Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 222 (2007) 428–440
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Trotter derived algorithms for molecular dynamics with constraints: Velocity Verlet revisited

V. Marry [a,*], G. Ciccotti [b]

[a] *Université Pierre et Marie Curie-Paris 6, UMR7612, LI2C, Case courrier 51, 4 Place Jussieu, F-75252 Paris Cedex 05, France*
[b] *Dipartimento di Fisica, Università "La Sapienza", P.le A.Moro, 2, 00185 Roma, Italy*

Received 14 February 2006; received in revised form 24 July 2006; accepted 26 July 2006
Available online 18 September 2006

Abstract

On the basis of the molecular dynamics algorithm proposed by Kalibaeva et al. [G. Kalibaeva, M. Ferrario, G. Ciccotti, Mol. Phys. 101 (2003) 765.] for systems with holonomic constraints in isobaric–isothermal ensemble, we discuss a new recursive algorithm which eliminates the inconsistency associated with the double calculation of constraint forces present in RATTLE. The algorithm is tested on bulk water and on a system containing a polymer with a large number of constraints to evaluate the CPU gain with respect to the usual RATTLE algorithm.

Keywords: Molecular dynamics; Algorithms; Holonomic constraints

1. Introduction

Molecular dynamics simulations, especially of large molecules like polymers or biological compounds, were made possible thanks to the introduction of constraints on atomic distances: this prevents strong chemical bonds, which require very stiff forces to be simulated properly, to ruin the efficiency of the simulation.

Let us denote by i, j two atoms linked by a constraint $\sigma = (\mathbf{r}_j - \mathbf{r}_i)^2 - d_{ij}^2 = 0$, where $\mathbf{r}_i$ and $\mathbf{r}_j$ are the positions of the atoms and d_{ij} the fixed distance between them. The resulting constraint force on atom i at t is given by $\mathbf{G}_i(t) = \lambda(t)\nabla_{r_i}\sigma(t)$, where $\lambda(t)$ is the Lagrange multiplier associated with the constraint $\sigma = 0$. Like the positions $\mathbf{r}_i$, Lagrange multipliers have to be determined at each time t. If there are k constraints and N is the number of atoms, we have $3N + k$ unknowns linked by $3N + k$ equations ($3N$ equations of motion and k constraints relationships), so that the unicity of the trajectory, starting from an initial configuration satisfying the constraints, is assured. It can be shown that the variables $\{\lambda\}$ can be theoretically calculated by recursively solving linear systems involving the λ's and their time derivatives [1]. However with this method, the constraints will be satisfied only up to the order of the chosen algorithm and the geometric characteristics of

---

[*] Corresponding author. Tel.: +33 1 44 27 22 03; fax: +33 1 44 27 38 34.
E-mail address: marry@ccr.jussieu.fr (V. Marry).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2006.07.033

the atomic structure bound by the constraints will be quickly lost. *SHAKE* was introduced to solve this problem [1,2]: it allows to satisfy rigorously all the constraints $\{\sigma_\alpha = 0\}_{\alpha=1,k}$, at each step of integration, without introducing extra errors beyond the algorithmic error. In the original paper, the algorithm SHAKE is associated with the classical Verlet algorithm, which does not need the velocities to compute the new positions: an iterative process allows to calculate the constraint forces at t (and Lagrange multipliers $\{\lambda_\alpha(t)\}$) in such a way that at each step the coordinates rigorously satisfy the constraints $\{\sigma_\alpha = 0\}_{\alpha=1,k}$ within a chosen accuracy [1,2]. In the velocity Verlet algorithm, instead, both coordinates and momenta are computed, so that the constraints $\{\sigma_\alpha = 0\}_{\alpha=1,k}$ and their time derivative $\{\dot{\sigma}_\alpha = 0\}_{\alpha=1,k}$ have to be imposed at each time step independently. That is why, in the velocity Verlet algorithm adapted to constraints, RATTLE [3], SHAKE is called once, after the implementation of atomic positions, to get the new positions at $t + \Delta t$ ($\{\lambda_\alpha(t)\}$); a second time, always at $t + \Delta t$, to evaluate the momenta at $t + \Delta t$ ($\{\lambda_\alpha(t + \Delta t)\}$); and then another time, at $t + 2\Delta t$, again in relation to the atomic positions, to compute again $\{\lambda_\alpha(t + \Delta t)\}_{\alpha=1,k}$. SHAKE is then used twice to calculate the λ's at the same time. To have an algorithm exact, as far as constraints are considered, the $\{\lambda_\alpha(t + \Delta t)\}_{\alpha=1,k}$ calculated at the end of the momentum step are not used for getting the positions at $t + 2\Delta t$ in the following step. The reason is that the constraints would not be exactly satisfied anymore, and, after a few steps, the atomic constrained structures would loose their geometrical characteristics. Having re-applied SHAKE, different values of λ's are computed and used in the algorithm. This inconsistency, although of minor numerical importance, requires also a double call to SHAKE (which is the signature of RATTLE) involving some waste of computer time.

