the thermostat/barostat are implemented in the routine `update_coupling()`. Such function is adapted to receive the parameter b as an input. Thus, the rescaling of the velocities can be accurately performed.

7.4.3 Three-stage integrators

As it has been remarked in Section 3.2.2.2, the three-stage integrators in (3.27) cannot be written in three equal velocity Verlet substeps similarly to (7.2). Thus, the ideas explained in the section above cannot be applied to this case and the general framework from Algorithm 9 leads to numerical instabilities after some steps. The most straightforward solution has been to add extra evaluations of the function `update_coords()` with the different parameters (which depend on b only as explained in Section 3.2.2.2). These additional evaluations of `update_coords()` are only considered in a case when the three-stage integrators are used. The current implementation is explained in Algorithm 11.

Algorithm 11 Three-stage integrators implementation

```
while number of integration steps do
    update_velocity()
    update_position()
    update_forces()
    update_velocity()
    update_position()
    update_forces()
    update_velocity()
    update_position()
    update_forces()
    update_velocity()
    gshmc()
end while
```

Special care has to be taken of the parallelization due to the extra force evaluations.

Clearly, Algorithm 11 is also applicable to the modified three-stage integrators (cf. Section 6.2).

7.5 MultiHMC-GROMACS vs GROMACS: Summary

- **Functionalities:** New functionalities introduced in MultiHMC-GROMACS and not available in the released version of GROMACS are summarized in Table 7.2.

sampling	HMC, GHMC, GSHMC
integrator	two-, three-, four-stage splitting integrators two- and three-stage modified splitting integrators AIA MAIA and e-MAIA
barostat	Andersen

TABLE 7.2: New functionalities in MultiHMC-GROMACS with respect to the released version of GROMACS.

- **Performance:** GSHMC combined with e-MAIA demonstrates up to 60 times better sampling efficiency than does MD combined with Verlet for some choices of time steps (Fernández-Pendás, Akhmatskaya, and Sanz-Serna, 2016; Akhmatskaya et al., 2017). The only methodology among the proposed and implemented in MultiHMC-GROMACS that introduces overheads is GSHMC (Escribano, Akhmatskaya, and Mujika, 2013). The current implementation's overheads are, on average, of 1-2 % with respect to MD with the v-rescale thermostat (Akhmatskaya et al., 2017).

- **Limitations:** MultiHMC-GROMACS is based on the version 4.5.4 of GROMACS described in (Pronk et al., 2013). The code is not available for its use with GPU parallelization (Páll et al., 2015; Abraham et al., 2015).

7.6 Conclusions

In this chapter, the MultiHMC-GROMACS software package has been presented. The implementation of new algorithms proposed in this thesis has been discussed in detail in the previous chapters. Here we supply the implementation details of the well-established methodologies presented in Chapters 2 and 3 which do not appear in the released version of GROMACS. The current structure of MultiHMC-GROMACS provides the flexibility for introducing different Hybrid Monte Carlo algorithms. Switching from one methodology to another is regulated by the values of input parameters. The MultiHMC-GROMACS code also offers a general framework for introducing new integrators and algorithms that can be expressed in a Trotter formulation. Two-, three- and four-stage integrators in original and modified formulations, and the adaptive integration schemes for HMC, MD and GSHMC (AIA, MAIA, e-MAIA) have been successfully implemented in MultiHMC-GROMACS. The two-stage integrators have also been combined with the v-rescale, Nosé-Hoover, and MTTK thermostats and barostats. Since the multi-step algorithms can be easily expressed in the Trotter form, the current structure allows for a smooth implementation of this kind of methodologies.

7.7 Published papers

1. **M. Fernández-Pendás**, B. Escribano, T. Radivojević, and E. Akhmatskaya (2014). "Constant pressure hybrid Monte Carlo simulations in GROMACS". In: *Journal of Molecular Modelling* 20.12, p. 2487. URL: http://dx.doi.org/10.1007/s00894-014-2487-y

2. **M. Fernández-Pendás**, E. Akhmatskaya, and J. M. Sanz-Serna (2016). "Adaptive multi-stage integrators for optimal energy conservation in molecular simulations". In: *Journal of Computational Physics* 327, pp. 434–449. URL: http://www.sciencedirect.com/science/article/pii/S0021999116304569

3. E. Akhmatskaya, **M. Fernández-Pendás**, T. Radivojević, and J. M. Sanz-Serna (2017). "Adaptive splitting integrators for enhancing sampling efficiency of modified Hamiltonian Monte Carlo methods in molecular simulations". In: *Langmuir* 33.42, pp. 11530–11542. URL: https://doi.org/10.1021/acs.langmuir.7b01372

Chapter 8

Conclusions, Future Work and Contributions

8.1 Conclusions

In this thesis, we developed the methodologies for enhancing the sampling abilities and improving the accuracy of the Hybrid Monte Carlo methods applied to molecular simulations. For this purpose, two main directions have been explored: splitting integrators and importance sampling.

First, we investigated an effect of splitting integration schemes on the performance of Hybrid Monte Carlo methods and proposed a novel methodology called the Adaptive Integration Approach (AIA). This algorithm offers, for any chosen time step, a system-specific integrator which guarantees the best energy conservation for harmonic forces achievable by an integrator from a family of two-stage splitting schemes, including velocity Verlet. While improvements in energy conservation do not necessarily imply dramatic changes in sampling, they improve acceptance rates in Hybrid Monte Carlo methods. The performed experiments showed that in molecular dynamics AIA leads to improvements of sampling as measured by the metrics considered. The improved sampling may arise as a consequence of either enhanced accuracy with a given time step or due to the possibility of longer time steps. The AIA scheme can be implemented, without introducing computational overheads in simulations, in any software package which includes MD and/or HMC. The analysis of integrated autocorrelation functions and folding evolution demonstrated, for selected sizes of time steps, that AIA possesses up to 5 times better sampling performance than the other tested schemes.

Though the first importance sampling hybrid Monte Carlo algorithms, or Modified hybrid Monte Carlo (MHMC), have been developed in the early 2000s and showed some promising results, our objective was to improve further accuracy and performance of such methods and to extend their applicability to a wide range of problems. The special attention was paid to a particular MHMC method, the Generalized Shadow Hybrid Monte Carlo (GSHMC). The GSHMC method was initially available only in the NVT ensemble. However, in this thesis, it was also adapted to the NPT ensemble using an Andersen barostat. The newly developed NPT-GSHMC method showed the same level of accuracy as was demonstrated by NPT-MD and NVT-GSHMC, and the comparable sampling efficiency to NVT-GSHMC, outperforming in this category NPT-MD. Then, the GSHMC algorithm, and HMC and GHMC also, were extended for the first time to the grand canonical ensemble. The validity of the three algorithms was proved in simulations of Lennard-Jones fluids at different conditions. All three new methods reproduced well the predicted data. Also, the new algorithms sampled up to 16 times better than a well-established MC algorithm. Among three new methods, GSHMC showed the best accuracy and sampling efficiency.

Finally, both approaches were combined: we introduced multi-stage integrators for enhanced sampling with modified Hamiltonian Monte Carlo (MHMC) methods. The proposed two- and three-stage integration methods provide better conservation of modified Hamiltonians than does the Verlet integrator commonly used in MHMC. For two-stage modified integrators, we also proposed an adaptive integration approach ultimately leading to enhancing the accuracy and sampling efficiency of MHMC methods. Given a simulation system and a user-chosen time step, the new algorithm called Modified Adaptive Integration Approach (MAIA) identifies the two-stage numerical integrator which, when used in the Hamiltonian dynamics step of an MHMC method, provides the best conservation of the relevant modified Hamiltonian and thus the highest acceptance of the proposed trajectories. An enhanced variant of MAIA, e-MAIA, tailored to GSHMC methods, additionally supplies a value of the parameter φ that, for the problem under consideration, keeps the momentum acceptance at a user-desired level. The MAIA algorithm was implemented, with no computational overhead during simulations, in MultiHMC-GROMACS. The effect of the use of MAIA on the sampling efficiency of GSHMC was demonstrated in simulations with constrained atomistic and unconstrained coarse-grained benchmarks and compared with the performance of other suitable integration schemes, including the famous velocity Verlet integrator. The tests revealed that the replacement in GSHMC of any two-stage integrator with e-MAIA leads systematically to improvements in sampling efficiency of up to an order of magnitude. The performance comparison of GSHMC, HMC, and MD combined with their best choices of numerical integrators (e-MAIA, AIA, AIA, respectively) confirmed the efficiency and robustness of the GSHMC-MAIA combination, whose advantages are especially noticeable when using the longest possible simulation time steps. For such cases, GSHMC, while maintaining good accuracy in simulation, provided a sampling efficiency (as measured with IACF) up to 30 times higher than the efficiency that may be achieved with MD.