Since velocity Verlet algorithm, contrary to the original Verlet algorithm, is compatible with the rigourous formalism introduced by Tuckerman et al. [4] to simulate non-hamiltonian systems (e.g. NVT, NPT Nosé style dynamics), it can be interesting to avoid the inconsistency of the double calculation of constraints, introduced by Andersen [3]. Recently Kalibaeva et al. introduced a correct treatment of constraints in the formalism proposed by Tuckerman et al., to produce correct and simple MD algorithms to simulate isothermal–isobaric ensembles in systems subject to holonomic constraints and with a molecular representation of the virial [5]. They showed that, by monitoring the parameter number of degrees of freedom in the Nosé dynamics, the use of a Nosé-Hoover chain in the equations of motion [6,7] is not necessary to retrieve the proper distribution function in NPT ensemble. In this algorithm, however, the constraints were still calculated twice, in accordance with RATTLE algorithm. We will show in this paper that the calculation of the constraint forces at time t, achieved after positions implementation, can be used to evaluate the momenta at $t - \Delta t$ without computing the Lagrange multipliers. We will so show that the inconsistency associated with the double calculation of constraints can be avoided. In Section 2, we propose the new version of Kalibaeva algorithm, which needs the evaluation of constraint forces only once in a step. In Section 3, we illustrate the implementation in bulk water, but we evaluate also the fraction of CPU time that can be saved when the system contains a molecule connected by a large number of constraints such as it happens for long polymers. Section 4 presents some concluding remarks.

2. Implementation of the dynamics in (NPT) ensemble

2.1. Equations of motion

The system is composed of n molecules containing n_μ atoms each and is submitted to $G = \sum_{\mu=1}^{n} k_\mu$ holonomic molecular constraints of type $\sigma_\mu^\alpha(r) = 0$ where $\alpha = 1, k_\mu$ and $\mu = 1, n$ with k_μ the number of constraints inside the molecule μ. The equations of motion we used to describe the dynamics of this system are almost the same as Kalibaeva et al. [5] but with center-of-mass positions and relative positions of atoms decoupled.

The equations of motion become

$$\dot{\mathbf{r}}'_{\mu i} = \frac{\mathbf{p}'_{\mu i}}{m_{\mu i}}$$

$$\dot{\mathbf{p}}'_{\mu i} = \mathbf{F}_{\mu i} + \mathbf{G}_{\mu i} - \frac{m_{\mu i}}{M_\mu}\mathbf{F}_\mu - \mathbf{p}'_{\mu i}\frac{P_S}{M_S}$$

$$\begin{aligned}
\dot{\mathbf{R}}_\mu &= \frac{\mathbf{P}_\mu}{M_\mu} + \mathbf{R}_\mu \frac{P_V}{3VM_V} \\
\dot{\mathbf{P}}_\mu &= \mathbf{F}_\mu - \mathbf{P}_\mu \left[\frac{P_S}{M_S} + \frac{P_V}{3VM_V}\right] \\
\dot{V} &= \frac{P_V}{M_V} \\
\dot{P}_V &= [\Pi - \Pi_{\mathrm{EXT}}] - P_V \frac{P_\xi}{M_\xi} \\
\dot{\eta} &= \frac{P_S}{M_S} \\
\dot{\xi} &= \frac{P_\xi}{M_\xi} \\
\dot{P}_S &= gk_{\mathrm{B}}[T - T_{\mathrm{EXT}}] \\
\dot{P}_\xi &= \frac{P_V^2}{M_V} - k_{\mathrm{B}}T_{\mathrm{EXT}}
\end{aligned} \tag{1}$$