The in-house software package called MultiHMC-GROMACS was also presented. The implementation of new algorithms proposed in the thesis was discussed in detail. The implementation details of the well-established methodologies that do not appear in the released version of GROMACS, such as HMC or the multi-stage integrators, were supplied. The current structure of MultiHMC-GROMACS provides the flexibility for introducing different Hybrid Monte Carlo algorithms. Switching from one methodology to another is regulated by the values of input parameters. The MultiHMC-GROMACS code also offers a general framework for introducing new integrators and algorithms that can be expressed in a Trotter formulation. Two-, three- and four-stage integrators in original and modified formulations as well as the adaptive integration schemes AIA, MAIA and e-MAIA were successfully implemented in MultiHMC-GROMACS. The two-stage integrators were also combined with the v-rescale, Nosé-Hoover and MTTK thermostats and barostats, and with the RATTLE algorithm for solving constraints. Since the multi-step algorithms can be easily expressed in the Trotter form, the current structure allows for a smooth implementation of this kind of methodologies.

8.2 Future work

Several ideas for future work can be suggested for improving the methods and results presented in this thesis.

The AIA and MAIA schemes proposed in Chapters 3 and 6 may be extended in a natural way to multiple-time step (MTS) algorithms such as those based on Reversible multiple time

scale molecular dynamics (Tuckerman, Berne, and Martyna, 1992), the Generalized Hybrid Monte Carlo method (Escribano et al., 2015), the Stochastic, resonance-free multiple time step algorithm (Leimkuhler, Margul, and Tuckerman, 2013), etc. AIA and MAIA ideas can be also easily extended to three-stage integrators.

The NPT-GSHMC method presented in Chapter 5 can be improved by introducing a weak coupling thermostat (Faller and de Pablo, 2002) or anisotropic changes in the box (Parrinello and Rahman, 1981). The grand canonical algorithms, also proposed in Chapter 5, are only valid for homogeneous systems. A future goal is to test them with rigid water models for the potential use in simulation of proteins in water. Another possible future direction is to improve acceptance rates of the new methods. In very dense systems, the placement of a new particle can play a fundamental role in sampling efficiency of Monte Carlo based methods, since a completely random placement can lead to dramatic changes of the energy and thus rejections in the Metropolis tests. One possible way for improving the placement of inserted particles and increasing the acceptance rates is combining the current algorithms with the cavity-based methods (Mezei, 1980; Mezei, 1987; Deitrick, Scriven, and Davis, 1989). Another possible extension to the algorithms presented here would be to allow continuous changes in D (Cağin and Pettitt, 1991; Lo and Palmer, 1995; Boinepalli and Attard, 2003) and investigate the effect of such changes on the overall efficiency of simulations.

As to the MultiHMC-GROMACS package presented in Chapter 7, some limitations of the version in use have to be overcome. Currently, MultiHMC-GROMACS is based on the version 4.5.4 of GROMACS (Pronk et al., 2013) and thus is not available for its use with GPU parallelization (Páll et al., 2015; Abraham et al., 2015). Therefore, the upgrade of MultiHMC-GROMACS to the latest version of GROMACS is another future task.

8.3 Contributions

The methodologies and results generated in this thesis have been published in the high impact journals and presented at the international conferences.

Publications:

1. **M. Fernández-Pendás**, B. Escribano, T. Radivojević, and E. Akhmatskaya (2014). "Constant pressure hybrid Monte Carlo simulations in GROMACS". In: *Journal of Molecular Modelling* 20.12, p. 2487. URL: http://dx.doi.org/10.1007/s00894-014-2487-y

2. **M. Fernández-Pendás**, E. Akhmatskaya, and J. M. Sanz-Serna (2016). "Adaptive multi-stage integrators for optimal energy conservation in molecular simulations". In: *Journal of Computational Physics* 327, pp. 434–449. URL: http://www.sciencedirect.com/science/article/pii/S0021999116304569

3. B. Escribano, A. Lozano, T. Radivojević, **M. Fernández-Pendás**, J. Carrasco, and E. Akhmatskaya (2017). "Enhancing sampling in atomistic simulations of solid-state materials for batteries: a focus on olivine NaFePO4". In: *Theoretical Chemistry Accounts* 136.4, p. 43. URL: http://dx.doi.org/10.1007/s00214-017-2064-4

4. E. Akhmatskaya, **M. Fernández-Pendás**, T. Radivojević, and J. M. Sanz-Serna (2017). "Adaptive splitting integrators for enhancing sampling efficiency of modified Hamiltonian Monte Carlo methods in molecular simulations". In: *Langmuir* 33.42, pp. 11530–11542. URL: https://doi.org/10.1021/acs.langmuir.7b01372

5. T. Radivojević, **M. Fernández-Pendás**, J. M. Sanz-Serna, and E. Akhmatskaya (2018). "Multi-stage splitting integrators for sampling with modified Hamiltonian Monte Carlo methods". In: *submitted*

Presentations:

1. **M. Fernández-Pendás**, B. Escribano, E. Akhmatskaya, and T. Radivojević (July 16, 2014). "Constant Pressure hybrid Monte Carlo simulations in GROMACS" (poster). In: CCP5 Summer School 2014 - Methods in Molecular Simulations, Manchester University (United Kingdom)

2. **M. Fernández-Pendás**, E. Akhmatskaya, and J. M. Sanz-Serna (September 15, 2015). "Adaptive multi-stage integrators with optimal energy conservation". In: SciCADE 2015, University of Potsdam (Germany)

3. **M. Fernández-Pendás**, E. Akhmatskaya, and J. M. Sanz-Serna (February 15, 2016). "Adaptive multi-stage integrators for optimal energy conservation in molecular simulations" (poster). In: CECAM workshop on Models for Protein Dynamics 1976-2016, CECAM-HQ-EPFL, Lausanne (Switzerland)

4. **M. Fernández-Pendás**, E. Akhmatskaya, and J. M. Sanz-Serna (September 29, 2016). "Adaptive multi-stage integrators for optimal energy conservation in molecular simulations" (poster). In: DIPC summer school on computational methods for biological molecules, DIPC, Donostia (Spain)

5. **M. Fernández-Pendás**, T. Radivojević, J. M. Sanz-Serna, and E. Akhmatskaya (January 26, 2017). "Adaptive splitting integrators for enhancing sampling efficiency of shadow Hamiltonian Monte Carlo methods". In: IMUVA Seminar on invitation of Prof. J. M. Sanz-Serna, Universidad de Valladolid (Spain)

Contributed presentations:

1. T. Radivojević, E. Akhmatskaya, B. Escribano, and **M. Fernández-Pendás** (January 22, 2014). "Momentum Flips in Generalized Hybrid/Hamiltonian Monte Carlo Methods". In: IMUVA Seminar on invitation of Prof. J. M. Sanz-Serna, Universidad de Valladolid (Spain)

2. E. Akhmatskaya, B. Escribano, **M. Fernández-Pendás**, and I. Terterov (December 4, 2014). "nhanced Sampling Hybrid Monte Carlo Methods in GROMACS Package: MultiHMC-GROMACS". In: ITQB seminar on invitation of Prof. A. Baptista, Universidade Nova de Lisboa (Portugal)

3. E. Akhmatskaya, **M. Fernández-Pendás**, T. Radivojević, and J. M. Sanz-Serna (September 11, 2017). "Adaptive two-stage integrators for sampling algorithms based on Hamiltonian dynamics". In: ICERM topical workshop “Stochastic numerical algorithms, multiscale modeling and high-dimensional data analytics”, Brown University (USA)

4. E. Akhmatskaya, **M. Fernández-Pendás**, T. Radivojević, and J. M. Sanz-Serna (September 11, 2017). "Adaptive splitting integrators for modified Hamiltonian Monte Carlo methods". In: SciCADE 2017, University of Bath (United Kingdom)

Appendix A

Stability Analysis

A.1 Calculation of the transition matrix for different integrators: Harmonic oscillator

We consider the harmonic oscillator with equations of motion (3.6) where the forces are $F(q) = -\omega^2 q$ and the masses are assumed to be trivial. Then, we show how to calculate the transition matrix S in (3.8) with the velocity Verlet integrator formulated as

1. $v(t + \Delta t/2) = v(t) + \frac{\Delta t}{2} F(q(t))$;
2. $q(t + \Delta t) = q(t) + \Delta t\ v(t + \Delta t/2)$;
3. $v(t + \Delta t) = v(t + \Delta t/2) + \frac{\Delta t}{2} F(q(t + \Delta t))$.