where $\mathbf{R}_\mu$ and $\mathbf{P}_\mu$, with $\mu = 1,\ldots,n$, are respectively the center-of-mass positions and momenta of the n molecules, $\boldsymbol{r}'_{\mu i}$ and $\mathbf{p}'_{\mu i}$, with $i = 1,\ldots,n_\mu$, are the positions and momenta of the n_μ atoms of a molecule relative to those of the center of mass ($\mathbf{r}'_{\mu i} = \mathbf{r}_{\mu i} - \mathbf{R}_\mu$ and $\mathbf{p}'_{\mu i} = \mathbf{p}_{\mu i} - \frac{m_{\mu i}}{M_\mu}\mathbf{P}_\mu$ where $m_{\mu i}$ is the mass of the atom i of the molecule μ and M_μ the total mass of the molecule); $\mathbf{F}_{\mu i}$ are the forces on the atoms, $\mathbf{F}_\mu$ the total forces on the molecules and $\mathbf{G}_{\mu i}$ are the constraint forces. V and P_V are respectively the volume and the volume momentum of the system, associated with the inertial factor M_V; g is a parameter, to be put equal to $3N - G$; k_{B}, T_{EXT} and Π_{EXT} are respectively Boltzmann constant, external temperature and pressure. For the internal molecular pressure, we have:

$$\Pi = \frac{1}{3V}\sum_\mu \left[\frac{\mathbf{P}_\mu^2}{M_\mu} + \mathbf{F}_\mu.\mathbf{R}_\mu\right] - \frac{\partial \phi}{\partial V}$$

where ϕ is the potential, which may have an explicit volume dependence, and, for the internal temperature,

$$T = \frac{1}{gk_{\mathrm{B}}}\left[\sum_{i,\mu}\frac{\mathbf{p}'^2_{\mu i}}{m_{\mu i}} + \sum_\mu \frac{\mathbf{P}_\mu^2}{M_\mu}\right]$$

(ξ, P_ξ) and (η, P_S) are the variables corresponding to the thermostats for volume and particle momenta respectively. Their associated inertial factors are M_ξ and M_S.

The conserved quantity is

$$H' = H(p', r', R, P) + gk_{\mathrm{B}}T_{\mathrm{EXT}}\eta + k_{\mathrm{B}}T_{\mathrm{EXT}}\xi + \Pi_{\mathrm{EXT}}V + \frac{P_V^2}{2M_V} + \frac{P_S^2}{2M_S} + \frac{P_\xi^2}{2M_\xi} = C1$$

where $H(p',r',R,P)$ is the sum of the kinetic and potential energies of the original system.

It can be checked that the equations for $\mathbf{r}_{\mu i} = \mathbf{r}'_{\mu i} + \mathbf{R}_\mu$ and $\mathbf{p}_{\mu i} = \mathbf{p}'_{\mu i} + \frac{m_{\mu i}}{M_\mu}\mathbf{P}_\mu$ are exactly the equations of motion of Kalibaeva et al. However, if $N = \sum_{\mu=1}^{n} n_\mu$ is the total number of atoms of the system, we have here $6(N + n)$ equations for the atomic degrees of freedom. However, the variables are linked by $6n$ additional constraints: $\boldsymbol{\sigma}_\mu^{k_\mu+1} = \sum_{i=1}^{n_\mu} m_{\mu i}\mathbf{r}'_{\mu i} = 0$ and $\dot{\boldsymbol{\sigma}}_\mu^{k_\mu+1} = \sum_{i=1}^{n_\mu}\mathbf{p}'_{\mu i} = 0$. The constraint forces are given by

$$\mathbf{G}_{\mu i} = \sum_{\alpha=1}^{k_\mu+1} \lambda_\mu^\alpha \nabla_{\mathbf{r}_{\mu i}} \sigma_\mu^\alpha$$

where λ_μ^α are the Lagrange multipliers.

The system can be treated with the $6(N + n)$ equations and the $6n$ additional constraints, or $n_\mu - 1$ independent variables can be chosen for each molecule μ, leading to $6(N - n)$ equations for a subset of $\mathbf{r}'$ and $\mathbf{p}'$. In the latter case, all linearly dependent variables are eliminated.

It can be shown [5] that the partition function, deriving from these equations of motions and constraints, corresponds to the correct isothermal–isobaric distribution in the presence of constraints.