Thus, for a time step Δt and the problem defined in (3.7), the matrix S is computed as the product of the three evolutions of velocities and positions above:

$$\begin{aligned} S &= \begin{pmatrix} 1 & 0 \\ -\frac{\omega\Delta t}{2} & 1 \end{pmatrix} \begin{pmatrix} 1 & \omega\Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -\frac{\omega\Delta t}{2} & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 - \frac{(\omega\Delta t)^2}{2} & \omega\Delta t \\ -\omega\Delta t + \frac{(\omega\Delta t)^3}{4} & 1 - \frac{(\omega\Delta t)^2}{2} \end{pmatrix}. \end{aligned}$$

In the case of the two-stage integrators of the family (3.25) the matrix S for the harmonic oscillator case is calculated as

$$\begin{aligned} S &= \begin{pmatrix} 1 & 0 \\ -b\omega\Delta t & 1 \end{pmatrix} \begin{pmatrix} 1 & \frac{\omega\Delta t}{2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -(1-2b)\omega\Delta t & 1 \end{pmatrix} \begin{pmatrix} 1 & \frac{\omega\Delta t}{2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -b\omega\Delta t & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 - \frac{(\omega\Delta t)^2}{2} + b(1-2b)\frac{(\omega\Delta t)^4}{4} & \omega\Delta t + (2b-1)\frac{(\omega\Delta t)^3}{4} \\ -\omega\Delta t + b(1-b)(\omega\Delta t)^3 - b^2(1-2b)\frac{(\omega\Delta t)^5}{4} & 1 - \frac{(\omega\Delta t)^2}{2} + b(1-2b)\frac{(\omega\Delta t)^4}{4} \end{pmatrix}. \end{aligned}$$

In the case of the three-stage integrators of the family (3.27) the matrix S for the harmonic oscillator case is calculated as

$$\begin{aligned} S &= \left(\begin{smallmatrix} 1 & 0 \\ -b\omega\Delta t & 1 \end{smallmatrix}\right) \left(\begin{smallmatrix} 1 & a\omega\Delta t \\ 0 & 1 \end{smallmatrix}\right) \left(\begin{smallmatrix} 1 & 0 \\ -(\frac{1}{2}-b)\omega\Delta t & 1 \end{smallmatrix}\right) \left(\begin{smallmatrix} 1 & (1-2a)\omega\Delta t \\ 0 & 1 \end{smallmatrix}\right) \left(\begin{smallmatrix} 1 & 0 \\ -(\frac{1}{2}-b)\omega\Delta t & 1 \end{smallmatrix}\right) \left(\begin{smallmatrix} 1 & a\omega\Delta t \\ 0 & 1 \end{smallmatrix}\right) \left(\begin{smallmatrix} 1 & 0 \\ -b\omega\Delta t & 1 \end{smallmatrix}\right) \\ &= \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \end{aligned}$$

where

$$A = D = 1 - \frac{(\omega\Delta t)^2}{2} + a(1 - 4b^2 - a(2 - 4b))\frac{(\omega\Delta t)^4}{4} + a^2(2a - 1)(1 - 2b)^2 b\frac{(\omega\Delta t)^6}{4},$$

$$B = \omega\Delta t + a(1 - a)(2b - 1)(\omega\Delta t)^3 + a^2(1 - 2a)(1 - 2b)^2\frac{(\omega\Delta t)^5}{4},$$

$$C = -\omega\Delta t + (1 - 2a(1 - 2b)^2)\frac{(\omega\Delta t)^3}{4} + a(2a(1 - b) - 1)b(1 - 2b)\frac{(\omega\Delta t)^5}{2} + a^2(1 - 2a)(1 - 2b)^2 b^2\frac{(\omega\Delta t)^7}{4}.$$

A.2 Limitations on a time step in the GROMACS code

If one investigates the restrictions on a choice of a time step in the GROMACS package, the following statement can be found inside of the code:

> The stability limit of leapfrog or velocity verlet is 4.44 steps per oscillational period.

It does not seem consistent with the well known linear stability limit for the velocity Verlet integrator of $h < 2/\omega$, where ω is the angular frequency of the harmonic oscillator. However, it is consistent with the restriction due to non-linear instability studied (cf. (Skeel, Zhang, and Schlick, 1997))

$$h < \sqrt{2}/\omega. \tag{A.1}$$

It has been explained above that this condition allows avoiding some non-physical resonances that might be introduced by the symplectic integrator. More details can be found in Section 3.2.1.1.

Using equation (A.1) and the definition of ω for the harmonic oscillator it is easy to show:

$$\left.\begin{aligned} h < \sqrt{2}/\omega \Rightarrow \omega h < \sqrt{2} \\ \omega = \frac{2\pi}{T} \end{aligned}\right\} \Rightarrow \frac{2\pi}{T}h < \sqrt{2}, \tag{A.2}$$

where T denotes the period of the harmonic oscillator. Now, we can define h as

$$h := \frac{T}{n_{\text{steps}_T}}, \tag{A.3}$$

with n_{steps_T} being the number of steps performed per period. Substituting (A.3) in (A.2), one obtains

$$\frac{2\pi}{T}\frac{T}{n_{\text{steps}_T}} < \sqrt{2} \Rightarrow n_{\text{steps}_T} > \frac{2\pi}{\sqrt{2}} = \sqrt{2}\pi \approx 4.44.$$

Appendix B

Numerical Derivatives of Positions and Momenta

- $\dot{q} = M^{-1}p.$
- $\dot{p} = -U_q.$
- $\ddot{q} = M^{-1}\dot{p} = -M^{-1}U_q.$
- $\ddot{p} = -U_{qq}M^{-1}p.$
- $\dddot{q} = -M^{-1}U_{qq}M^{-1}p.$
- $\dddot{p} = -U_{qqq}M^{-1}pM^{-1}p + U_{qq}M^{-1}U_q.$[1]
- $q^{(4)} = -M^{-1}U_{qqq}M^{-1}pM^{-1}p + M^{-1}U_{qq}M^{-1}U_q.$
- $p^{(4)} = -U_{qqqq}M^{-1}pM^{-1}pM^{-1}p + 3U_{qqq}M^{-1}U_qM^{-1}p + U_{qq}M^{-1}U_{qq}M^{-1}p.$
- $q^{(5)} = -M^{-1}U_{qqqq}M^{-1}pM^{-1}pM^{-1}p + 3M^{-1}U_{qqq}M^{-1}U_qM^{-1}p + M^{-1}U_{qq}M^{-1}U_{qq}M^{-1}p.$

[1] The higher order derivatives are needed for the shadow Hamiltonians of order higher than fourth.

Appendix C

Lennard Jones Simulations

C.1 Reduced units in the Lennard-Jones simulations

The parameters ϵ and σ in (5.27) are used to define the reduced units traditionally used in molecular simulations (Allen and Tildesley, 1989). We choose the mass of the particles m to be the unit of mass, the hard-core radius of the potential energy function σ to be the unit of length and the depth of the potential energy function ϵ to be the unit of energy. Thus, the following reduced units are defined:

- temperature $T^* = \frac{k_B T}{\epsilon} = \frac{1}{\beta\epsilon}$,
- volume $V^* = \frac{V}{\sigma^3}$,
- density $\rho^* = \rho\sigma^3$,
- pressure $P^* = \frac{P\sigma^3}{\epsilon}$,
- time $t^* = \sqrt{\frac{\epsilon}{m\sigma^2}}t$,
- force $F^* = \frac{F\sigma}{\epsilon}$,
- potential energy $U^* = \frac{U}{N\epsilon}$ (Yao, Greenkorn, and Chao, 1982).

In our context, it is more interesting to have in reduced units the Planck's constant h rather than the thermal de Broglie wavelength as in (Yao, Greenkorn, and Chao, 1982; Rowley, Nicholson, and Parsonage, 1975) (cf. equations (5.21)-(5.22)). Thus, since the Planck's constant has units of Julius times seconds, it has to be normalized with the reduced time and the reduced energy[1]. Then,

$$h^* = \frac{h}{\epsilon\sigma\sqrt{m/\epsilon}} = \frac{h}{\sigma\sqrt{m\epsilon}}.$$

Then, from this definition one can get

$$\Lambda = \frac{h}{\sqrt{2\pi m k_B T}} = \frac{h^*\sigma\sqrt{m\epsilon}}{\sqrt{2\pi m T^*\epsilon}} = \frac{h^*\sigma}{\sqrt{2\pi T^*}} \Rightarrow \Lambda^* = \frac{\Lambda}{\sigma}.$$

The chemical potential, since it has energy units, in reduced units is defined as (cf. (Yao, Greenkorn, and Chao, 1982))

$$\mu^* = \frac{\mu}{\epsilon}.$$

[1] See (Mohazzabi and Mansoori, 2005), where it is done for the concrete case of argon. The procedure works in general.

Thus, it is easy to see that the quantity $\beta\mu$ that appears several times (cf. equations (5.11) or (5.21)-(5.22)) can be easily made dimensionless:

$$\beta\mu = \beta\mu^*\epsilon = \frac{\mu^*}{T^*}.$$

C.2 Computation of the forces and the virial with a Lennard Jones potential

We first introduce the following notation:

$$\mathbf{q}_{ij} = q_i - q_j, \qquad q_{ij} = |\mathbf{q}_{ij}|.$$

In the absence of external forces, the potential can be represented in the simplest case as a sum of pairwise interactions:

$$U = \sum_{i=1}^{N} \sum_{j>i}^{N} u(q_{ij}).$$

It is clear that the condition $j > i$ prevents the double counting of the particle pairs. The forces acting on the particles are composed in such a case of the individual interactions with the rest of the particles

$$F_i = \sum_{j \neq i}^{N} f_{ij},$$

where

$$f_{ij} = -\frac{\mathrm{d}u(q_{ij})}{\mathrm{d}q_{ij}} \cdot \frac{\mathbf{q}_{ij}}{q_{ij}}.$$

Thus, with the notation presented above, the potential energy in (5.27) can be written as

$$U(r) = 4\epsilon \left[\left(\frac{\sigma}{q_{ij}} \right)^{12} - \left(\frac{\sigma}{q_{ij}} \right)^{6} \right].$$

The inter-particle forces arising from the Lennard Jones potential above have the form

$$f_{ij} = \frac{48\epsilon}{r_{ij}^2} \left[\left(\frac{\sigma}{q_{ij}} \right)^{12} - \frac{1}{2} \left(\frac{\sigma}{q_{ij}} \right)^{6} \right] \mathbf{q}_{ij}.$$

The virial is defined as (cf. (Goldstein, 1980))

$$\mathrm{vir} = \frac{1}{3} \sum_{j>i} f_{ij} \cdot \mathbf{q}_{ij}.$$

Thus, in the case of a Lennard Jones potential. it can be computed as

$$\mathrm{vir} = \frac{1}{3} \sum_{j>i} \left\{ 48\epsilon \left[\left(\frac{\sigma}{q_{ij}} \right)^{12} - \frac{1}{2} \left(\frac{\sigma}{q_{ij}} \right)^{6} \right] \right\}.$$

Bibliography

Abraham, M. J., T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, and E. Lindahl (2015). "GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers". In: *SoftwareX* 1-2, pp. 19–25. ISSN: 2352-7110. URL: http://www.sciencedirect.com/science/article/pii/S2352711015000059.

Adams, D. J. (1974). "Chemical potential of hard-sphere fluids by Monte Carlo methods". In: *Molecular Physics* 28.5, pp. 1241–1252. URL: http://dx.doi.org/10.1080/00268977400102551.

— (1975). "Grand canonical ensemble Monte Carlo for a Lennard-Jones fluid". In: *Molecular Physics* 29.1, pp. 307–311. URL: http://dx.doi.org/10.1080/00268977500100221.

— (1979). "Calculating the high-temperature vapour line by Monte Carlo". In: *Molecular Physics* 37.1, pp. 211–221. URL: http://dx.doi.org/10.1080/00268977900100171.

Akhmatskaya, E., N. Bou-Rabee, and S. Reich (2009). "A comparison of generalized hybrid Monte Carlo methods with and without momentum flip". In: *Journal of Computational Physics* 228.6, pp. 2256–2265. ISSN: 0021-9991. URL: http://www.sciencedirect.com/science/article/pii/S0021999108006323.

Akhmatskaya, E., R. Nobes, and S. Reich (2011). "Method, apparatus and computer program for molecular simulation". US patent (granted).

Akhmatskaya, E. and S. Reich (2006). "The Targeted Shadowing Hybrid Monte Carlo (TSHMC) Method". In: *New Algorithms for Macromolecular Simulation*. Ed. by Benedict Leimkuhler, Christophe Chipot, Ron Elber, Aatto Laaksonen, Alan Mark, Tamar Schlick, Christoph Schütte, and Robert Skeel. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 141–153. ISBN: 978-3-540-31618-3. DOI: 10.1007/3-540-31618-3_9. URL: http://dx.doi.org/10.1007/3-540-31618-3_9.

— (2008). "GSHMC: An efficient method for molecular simulation". In: *Journal of Computational Physics* 227.10, pp. 4934–4954. ISSN: 0021-9991. URL: http://www.sciencedirect.com/science/article/pii/S0021999108000533.

— (2011a). "Meso-GSHMC: A stochastic algorithm for meso-scale constant temperature simulations". In: *Procedia Computer Science* 4, pp. 1353–1362. ISSN: 1877-0509. URL: http://www.sciencedirect.com/science/article/pii/S1877050911002043.

— (2011b). "New Hybrid Monte Carlo Methods for Efficient Sampling : from Physics to Biology and Statistics (Selected Papers of the Joint International Conference of Supercomputing in Nuclear Applications and Monte Carlo : SNA + MC 2010)". In: *Progress in nuclear science and technology* 2, pp. 447–462. ISSN: 2185-4823. URL: http://ci.nii.ac.jp/naid/40019316083/en/.

Akhmatskaya, E., S. Reich, and R. Nobes (2009). "Method, apparatus and computer program for molecular simulation". GB patent (published).

Akhmatskaya, E., T. van Mourik, H. Früchtl, A. Heidenreich, K. Rademann, F. Emmerling, and E. Rössler (2013). "Computational study of polymorphism in drugs". In: *HPC-Europa Annual Report Book*, pp. 994–997.

Akhmatskaya, E., M. Fernández-Pendás, T. Radivojević, and J. M. Sanz-Serna (2017). "Adaptive splitting integrators for enhancing sampling efficiency of modified Hamiltonian Monte Carlo methods in molecular simulation". In: *Langmuir* 33.42, pp. 11530–11542. URL: https://doi.org/10.1021/acs.langmuir.7b01372.

Alder, B. J. and T. E. Wainwright (1959). "Studies in Molecular Dynamics. I. General Method". In: *The Journal of Chemical Physics* 31.2, pp. 459–466. URL: http://dx.doi.org/10.1063/1.1730376.

Alexander, F. J., G. L. Eyink, and J. M. Restrepo (2005). "Accelerated Monte Carlo for Optimal Estimation of Time Series". In: *Journal of Statistical Physics* 119.5, pp. 1331–1345. ISSN: 1572-9613. URL: https://doi.org/10.1007/s10955-005-3770-1.

Allen, M. P. and D. J. Tildesley (1989). *Computer Simulation of Liquids*. New York, NY, USA: Clarendon Press. ISBN: 0-19-855645-4.

Andersen, H. C. (1980). "Molecular dynamics simulations at constant pressure and/or temperature". In: *The Journal of Chemical Physics* 72.4, pp. 2384–2393. URL: http://dx.doi.org/10.1063/1.439486.

— (1983). "Rattle: A "velocity" version of the shake algorithm for molecular dynamics calculations". In: *Journal of Computational Physics* 52.1, pp. 24–34. ISSN: 0021-9991. URL: http://www.sciencedirect.com/science/article/pii/0021999183900141.

Arnold, V.I. (1989). *Mathematical methods of classical mechanics*. Vol. 60. Springer.

Barker, J. A. and D. Henderson (1976). "What is "liquid"? Understanding the states of matter". In: *Reviews of Modern Physics* 48 (4), pp. 587–671. URL: https://link.aps.org/doi/10.1103/RevModPhys.48.587.

Bazari, W. L., P. Matsudaira, M. Wallek, T. Smeal, R. Jakes, and Y. Ahmed (1988). "Villin sequence and peptide map identify six homologous domains". In: *Proceedings of the National Academy of Sciences* 85.14, pp. 4986–4990. URL: http://www.pnas.org/content/85/14/4986.abstract.

Berendsen, H. J. C., D. van der Spoel, and R. van Drunen (1995). "GROMACS: A message-passing parallel molecular dynamics implementation". In: *Computer Physics Communications* 91.1, pp. 43–56. URL: http://www.sciencedirect.com/science/article/pii/001046559500042E.

Berendsen, H. J. C., J. P. M. Postma, W. F. van Gunsteren, and J. Hermans (1981). "Interaction Models for Water in Relation to Protein Hydration". In: *Intermolecular Forces: Proceedings of the Fourteenth Jerusalem Symposium on Quantum Chemistry and Biochemistry Held in Jerusalem, Israel, April 13–16, 1981*. Ed. by Bernard Pullman. Dordrecht: Springer Netherlands, pp. 331–342. ISBN: 978-94-015-7658-1. URL: http://dx.doi.org/10.1007/978-94-015-7658-1_21.

Berendsen, H. J. C., J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak (1984). "Molecular dynamics with coupling to an external bath". In: *The Journal of Chemical Physics* 81.8, pp. 3684–3690. URL: http://dx.doi.org/10.1063/1.448118.

Beskos, A., N. Pillai, G. Roberts, J. M. Sanz-Serna, and A. Stuart (2013). "Optimal tuning of the hybrid Monte Carlo algorithm". In: *Bernoulli* 19.5A, pp. 1501–1534. URL: http://dx.doi.org/10.3150/12-BEJ414.

Betancourt, M. (2017). "The Convergence of Markov chain Monte Carlo Methods: From the Metropolis method to Hamiltonian Monte Carlo". In: *ArXiv e-prints*. arXiv: 1706.01520.

Betancourt, M. J., S. Byrne, and M. Girolami (2014). "Optimizing The Integrator Step Size for Hamiltonian Monte Carlo". In: *ArXiv e-prints*. arXiv: 1411.6669.

Blanes, S., F. Casas, and A. Murua (2008). "Splitting and composition methods in the numerical integration of differential equations". In: *Boletín Sociedad Española de Matemática Aplicada* 45, pp. 87–143. URL: http://cds.cern.ch/record/1143448.

Blanes, S., F. Casas, and J. M. Sanz-Serna (2014). "Numerical Integrators for the Hybrid Monte Carlo Method". In: *SIAM Journal of Scientific Computing* 36.4, A1556–A1580. URL: http://dx.doi.org/10.1137/130932740.

Board, J. A., L. V. Kale, K. Schulten, R. D. Skeel, and T. Schlick (1994). "Modeling biomolecules: larger scales, longer durations". In: *IEEE Computational Science and Engineering* 1.4, pp. 19–30. ISSN: 1070-9924. URL: http://ieeexplore.ieee.org/document/338771/.

Boinepalli, S. and P. Attard (2003). "Grand canonical molecular dynamics". In: *The Journal of Chemical Physics* 119.24, pp. 12769–12775. URL: http://dx.doi.org/10.1063/1.1629079.