2.2. Explicit reversible integrator

Let us denote $\Gamma = [\mathbf{r}^N, \mathbf{p}^N, V, P_V, \eta, P_\eta, \xi, P_\xi]$ the vector of the $6(N+n)+6$ variables of the system. $\Gamma(t)$ at time t can be obtained from Γ by applying the evolution operator [8]:

$$\Gamma(t) = \exp(\mathrm{i}Lt)$$

To get a useful algorithm, the Liouville operator iL, associated with the dynamics of Eq. (1), can be decomposed as

$$\mathrm{i}L = \mathrm{i}L_1 + \mathrm{i}L_2 + \cdots + \mathrm{i}L_{13} + \mathrm{i}L_{14}$$

where:

$$\begin{aligned}
\mathrm{i}L_1 &= \frac{\mathbf{p}'_{\mu i}}{m_{\mu i}} \nabla_{\mathbf{r}'_{\mu i}} \\
\mathrm{i}L_2 &= \frac{\mathbf{P}_\mu}{M_\mu} \nabla_{\mathbf{R}_\mu} \\
\mathrm{i}L_3 &= \frac{P_V}{3VM_V} \mathbf{R}_\mu \nabla_{\mathbf{R}_\mu} \\
\mathrm{i}L_4 &= \frac{P_V}{M_V} \frac{\partial}{\partial V} \\
\mathrm{i}L_5 &= [\Pi(t) - \Pi_{\mathrm{EXT}}] \frac{\partial}{\partial P_V} \\
\mathrm{i}L_6 &= -P_V \frac{P_\xi}{M_\xi} \frac{\partial}{\partial P_V} \\
\mathrm{i}L_7 &= gk_{\mathrm{B}}[T(t) - T_{\mathrm{EXT}}] \frac{\partial}{\partial P_S} \\
\mathrm{i}L_8 &= \mathbf{F}_\mu \nabla_{\mathbf{P}_\mu} \\
\mathrm{i}L_9 &= -\left[\frac{P_S}{M_S} + \frac{P_V}{3VM_V}\right] \mathbf{P}_\mu \nabla_{\mathbf{P}_\mu} \\
\mathrm{i}L_{10} &= \left[\mathbf{F}_{\mu i} + \mathbf{G}_{\mu i} - \frac{m_{\mu i}}{M_\mu} \mathbf{F}_\mu\right] \nabla_{\mathbf{p}'_{\mu i}} \\
\mathrm{i}L_{11} &= -\frac{P_S}{M_S} \mathbf{p}'_{\mu i} \nabla_{\mathbf{p}'_{\mu i}} \\
\mathrm{i}L_{12} &= \left[\frac{P_V^2}{M_V} - k_{\mathrm{B}} T_{\mathrm{EXT}}\right] \frac{\partial}{\partial P_\xi} \\
\mathrm{i}L_{13} &= \frac{P_\xi}{M_\xi} \frac{\partial}{\partial \xi} \\
\mathrm{i}L_{14} &= \frac{P_S}{M_S} \frac{\partial}{\partial \eta}
\end{aligned}$$

The propagator for a time step Δt can be obtained by applying Trotter formula to this decomposition of L [9]

$$\begin{aligned}
\exp(\mathrm{i}L\Delta t) = {}& \exp(\mathrm{i}L_{14}\Delta t/2) \exp(\mathrm{i}L_{13}\Delta t/2) \times \cdots \times \exp(\mathrm{i}L_2\Delta t/2) \exp(\mathrm{i}L_1\Delta t) \exp(\mathrm{i}L_2\Delta t/2) \times \cdots \\
& \times \exp(\mathrm{i}L_{13}\Delta t/2) \exp(\mathrm{i}L_{14}\Delta t/2)
\end{aligned}$$

To get a recursive formulation, these operators are applied from the left [10]. The useful properties to consider for our transformations are:

$$\mathrm{e}^{C\frac{\partial}{\partial x}\Delta t} f(x) = f(x + C\Delta t)$$

easily to derive by a Taylor expansion of the l.h.s. and

$$\mathrm{e}^{Cx\frac{\partial}{\partial x}\Delta t} f(x) = \mathrm{e}^{C\frac{\partial}{\partial \ln x}\Delta t} f(\mathrm{e}^{\ln x}) = f(\mathrm{e}^{C\Delta t} x)$$

The algorithm then reads:

Steps 1–3. Application of operators $\mathrm{i}L_{14}$–$\mathrm{i}L_{12}$ from the left:

$$\eta^{\mathrm{I}} = \eta(t) + \frac{P_S(t)}{M_S}\frac{\Delta t}{2}$$

$$\xi^{\mathrm{I}} = \xi(t) + \frac{P_\xi(t)}{M_\xi}\frac{\Delta t}{2}$$

$$P_\xi^{\mathrm{I}} = P_\xi(t) + \left[\frac{P_V(t)^2}{M_V} - k_{\mathrm{B}} T_{\mathrm{EXT}}\right]\frac{\Delta t}{2}$$

Steps 4–5. Application of operators $\mathrm{i}L_{11}$–$\mathrm{i}L_{10}$:

At the end of Step 5, the propagation of atoms' momenta is done. Here is where our algorithm differs from the standard procedure. In the following steps, the standard algorithm and ours will be explained separately.

- Standard procedure

In the standard procedure, $\mathbf{p}'^{\mathrm{I}}_{\mu i}$ are calculated from $\mathbf{p}'_{\mu i}(t)$, the momenta of the atoms at time t. The application of L_{11} on $\mathbf{p}'_{\mu i}$ leads to

$$\mathbf{p}'^{\mathrm{I}}_{\mu i} = \mathbf{p}'_{\mu i}(t) a(t)$$

where

$$a(t) = \exp\left[-\frac{\Delta t}{2}\frac{P_S(t)}{M_S}\right]$$

Then, the application of L_{10} leads to

$$\mathbf{p}'^{\mathrm{II}}_{\mu i} = \mathbf{p}'^{\mathrm{I}}_{\mu i} + \frac{\Delta t}{2}\left[\mathbf{F}_{\mu i}(t) + \mathbf{G}_{\mu i}(t) - \frac{m_{\mu i}}{M_\mu}\mathbf{F}_\mu(t)\right]$$

At this point, $\mathbf{G}_{\mu i}(t)$ is unknown, then the equation is split in two to isolate the term coming from the constraints:

$$\mathbf{p}'^{\mathrm{II}}_{\mu i} = \mathbf{p}'^{\mathrm{WII}}_{\mu i} + \mathbf{p}'^{\mathrm{CII}}_{\mu i}$$

Only:

$$\mathbf{p}'^{\mathrm{WII}}_{\mu i} = \mathbf{p}'^{\mathrm{I}}_{\mu i} + \frac{\Delta t}{2}\left[\mathbf{F}_{\mu i}(t) - \frac{m_{\mu i}}{M_\mu}\mathbf{F}_\mu(t)\right] = \mathbf{p}'^{\mathrm{II}}_{\mu i} - \frac{\Delta t}{2}\mathbf{G}_{\mu i}(t)$$

can be calculated. $\mathbf{p}'^{\mathrm{CII}}_{\mu i} = \frac{\Delta t}{2}\mathbf{G}_{\mu i}(t)$ and $\mathbf{p}'^{\mathrm{II}}_{\mu i}$ will remain unknown until Step 14, when the positions will be implemented and plugged into SHAKE procedure to get Lagrange multipliers at t.

- Our algorithm

As said before, in the standard procedure $\mathbf{p}'_{\mu i}(t)$ have been obtained in the implementation of the previous step by applying SHAKE on velocities. Let us suppose now that we did not apply SHAKE on velocities in the previous step and that, then, $\mathbf{p}'_{\mu i}(t)$ are still unknown. The momenta we know at this stage are only incomplete momenta, which do not include constraint forces:

$$\mathbf{p}'^{*}_{\mu i}(t) = \mathbf{p}'_{\mu i}(t) - \frac{\Delta t}{2}\mathbf{G}_{\mu i}(t)\exp\left[-\frac{\Delta t}{2}\frac{P_S(t)}{M_S}\right]$$

Neither $\mathbf{p}'_{\mu i}(t)$ nor $\mathbf{G}_{\mu i}(t)$ are known. L_{11} and L_{10} are then applied to the pseudo momenta $\mathbf{p}'^{*}_{\mu i}(t)$. To make the understanding easier, from now we will note with an asterisk the momenta obtained from this pseudo momenta ($\mathbf{p}'^{*\mathrm{I}}_{\mu i}$, $\mathbf{p}'^{*\mathrm{II}}_{\mu i}$, etc.) in comparison with the momenta calculated from real $\mathbf{p}'_{\mu i}$ at time t ($\mathbf{p}'^{\mathrm{I}}_{\mu i}$, $\mathbf{p}'^{\mathrm{II}}_{\mu i}$, etc.).