Bou-Rabee, N. and J. M. Sanz-Serna (2017a). "Geometric integrators and the Hamiltonian Monte Carlo method". In: *ArXiv e-prints*. arXiv: 1711.05337.

— (2017b). "Randomized Hamiltonian Monte Carlo". In: *The Annals of Applied Probability* 27.4, pp. 2159–2194. URL: https://doi.org/10.1214/16-AAP1255.

Bowers, K. J., E. Chow, H. Xu, R. O. Dror, M. P. Eastwood, B. A. Gregersen, J. L. Klepeis, I. Kolossvary, M. A. Moraes, F. D. Sacerdoti, J. K. Salmon, Y. Shan, and D. E. Shaw (2006). "Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters". In: *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing, Tampa, Florida*. SC '06. New York, USA: ACM. ISBN: 0-7695-2700-0. URL: http://doi.acm.org/10.1145/1188455.1188544.

Brooks, B. R., C. L. Brooks, A. D. Mackerell, L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, and M. Karplus (2009). "CHARMM: The biomolecular simulation program". In: *Journal of Computational Chemistry* 30.10, pp. 1545–1614. ISSN: 1096-987X. URL: http://dx.doi.org/10.1002/jcc.21287.

Bussi, G., D. Donadio, and M. Parrinello (2007). "Canonical sampling through velocity rescaling". In: *Journal of Chemical Physics* 126.1, p. 014101. URL: http://dx.doi.org/10.1063/1.2408420.

Bussi, G. and M. Parrinello (2007). "Accurate sampling using Langevin dynamics". In: *Physical Review E* 75 (5), p. 056707. URL: https://link.aps.org/doi/10.1103/PhysRevE.75.056707.

Campos, C. M. and J. M. Sanz-Serna (2017). "Palindromic 3-stage splitting integrators, a roadmap". In: *Journal of Computational Physics* 346, pp. 340–355. ISSN: 0021-9991. URL: http://www.sciencedirect.com/science/article/pii/S0021999117304515.

Cancès, E., F. Legoll, and G. Stoltz (2007). "Theoretical and numerical comparison of some sampling methods for molecular dynamics". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 41.2, pp. 351–389.

Cağin, T. and B. M. Pettitt (1991). "Molecular dynamics with a variable number of molecules". In: *Molecular Physics* 72.1, pp. 169–175. URL: http://dx.doi.org/10.1080/00268979100100111.

Chao, W. L., J. Solomon, D. L. Michels, and F. Sha (2015). "Exponential Integration for Hamiltonian Monte Carlo". In: *International Conference on Machine Learning – ICML 2015*, pp. 1142–1151.

Chen, L., Z. Qin, and J. S. Liu (2000). "Exploring Hybrid Monte Carlo in Bayesian Computation". In: p. 2000.

Creutz, M. (1988). "Global Monte Carlo algorithms for many-fermion systems". In: *Physical Review D* 38 (4), pp. 1228–1238. URL: https://link.aps.org/doi/10.1103/PhysRevD.38.1228.

Creutz, M. and A. Gocksch (1989). "Higher-order hybrid Monte Carlo algorithms". In: *Physical Review Letters* 63 (1), pp. 9–12. URL: https://link.aps.org/doi/10.1103/PhysRevLett.63.9.

Darden, T., D. York, and L. Pedersen (1993). "Particle mesh Ewald: An $N \cdot \log(N)$ method for Ewald sums in large systems". In: *The Journal of Chemical Physics* 98.12, pp. 10089–10092. URL: https://doi.org/10.1063/1.464397.

Dauxois, T. (2008). "Fermi, Pasta, Ulam, and a mysterious lady". In: *Physics Today* 6.1, pp. 55–57. URL: http://physicstoday.scitation.org/doi/10.1063/1.2835154.

De Raedt, H. and B. De Raedt (1983). "Applications of the generalized Trotter formula". In: *Physiscal Review A* 28 (6), pp. 3575–3580. URL: https://link.aps.org/doi/10.1103/PhysRevA.28.3575.

Deitrick, G. L., L. E. Scriven, and H. T. Davis (1989). "Efficient molecular simulation of chemical potentials". In: *The Journal of Chemical Physics* 90.4, pp. 2370–2385. URL: http://dx.doi.org/10.1063/1.455979.

Duane, S., A .D. Kennedy, B. J. Pendleton, and D. Roweth (1987). "Hybrid Monte Carlo". In: *Physics Letters B* 195.2, pp. 216–222. URL: http://www.sciencedirect.com/science/article/pii/037026938791197X.

Escribano, B., E. Akhmatskaya, and J. I. Mujika (2013). "Combining stochastic and deterministic approaches within high efficiency molecular simulations". In: *Central European Journal of Mathematics* 11.4, pp. 787–799. ISSN: 1644-3616. URL: http://dx.doi.org/10.2478/s11533-012-0164-x.

Escribano, B., E. Akhmatskaya, S. Reich, and J. M. Azpiroz (2015). "Multiple-time-stepping generalized hybrid Monte Carlo methods". In: *Journal of Computational Physics* 280, pp. 1–20. ISSN: 0021-9991. URL: http://www.sciencedirect.com/science/article/pii/S0021999114006433.

Escribano, B., A. Lozano, T. Radivojević, M. Fernández-Pendás, J. Carrasco, and E. Akhmatskaya (2017). "Enhancing sampling in atomistic simulations of solid-state materials for batteries: a focus on olivine NaFePO4". In: *Theoretical Chemistry Accounts* 136.4, p. 43. ISSN: 1432-2234. URL: http://dx.doi.org/10.1007/s00214-017-2064-4.

Essmann, U., L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen (1995). "A smooth particle mesh Ewald method". In: *The Journal of Chemical Physics* 103.19, pp. 8577–8593. URL: https://doi.org/10.1063/1.470117.

Faller, R. and J. J. de Pablo (2002). "Constant pressure hybrid Molecular Dynamics-Monte Carlo simulations". In: *The Journal of Chemical Physics* 116.1, pp. 55–59. URL: http://aip.scitation.org/doi/abs/10.1063/1.1420460.

Fang, Y., J. M. Sanz-Serna, and R. D. Skeel (2014). "Compressible generalized hybrid Monte Carlo". In: *The Journal of Chemical Physics* 140.17, p. 174108. URL: http://dx.doi.org/10.1063/1.4874000.

Fermi, E., J. Pasta, and S. Ulam (1955). "Studies of nonlinear problems". In: *Technical Report LA-1940*. Los Alamos National Laboratory.

Fernández-Pendás, M., E. Akhmatskaya, and J. M. Sanz-Serna (2016). "Adaptive multi-stage integrators for optimal energy conservation in molecular simulations". In: *Journal of Computational Physics* 327, pp. 434–449. ISSN: 0021-9991. URL: http://www.sciencedirect.com/science/article/pii/S0021999116304569.

Fernández-Pendás, M., B. Escribano, T. Radivojević, and E. Akhmatskaya (2014). "Constant pressure hybrid Monte Carlo simulations in GROMACS". In: *Journal of Molecular Modelling* 20.12, p. 2487. ISSN: 0948-5023. URL: http://dx.doi.org/10.1007/s00894-014-2487-y.

Feynman, R. P., R. B. Leighton, and M. L. Sands (1964). *The Feynman Lectures on Physics. Volume I: Mainly mechanics, radiation, and heat.* The Feynman Lectures on Physics. Addison-Wesley. ISBN: 9780201021165.

Fornberg, B. (1988). "Generation of finite difference formulas on arbitrarily spaced grids". In: *Mathematics of Computation* 51, pp. 699–706. URL: https://doi.org/10.1090/S0025-5718-1988-0935077-0.

Geyer, C. J. (1992). "Practical Markov Chain Monte Carlo". In: *Statistical Science* 7.4, pp. 473–483. URL: http://www.jstor.org/stable/2246094.

Girolami, M. and B. Calderhead (2011). "Riemann manifold Langevin and Hamiltonian Monte Carlo methods". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.2, pp. 123–214. ISSN: 1467-9868. URL: http://dx.doi.org/10.1111/j.1467-9868.2010.00765.x.

Goldstein, H. (1980). *Classical Mechanics*. Addison-Wesley series in physics. Addison-Wesley Publishing Company. ISBN: 9780201029185.

Gupta, R., G. W. Kilcup, and S. R. Sharpe (1988). "Tuning the hybrid Monte Carlo algorithm". In: *Physiscal Review D* 38 (4), pp. 1278–1287. URL: https://link.aps.org/doi/10.1103/PhysRevD.38.1278.

Gupta, S., A. Irbäc, F. Karsch, and B. Petersson (1990). "The acceptance probability in the hybrid Monte Carlo method". In: *Physics Letters B* 242.3, pp. 437–443. ISSN: 0370-2693. URL: http://www.sciencedirect.com/science/article/pii/037026939091790I.

Hairer, E., C. Lubich, and G. Wanner (2006). *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. 2nd ed. Dordrecht: Springer. URL: https://cds.cern.ch/record/1250576.

Hansmann, U. H. E., Y. Okamoto, and F. Eisenmenger (1996). "Molecular dynamics, Langevin and hydrid Monte Carlo simulations in a multicanonical ensemble". In: *Chemical Physics Letters* 259.3, pp. 321–330. ISSN: 0009-2614. URL: http://www.sciencedirect.com/science/article/pii/0009261496007610.

Hartmann, C. (2008). "An Ergodic Sampling Scheme for Constrained Hamiltonian Systems with Applications to Molecular Dynamics". In: *Journal of Statistical Physics* 130.4, pp. 687–711. ISSN: 1572-9613. URL: http://dx.doi.org/10.1007/s10955-007-9470-2.

Hasenbusch, M. (2001). "Speeding up the hybrid Monte Carlo algorithm for dynamical fermions". In: *Physics Letters B* 519.1, pp. 177–182. ISSN: 0370-2693. URL: http://www.sciencedirect.com/science/article/pii/S0370269301011029.

Hastings, W. K. (1970). "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1, pp. 97–109. URL: http://dx.doi.org/10.1093/biomet/57.1.97.

Heermann, D. W., P. Nielaba, and M. Rovere (1990). "Hybrid molecular dynamics". In: *Computer Physics Communications* 60.3, pp. 311–318. URL: http://www.sciencedirect.com/science/article/pii/0010465590900305.

Hess, B., H. Bekker, H. J. C. Berendsen, and J. G. E. M. Fraaije (1997). "LINCS: A linear constraint solver for molecular simulations". In: *Journal of Computational Chemistry* 18.12, pp. 1463–1472. URL: http://dx.doi.org/10.1002/(SICI)1096-987X(199709)18:12<1463::AID-JCC4>3.0.CO;2-H.

Hess, B., C. Kutzner, D. van der Spoel, and E. Lindahl (2008). "GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation". In: *Journal of Chemical Theory and Computation* 4.3, pp. 435–447. URL: http://dx.doi.org/10.1021/ct700301q.

Hill, T. L. (1956). *Statistical Mechanics: Principles and selected applications*. Dover Publications. ISBN: 0486653900.

— (1960). *An Introduction to Statistical Thermodynamics*. Dover Publications. ISBN: 0486652424.

Hoover, W. G. (1985). "Canonical dynamics: Equilibrium phase-space distributions". In: *Physical Review A* 31 (3), pp. 1695–1697. URL: https://link.aps.org/doi/10.1103/PhysRevA.31.1695.

Horowitz, A. M. (1991). "A generalized guided Monte Carlo algorithm". In: *Physics Letters B* 268.2, pp. 247–252. URL: http://www.sciencedirect.com/science/article/pii/0370269391908125.

Izaguirre, J. A. and S. S. Hampton (2004). "Shadow hybrid Monte Carlo: an efficient propagator in phase space of macromolecules". In: *Journal of Computational Physics* 200.2, pp. 581–604. URL: http://www.sciencedirect.com/science/article/pii/S0021999104001809.

Jeltsch, R. and O. Nevanlinna (1981). "Stability of explicit time discretizations for solving initial value problems". In: *Numerische Mathematik* 37.1, pp. 61–91. ISSN: 0945-3245. URL: http://dx.doi.org/10.1007/BF01396187.

Johnson, J. K., J. A. Zollweg, and K. E. Gubbins (1993). "The Lennard-Jones equation of state revisited". In: *Molecular Physics* 78.3, pp. 591–618. URL: https://doi.org/10.1080/00268979300100411.

Joó, B., B. Pendleton, A. D. Kennedy, A. C. Irving, J. C. Sexton, S. M. Pickles, and S. P. Booth (2000). "Instability in the molecular dynamics step of a hybrid Monte Carlo algorithm in dynamical fermion lattice QCD simulations". In: *Physical Review D* 62 (11), p. 114501. URL: https://link.aps.org/doi/10.1103/PhysRevD.62.114501.

Jost, G., H. Jin, D. an Mey, and F. Hatay (2003). "Comparing the OpenMP, MPI, and Hybrid Programming Paradigms on an SMP Cluster". In: *Fifth European Workshop on OpenMP (EWOMP03), in Aachen, Germany*. Vol. 3. URL: https://ntrs.nasa.gov/search.jsp?R=20030107321.

Jung, H. J., J. Y. Lee, S. H. Kim, Y. J. Eu, S. Y. Shin, M. Milescu, K. J. Swartz, and J. I. Kim (2005). "Solution Structure and Lipid Membrane Partitioning of VSTx1, an Inhibitor of the KvAP Potassium Channel," in: *Biochemistry* 44.16, pp. 6015–6023. URL: http://dx.doi.org/10.1021/bi0477034.

Kennedy, A. D. and B. Pendleton (2001). "Cost of the generalised hybrid Monte Carlo algorithm for free field theory". In: *Nuclear Physics B* 607.3, pp. 456–510. URL: http://www.sciencedirect.com/science/article/pii/S0550321301001298.

Klepeis, J. L., K. Lindorff-Larsen, R. O. Dror, and D. E. Shaw (2009). "Long-timescale molecular dynamics simulations of protein structure and function". In: *Current Opinion in Structural Biology* 19.2. Theory and simulation / Macromolecular assemblages, pp. 120–127. ISSN: 0959-440X. URL: http://www.sciencedirect.com/science/article/pii/S0959440X09000372.

Kolb, A. and B. Dünweg (1999). "Optimized constant pressure stochastic dynamics". In: *The Journal of Chemical Physics* 111.10, pp. 4453–4459. URL: http://dx.doi.org/10.1063/1.479208.

Kroese, D. P., T. Brereton, T. Taimre, and Z. I. Botev (2014). "Why the Monte Carlo method is so important today". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 6.6, pp. 386–392. ISSN: 1939-0068. URL: http://dx.doi.org/10.1002/wics.1314.

Leimkuhler, B., D. T. Margul, and M. E. Tuckerman (2013). "Stochastic, resonance-free multiple time-step algorithm for molecular dynamics with very large time steps". In: *Molecular Physics* 111.22-23, pp. 3579–3594. URL: http://dx.doi.org/10.1080/00268976.2013.844369.

Leimkuhler, B. J. and S. Reich (2004). *Simulating Hamiltonian dynamics*. Cambridge monographs on applied and computational mathematics. Cambridge: Cambridge University. URL: https://cds.cern.ch/record/835066.

— (2009). "A Metropolis adjusted Nosé-Hoover thermostat". In: *ESAIM: M2AN* 43.4, pp. 743–755. URL: https://doi.org/10.1051/m2an/2009023.

Leimkuhler, B. J. and R. D. Skeel (1994). "Symplectic Numerical Integrators in Constrained Hamiltonian Systems". In: *Journal of Computational Physics* 112.1, pp. 117–125. ISSN: 0021-9991. URL: http://www.sciencedirect.com/science/article/pii/S0021999184710850.

Lin, S.-T., M. Blanco, and W. A. Goddard III (2003). "The two-phase model for calculating thermodynamic properties of liquids from molecular dynamics: Validation for the phase diagram of Lennard-Jones fluids". In: *The Journal of Chemical Physics* 119.22, pp. 11792–11805. URL: https://doi.org/10.1063/1.1624057.

Lippert, R. A., K. J. Bowers, R. O. Dror, M. P. Eastwood, B. A. Gregersen, J. L. Klepeis, I. Kolossvary, and D. E. Shaw (2007). "A common, avoidable source of error in molecular dynamics integrators". In: *The Journal of Chemical Physics* 126.4, p. 046101. URL: http://dx.doi.org/10.1063/1.2431176.

Lo, C. and B. Palmer (1995). "Alternative Hamiltonian for molecular dynamics simulations in the grand canonical ensemble". In: *The Journal of Chemical Physics* 102.2, pp. 925–931. URL: http://dx.doi.org/10.1063/1.469159.

Mandziuk, M. and T. Schlick (1995). "Resonance in the dynamics of chemical systems simulated by the implicit midpoint scheme". In: *Chemical Physics Letters* 237.5, pp. 525–535. ISSN: 0009-2614. URL: http://www.sciencedirect.com/science/article/pii/000926149500316V.

Marrink, S. J., H. J. Risselada, S. Yefimov, D. P. Tieleman, and A. H. de Vries (2007). "The MARTINI Force Field: Coarse Grained Model for Biomolecular Simulations". In: *The Journal of Physical Chemistry B* 111.27, pp. 7812–7824. URL: https://doi.org/10.1021/jp071097f.

Martyna, G. J., M. L. Klein, and M. Tuckerman (1992). "Nosé-Hoover chains: The canonical ensemble via continuous dynamics". In: *The Journal of Chemical Physics* 97.4, pp. 2635–2643. URL: http://dx.doi.org/10.1063/1.463940.

Martyna, G. J., D. J. Tobias, and M. L. Klein (1994). "Constant pressure molecular dynamics algorithms". In: *The Journal of Chemical Physics* 101.5, pp. 4177–4189. URL: http://dx.doi.org/10.1063/1.467468.

Martyna, G. J., M. E. Tuckerman, D. J. Tobias, and M. L. Klein (1996). "Explicit reversible integrators for extended systems dynamics". In: *Molecular Physics* 87.5, pp. 1117–1157. URL: http://dx.doi.org/10.1080/00268979600100761.