By applying L_{11} on the pseudo momenta, we get

$$\mathbf{p}'^{*\mathrm{I}}_{\mu i} = \mathbf{p}'^{*}_{\mu i}(t)a(t)$$

it is easy to show that real and pseudo momenta are linked:

$$\mathbf{p}'^{*\mathrm{I}}_{\mu i} = \mathbf{p}'^{\mathrm{I}}_{\mu i} - \frac{\Delta t}{2}\mathbf{G}_{\mu i}(t)a(t)^2$$

The application of L_{10} on $\mathbf{p}'^{*\mathrm{I}}_{\mu i}$ then leads to:

$$\mathbf{p}'^{*\mathrm{II}}_{\mu i} = \mathbf{p}'^{*\mathrm{I}}_{\mu i} + \frac{\Delta t}{2}\left[\mathbf{F}_{\mu i}(t) + \mathbf{G}_{\mu i}(t) - \frac{m_{\mu i}}{M_\mu}\mathbf{F}_\mu(t)\right] = \mathbf{p}'^{*\mathrm{WII}}_{\mu i} + \mathbf{p}'^{*\mathrm{CII}}_{\mu i}$$

which implicitly defines $\mathbf{p}'^{*\mathrm{CII}}_{\mu i}$ and gives

$$\mathbf{p}'^{*\mathrm{WII}}_{\mu i} = \mathbf{p}'^{\mathrm{II}}_{\mu i} - \frac{\Delta t}{2}\mathbf{G}_{\mu i}(t)[a(t)^2 + 1]$$

At this stage, $\mathbf{p}'^{*\mathrm{WII}}_{\mu i}$ is calculated while $\mathbf{p}'^{*\mathrm{CII}}_{\mu i}$ and $\mathbf{p}'^{\mathrm{II}}_{\mu i}$ will remain unknown until Step 14, like in the standard procedure.

Steps 6–7. Application of propagators iL_9–iL_8:

All transformations concerning center-of-mass positions and momenta do not raise any problem since $\sum_{i=1}^{n_\mu}\mathbf{G}_{\mu i} = 0$ in each molecule,

$$\mathbf{P}^{\mathrm{I}}_\mu = \mathbf{P}_\mu(t)b(t)$$

where

$$b(t) = \exp\left[-\frac{\Delta t}{2}\left(\frac{P_S(t)}{M_S} + \frac{P_V(t)}{3V(t)M_V}\right)\right]$$

and

$$\mathbf{P}^{\mathrm{II}}_\mu = \mathbf{P}^{\mathrm{I}}_\mu + \frac{\Delta t}{2}\mathbf{F}_\mu(t)$$

Step 8. Application of propagator iL_7:

$$P^{\mathrm{I}}_S = P_S(t) + gk_{\mathrm{B}}[T^{\mathrm{I}} - T_{\mathrm{EXT}}]\frac{\Delta t}{2}$$

with

$$T^{\mathrm{I}} = \frac{1}{gk_{\mathrm{B}}}\left[\sum_{i,\mu}\frac{\mathbf{p}'^{\mathrm{II}2}_{\mu i}}{m_{\mu i}} + \sum_\mu \frac{\mathbf{P}^{\mathrm{II}2}_\mu}{M_\mu}\right]$$

At this step, we are not able to calculate T^{I} since the expression of $\mathbf{p}'^{\mathrm{II}}_{\mu i}$ is unknown (in the standard procedure as well as in our algorithm). The completion of this step is postponed.