Mazur, A. K. (1997). "Common Molecular Dynamics Algorithms Revisited: Accuracy and Optimal Time Steps of Störmer-Leapfrog Integrators". In: *Journal of Computational Physics* 136.2, pp. 354–365. URL: http://www.sciencedirect.com/science/article/pii/S0021999197957405.

McKnight, C. J., P. T. Matsudaira, and P. S. Kim (1997). "NMR structure of the 35-residue villin headpiece subdomain". In: *Nature Structural & Molecular Biology* 4.3, pp. 180–184. URL: http://dx.doi.org/10.1038/nsb0397-180.

McLachlan, R. I. (1995). "On the Numerical Integration of Ordinary Differential Equations by Symmetric Composition Methods". In: *SIAM Journal of Scientific Computing* 16.1, pp. 151–168. URL: http://dx.doi.org/10.1137/0916010.

McLachlan, R. I. and G. Reinout W. Quispel (2002). "Splitting methods". In: *Acta Numerica* 11, pp. 341–434. URL: https://doi.org/10.1017/S0962492902000053.

Mehlig, B., D. W. Heermann, and B. M. Forrest (1992). "Hybrid Monte Carlo method for condensed-matter systems". In: *Physical Review B* 45 (2), pp. 679–685. URL: https://link.aps.org/doi/10.1103/PhysRevB.45.679.

Metropolis, N. (1987). "The begining of the Monte Carlo method". In: *Los Alamos Science* 12, pp. 125–130.

Metropolis, N. and S. Ulam (1949). "The Monte Carlo Method". In: *Journal of the American Statistical Association* 44.247, pp. 335–341. URL: https://amstat.tandfonline.com/doi/abs/10.1080/01621459.1949.10483310.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). "Equation of State Calculations by Fast Computing Machines". In: *The Journal of Chemical Physics* 21.6, pp. 1087–1092. URL: http://link.aip.org/link/?JCP/21/1087/1.

Mezei, M. (1980). "A cavity-biased (T, V, μ) Monte Carlo method for the computer simulation of fluids". In: *Molecular Physics* 40.4, pp. 901–906. URL: http://dx.doi.org/10.1080/00268978000101971.

— (1987). "Grand-canonical ensemble Monte Carlo study of dense liquid". In: *Molecular Physics* 61.3, pp. 565–582. URL: http://dx.doi.org/10.1080/00268978700101321.

Mohamed, L., M. A. Christie, and V. Demyanov (2010). "Comparison of Stochastic Sampling Algorithms for Uncertainty Quantification". In: *Technical report, Institute of Petroleum Engineering. SPE Reservoir Simulation Symposium.* URL: https://www.onepetro.org/journal-paper/SPE-119139-PA.

Mohazzabi, P. and G. A. Mansoori (2005). "Nonextensivity and nonintensivity in nanosystems: A molecular dynamics simulation". In: *Journal of Computational and Theoretical Nanoscience* 2.1, pp. 138–147. URL: http://www.ingentaconnect.com/content/asp/jctn/2005/00000002/00000001/art00013.

Neal, R. M. (2011). "MCMC Using Hamiltonian Dynamics". In: *Handbook of Markov Chain Monte Carlo*. Vol. 2. Chapman & Hall / CRC Press, pp. 113–162.

Nelson, M. T., W. Humphrey, A. Gursoy, A. Dalke, L. V. Kalé, R. D. Skeel, and K. Schulten (1996). "NAMD: a Parallel, Object-Oriented Molecular Dynamics Program". In: *The International Journal of Supercomputer Applications and High Performance Computing* 10.4, pp. 251–268. URL: https://doi.org/10.1177/109434209601000401.

Nicholson, D. and N. G. Parsonage (1982). *Statistical Mechanics and Computer simulation of Adsorption*. Academic Press: London.

Nicolas, J. J., K. E. Gubbins, W. B. Streett, and D. J. Tildesley (1979). "Equation of state for the Lennard-Jones fluid". In: *Molecular Physics* 37.5, pp. 1429–1454. URL: https://doi.org/10.1080/00268977900101051.

Norman, G. E. and V. S. Filinov (1969). "Investigations of phase transitions by a Monte-Carlo method". In: *High Temperature (USSR)* 7, pp. 216–222.

Nosé, S. (1984a). "A molecular dynamics method for simulations in the canonical ensemble". In: *Molecular Physics* 52.2, pp. 255–268. URL: http://dx.doi.org/10.1080/00268978400101201.

— (1984b). "A unified formulation of the constant temperature molecular dynamics methods". In: *The Journal of Chemical Physics* 81.1, pp. 511–519. URL: http://dx.doi.org/10.1063/1.447334.

Oh, S. and Y. Hori (2006). "Development of Golden Section Search Driven Particle Swarm Optimization and its Application". In: *2006 SICE-ICASE International Joint Conference*, pp. 2868–2873.

Okunbor, D. I. and R. D. Skeel (1994). "Canonical numerical methods for molecular dynamics simulations". In: *Journal of Computational Chemistry* 15.1, pp. 72–79. ISSN: 1096-987X. URL: http://dx.doi.org/10.1002/jcc.540150109.

Páll, S, M. J. Abraham, C. Kutzner, B. Hess, and E. Lindahl (2015). "Tackling Exascale Software Challenges in Molecular Dynamics Simulations with GROMACS". In: *Solving Software Challenges for Exascale: International Conference on Exascale Applications and Software, EASC 2014, Stockholm, Sweden, April 2-3, 2014, Revised Selected Papers*. Ed. by S. Markidis and E. Laure. Cham: Springer International Publishing, pp. 3–27. ISBN: 978-3-319-15976-8. URL: http://dx.doi.org/10.1007/978-3-319-15976-8_1.

Parrinello, M. and A. Rahman (1981). "Polymorphic transitions in single crystals: A new molecular dynamics method". In: *Journal of Applied Physics* 52.12, pp. 7182–7190. URL: http://dx.doi.org/10.1063/1.328693.

Plimpton, S. (1995). "Fast Parallel Algorithms for Short-Range Molecular Dynamics". In: *Journal of Computational Physics* 117.1, pp. 1–19. ISSN: 0021-9991. URL: http://www.sciencedirect.com/science/article/pii/S002199918571039X.

Predescu, C., R. A. Lippert, M. P. Eastwood, D. Ierardi, H. Xu, M. Ø. Jensen, K. J. Bowers, J. Gullingsrud, C. A. Rendleman, R. O. Dror, and D. E. Shaw (2012). "Computationally efficient molecular dynamics integrators with improved sampling accuracy". In: *Molecular Physics* 110.9-10, pp. 967–983. URL: http://dx.doi.org/10.1080/00268976.2012.681311.

Pronk, S., S. Páll, R. Schulz, P. Larsson, P. Bjelkmar, R. Apostolov, M. R. Shirts, J. C. Smith, P. M. Kasson, D. van der Spoel, B. Hess, and E. Lindahl (2013). "GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit". In: *Bioinformatics* 29.7, p. 845. URL: http://dx.doi.org/10.1093/bioinformatics/btt055.

Radivojević, T. (2016). "Enhancing Sampling in Computational Statistics Using Modified Hamiltonians". PhD thesis. Bilbao: University of the Basque Country (UPV/EHU).

Radivojević, T. and E. Akhmatskaya (2017). "Mix & Match Hamiltonian Monte Carlo". In: *ArXiv e-print*. arXiv: 1706.04032.

Radivojević, T., M. Fernández-Pendás, J. M. Sanz-Serna, and E. Akhmatskaya (2018). "Multistage splitting integrators for sampling with modified Hamiltonian Monte Carlo methods". In: *submitted*.

Rahman, A. (1964). "Correlations in the Motion of Atoms in Liquid Argon". In: *Physical Review* 136 (2A), A405–A411. URL: https://link.aps.org/doi/10.1103/PhysRev.136.A405.

Ramachandran, G. N., C. Ramakrishnan, and V. Sasisekharan (1963). "Stereochemistry of polypeptide chain configurations". In: *Journal of Molecular Biology* 7.1, pp. 95–99.

Rowley, L. A., D. Nicholson, and N. G. Parsonage (1975). "Monte Carlo grand canonical ensemble calculation in a gas-liquid transition region for 12-6 Argon". In: *Journal of Computational Physics* 17.4, pp. 401–414. URL: http://www.sciencedirect.com/science/article/pii/002199917590042X.

Ruth, R. (1983). "A canonical integration technique". In: *IEEE Transactions in Nuclear Science* NS-30.4, pp. 2669–2671.

Ryckaert, J.-P., G. Ciccotti, and H. J. C. Berendsen (1977). "Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes". In: *Journal of Computational Physics* 23.3, pp. 327–341. ISSN: 0021-9991. URL: http://www.sciencedirect.com/science/article/pii/0021999177900985.

Salomon-Ferrer, R., D. A. Case, and Ross C. Walker (2013). "An overview of the Amber biomolecular simulation package". In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 3.2, pp. 198–210. ISSN: 1759-0884. URL: http://dx.doi.org/10.1002/wcms.1121.

Sanz-Serna, J. M. (1991). "Two topics in nonlinear stability". In: *Advances in Numerical Analysis Volume I: Nonlinear Partial Equations and Dynamical Systems*. Ed. by Will Light. Clarendon Press, pp. 147–174. ISBN: 978-0-198-53438-9.

— (1992). "Symplectic integrators for Hamiltonian problems: an overview". In: *Acta Numerica* 1, pp. 243–286.

Sanz-Serna, J. M and M. P Calvo (1994). *Numerical Hamiltonian problems*. 1st ed. Applied Mathematics and Mathematical Computation 7. London: Chapman & Hall. ISBN: 0412542900.

Sanz-Serna, J. M. and M. N. Spijker (1986). "Regions of stability, equivalence theorems and the Courant-Friedrichs-Lewy condition". In: *Numerische Mathematik* 49.2, pp. 319–329. ISSN: 0945-3245. URL: http://dx.doi.org/10.1007/BF01389633.

Schlick, T. (2002). *Molecular Modeling and Simulation: An Interdisciplinary Guide*. Secaucus, NJ, USA: Springer-Verlag New York, Inc. ISBN: 038795404X.

Schlick, T., M. Mandziuk, R. D. Skeel, and K. Srinivas (1998). "Nonlinear Resonance Artifacts in Molecular Dynamics Simulations". In: *Journal of Computational Physics* 140.1, pp. 1–29. URL: http://www.sciencedirect.com/science/article/pii/S002199919895879X.

Sexton, J. C. and D. H. Weingarten (1992). "Hamiltonian evolution for the hybrid Monte Carlo algorithm". In: *Nuclear Physics B* 380.3, pp. 665–677. ISSN: 0550-3213. URL: http://www.sciencedirect.com/science/article/pii/055032139290263B.

Shaw, D. E., M. M. Deneroff, R. O. Dror, J. S. Kuskin, R. H. Larson, J. K. Salmon, C. Young, B. Batson, K. J. Bowers, J. C. Chao, M. P. Eastwood, J. Gagliardo, J. P. Grossman, C. R. Ho, D. J. Ierardi, I. Kolossváry, J. L. Klepeis, T. Layman, C. McLeavey, M. A. Moraes, R. Mueller, E. C. Priest, Y. Shan, J. Spengler, M. Theobald, B. Towles, and S. C. Wang (2008). "Anton, a Special-purpose Machine for Molecular Dynamics Simulation". In: *Communications of the ACM* 51.7, pp. 91–97. ISSN: 0001-0782. URL: http://doi.acm.org/10.1145/1364782.1364802.

Shih, A. Y., A. Arkhipov, P. L. Freddolino, and K. Schulten (2006). "Coarse Grained Protein-Lipid Model with Application to Lipoprotein Particles". In: *Journal of Physical Chemistry B* 110.8, pp. 3674–3684. URL: http://dx.doi.org/10.1021/jp0550816.

Skeel, R. D. (1999). "Integration Schemes for Molecular Dynamics and Related Applications". In: *The Graduate Student's Guide to Numerical Analysis '98: Lecture Notes from the VIII EPSRC Summer School in Numerical Analysis*. Ed. by Mark Ainsworth, Jeremy Levesley, and Marco Marletta. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 119–176. ISBN: 978-3-662-03972-4. URL: http://dx.doi.org/10.1007/978-3-662-03972-4_4.

Skeel, R. D., Guihua Zhang, and Tamar Schlick (1997). "A Family of Symplectic Integrators: Stability, Accuracy, and Molecular Dynamics Applications". In: *SIAM Journal on Scientific Computing* 18.1, pp. 203–222. URL: https://doi.org/10.1137/S1064827595282350.

Smit, B. (1992). "Phase diagrams of Lennard-Jones fluids". In: *The Journal of Chemical Physics* 96.11, pp. 8639–8640. URL: https://doi.org/10.1063/1.462271.

Stern, H. A. (2007). "Molecular simulation with variable protonation states at constant pH". In: *The Journal of Chemical Physics* 126.16, p. 164112. URL: http://dx.doi.org/10.1063/1.2731781.

Straatsma, T. P., H. J. C. Berendsen, and A. J. Stam (1986). "Estimation of statistical errors in molecular simulation calculations". In: *Molecular Physics* 57.1, pp. 89–95. URL: http://dx.doi.org/10.1080/00268978600100071.

Susukita, R., T. Ebisuzaki, B. G. Elmegreen, H. Furusawa, K. Kato, A. Kawai, Y. Kobayashi, T. Koishi, G. D. McNiven, T. Narumi, and K. Yasuoka (2003). "Hardware accelerator for molecular dynamics: MDGRAPE-2". In: *Computer Physics Communications* 155.2, pp. 115–131. ISSN: 0010-4655. URL: http://www.sciencedirect.com/science/article/pii/S0010465503003497.

Sweet, C. R., S. S. Hampton, R. D. Skeel, and J. A. Izaguirre (2009). "A separable shadow Hamiltonian hybrid Monte Carlo method". In: *Journal of Chemical Physics* 131.17, p. 174106. URL: http://dx.doi.org/10.1063/1.3253687.

Swope, W. C., H. C. Andersen, P. H. Berens, and K. R. Wilson (1982). "A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters". In: *The Journal of Chemical Physics* 76.1, pp. 637–649. URL: http://dx.doi.org/10.1063/1.442716.

Takaishi, T. and P. de Forcrand (2006). "Testing and tuning symplectic integrators for the hybrid Monte Carlo algorithm in lattice QCD". In: *Physical Review E* 73 (3), p. 036706. URL: https://link.aps.org/doi/10.1103/PhysRevE.73.036706.

Trotter, H. F. (1959). "On the Product of Semi-Groups of Operators". In: *Proceedings of the American Mathematical Society* 10.4, pp. 545–551. URL: http://www.jstor.org/stable/2033649.

Tuckerman, M., B. J. Berne, and G. J. Martyna (1992). "Reversible multiple time scale molecular dynamics". In: *The Journal of Chemical Physics* 97.3, pp. 1990–2001. URL: http://dx.doi.org/10.1063/1.463137.

Tuckerman, M. E. (2010). *Statistical Mechanics: Theory and Molecular Simulation*. 1st ed. Oxford University Press. ISBN: 978-0-19-852526-4.

Tuckerman, M. E., B. J. Berne, and G. J. Martyna (1991). "Molecular dynamics algorithm for multiple time scales: Systems with long range forces". In: *The Journal of Chemical Physics* 94.10, pp. 6811–6815. URL: http://dx.doi.org/10.1063/1.460259.

Tuckerman, M. E., B. J. Berne, G. J. Martyna, and M. L. Klein (1993). "Efficient molecular dynamics and hybrid Monte Carlo algorithms for path integrals". In: *The Journal of Chemical Physics* 99.4, pp. 2796–2808. URL: http://dx.doi.org/10.1063/1.465188.

Tuckerman, M. E., J. Alejandre, R. López-Rendón, A. L. Jochim, and G. J. Martyna (2006). "A Liouville-operator derived measure-preserving integrator for molecular dynamics simulations in the isothermal–isobaric ensemble". In: *Journal of Physics A: Mathematical and General* 39.19, p. 5629. URL: http://stacks.iop.org/0305-4470/39/i=19/a=S18.

van der Spoel, D. and E. Lindahl (2003). "Brute-Force Molecular Dynamics Simulations of Villin Headpiece: Comparison with NMR Parameters". In: *Journal of Physical Chemistry B* 107.40, pp. 11178–11187. URL: http://dx.doi.org/10.1021/jp034108n.

van der Spoel, D. and P. J. van Maaren (2006). "The Origin of Layer Structure Artifacts in Simulations of Liquid Water". In: *Journal of Chemical Theory and Computation* 2.1, pp. 1–11. URL: http://dx.doi.org/10.1021/ct0502256.

Verlet, L. (1967). "Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules". In: *Physical Review* 159 (1), pp. 98–103. URL: https://link.aps.org/doi/10.1103/PhysRev.159.98.

Wagoner, J. A. and V. S. Pande (2012). "Reducing the effect of Metropolization on mixing times in molecular dynamics simulations". In: *The Journal of Chemical Physics* 137.21, p. 214105. URL: http://dx.doi.org/10.1063/1.4769301.

Wallace, E. J. and M. S. P. Sansom (2007). "Carbon Nanotube/Detergent Interactions via Coarse-Grained Molecular Dynamics". In: *Nano Letters* 7.7, pp. 1923–1928. URL: http://dx.doi.org/10.1021/nl070602h.

Wee, C. L., M. S. P. Sansom, S. Reich, and E. Akhmatskaya (2008). "Improved Sampling for Simulations of Interfacial Membrane Proteins: Application of Generalized Shadow Hybrid Monte Carlo to a Peptide Toxin/Bilayer System". In: *Journal of Physical Chemistry B* 112.18, pp. 5710–5717. URL: http://dx.doi.org/10.1021/jp076712u.

Wood, W. W. (1968). "Monte Carlo Calculations for Hard Disks in the Isothermal-Isobaric Ensemble". In: *The Journal of Chemical Physics* 48.1, pp. 415–434. URL: https://doi.org/10.1063/1.1667938.

Yao, J., R. A. Greenkorn, and K. C. Chao (1982). "Monte Carlo simulation of the grand canonical ensemble". In: *Molecular Physics* 46.3, pp. 587–594. URL: http://dx.doi.org/10.1080/00268978200101411.