Steps 9–11. Application of propagators iL_6–iL_4:

$$P^{\mathrm{I}}_V = P_V(t)\exp\left(-\frac{P^{\mathrm{I}}_\xi}{M_\xi}\frac{\Delta t}{2}\right)$$

$$P^{\mathrm{II}}_V = P^{\mathrm{I}}_V + [\Pi^{\mathrm{I}} - \Pi_{\mathrm{EXT}}]\frac{\Delta t}{2}$$

with

$$\Pi^{\mathrm{I}} = \frac{1}{3V(t)}\sum_{\mu}\left[\frac{\mathbf{P}_{\mu}^{\mathrm{II}2}}{M_{\mu}} + \mathbf{F}_{\mu}(t).\mathbf{R}_{\mu}(t)\right] - \frac{\partial\phi}{\partial V}(t)$$

$$V^{\mathrm{I}} = V(t) + \frac{P_V^{\mathrm{II}}}{M_V}\frac{\Delta t}{2}$$

Steps 12–13. Application of propagators $\mathrm{i}L_3$–$\mathrm{i}L_2$:

$$\mathbf{R}_{\mu}^{\mathrm{I}} = \mathbf{R}_{\mu}(t)\exp\left[\frac{\Delta t}{2}\frac{P_V^{\mathrm{II}}}{3V^{\mathrm{I}}M_V}\right]$$

$$\mathbf{R}_{\mu}^{\mathrm{II}} = \mathbf{R}_{\mu}^{\mathrm{I}} + \frac{\Delta t}{2}\frac{\mathbf{P}_{\mu}^{\mathrm{II}}}{M_{\mu}}$$

Steps 14. Application of propagator $\mathrm{i}L_1$:
The application of $\mathrm{i}L_1$ on $\mathbf{r}'_{\mu i}(t)$ should lead to:

$$\mathbf{r}'_{\mu i}(t+\Delta t) = \mathbf{r}'_{\mu i}(t) + \Delta t\frac{\mathbf{p}'^{\mathrm{II}}_{\mu i}}{m_{\mu i}}$$

However $\mathbf{p}'^{\mathrm{II}}_{\mu i}$ is unknown.

- Standard procedure

In the standard procedure,

$$\mathbf{r}'^{\mathrm{I}}_{\mu i} = \mathbf{r}'_{\mu i}(t) + \Delta t\frac{\mathbf{p}'^{\mathrm{WII}}_{\mu i}}{m_{\mu i}}$$

is first calculated. Then $\mathbf{r}'^{\mathrm{I}}_{\mu i}$ is plugged in the subroutine SHAKE to obtain the Lagrange multipliers $\lambda_{\mu}^{\alpha}(t)$ and finally get $\mathbf{r}'_{\mu i}(t+\Delta t)$. $\mathbf{p}'^{\mathrm{CII}}_{\mu i}$ and then $\mathbf{p}'^{\mathrm{II}}_{\mu i}$ can be finally calculated, then T^{I} and P_S^{I} of step (8) too. It should be noted here that, in the standard procedure, at this stage, the Lagrangian multipliers at time t could be known as computed from imposing the constraints $\dot{\sigma} = 0$. The use of those λ's, however, is avoided since it would induce instabilities in the conservation of the σ's.

- Our algorithm

The implementation is done here with $\mathbf{p}'^{*\mathrm{II}}_{\mu i}$:

$$\mathbf{r}'^{*\mathrm{I}}_{\mu i} = \mathbf{r}'_{\mu i}(t) + \Delta t\frac{\mathbf{p}'^{*\mathrm{II}}_{\mu i}}{m_{\mu i}}$$

and

$$\mathbf{r}'^{*\mathrm{I}}_{\mu i} = \mathbf{r}'_{\mu i}(t+\Delta t) - \frac{\Delta t^2}{2m_{\mu i}}[1 + a(t)^2]\mathbf{G}_{\mu i}$$

Like previously, $\mathbf{r}'^{*\mathrm{I}}_{\mu i}$ can be plugged in the subroutine SHAKE to obtain the Lagrange multipliers λ_{μ}^{α} and finally $\mathbf{r}'_{\mu i}(t+\Delta t)$ from $\mathbf{r}'^{*\mathrm{I}}_{\mu i}$.

At this stage, the real momenta $\mathbf{p}'^{\mathrm{II}}_{\mu i} = \mathbf{p}'^{*\mathrm{II}}_{\mu i} - \frac{\Delta t}{2}\mathbf{G}_{\mu i}(t)[a(t)^2 + 1]$ can be calculated, then T^{I} and P_S^{I} of step (8) too.

So it is for the kinetic energy at time t, which can also be evaluated from the expressions of the momenta at time t

$$\mathbf{p}'_{\mu i}(t) = \mathbf{p}'^{*}_{\mu i}(t) + \frac{\Delta t}{2}\mathbf{G}_{\mu i}(t)\exp\left[-\frac{\Delta t}{2}\frac{P_S(t)}{M_S}\right]$$

unknown until now.

Steps 15–16. Application of propagators $\mathrm{i}L_2$–$\mathrm{i}L_3$:

$$\mathbf{R}_{\mu}^{\mathrm{III}} = \mathbf{R}_{\mu}^{\mathrm{II}} + \frac{\Delta t}{2}\frac{\mathbf{P}_{\mu}^{\mathrm{II}}}{M_{\mu}}$$

$$\mathbf{R}_{\mu}(t+\Delta t) = \mathbf{R}_{\mu}^{\mathrm{III}}\exp\left[\frac{\Delta t}{2}\frac{P_V^{\mathrm{II}}}{3V^{\mathrm{I}}M_V}\right]$$

Steps 17–19. Application of propagators iL_4–iL_6:

$$V(t+\Delta t) = V^{\mathrm{I}} + \frac{P_V^{\mathrm{II}}}{M_V}\frac{\Delta t}{2}$$

Here, all energies and forces at $t+\Delta t$ can be evaluated from $\mathbf{r}'_{\mu i}(t+\Delta t)$, $\mathbf{R}_\mu(t+\Delta t)$ and $V(t+\Delta t)$ at time $t+\Delta t$.

Then,

$$P_V^{\mathrm{III}} = P_V^{\mathrm{II}} + [\Pi^{\mathrm{II}} - \Pi_{\mathrm{EXT}}]\frac{\Delta t}{2}$$

where

$$\Pi^{\mathrm{II}} = \frac{1}{3V(t+\Delta t)}\sum_\mu \left[\frac{\mathbf{P}_\mu^{\mathrm{II}2}}{M_\mu} + \mathbf{F}_\mu(t+\Delta t).\mathbf{R}_\mu(t+\Delta t)\right] - \frac{\partial\phi}{\partial V}(t+\Delta t)$$

Finally,

$$P_V(t+\Delta t) = P_V^{\mathrm{III}}\exp\left(-\frac{P_\xi^{\mathrm{I}}}{M_\xi}\frac{\Delta t}{2}\right)$$

Step 20, application of propagator iL_7:

$$P_S(t+\Delta t) = P_S^{\mathrm{I}} + gk_{\mathrm{B}}[T^{\mathrm{I}} - T_{\mathrm{EXT}}]\frac{\Delta t}{2}$$

where T^{I} has the same expression as before.

Steps 21–22. Application of propagators iL_8–iL_9 :

$$\mathbf{P}_\mu^{\mathrm{III}} = \mathbf{P}_\mu^{\mathrm{II}} + \frac{\Delta t}{2}\mathbf{F}_\mu(t+\Delta t)$$
$$\mathbf{P}_\mu(t+\Delta t) = \mathbf{P}_\mu^{\mathrm{III}} b(t+\Delta t)$$

where

$$b(t+\Delta t) = \exp\left[-\frac{\Delta t}{2}\left(\frac{P_S(t+\Delta t)}{M_S} + \frac{P_V(t+\Delta t)}{3V(t+\Delta t)M_V}\right)\right]$$

Steps 23–24. Application of operators iL_{10}–iL_{11}:

By applying iL_{10} and iL_{11} on $\mathbf{p}'^{\mathrm{II}}_{\mu i}$, we get

$$\mathbf{p}'^{\mathrm{III}}_{\mu i} = \mathbf{p}'^{\mathrm{II}}_{\mu i} + \frac{\Delta t}{2}\left[\mathbf{F}_{\mu i}(t+\Delta t) + \mathbf{G}_{\mu i}(t+\Delta t) - \frac{m_{\mu i}}{M_\mu}\mathbf{F}_\mu(t+\Delta t)\right]$$

And then

$$\mathbf{p}'_{\mu i}(t+\Delta t) = \mathbf{p}'^{\mathrm{III}}_{\mu i} a(t+\Delta t)$$

However, $\mathbf{G}_{\mu i}(t+\Delta t)$ is unknown. Then the constraint term $\mathbf{p}'^{\mathrm{CIII}}_{\mu i}$ must be isolated and only:

$$\mathbf{p}'^{\mathrm{WIII}}_{\mu i} = \mathbf{p}'^{\mathrm{II}}_{\mu i} + \frac{\Delta t}{2}\left[\mathbf{F}_{\mu i}(t+\Delta t) - \frac{m_{\mu i}}{M_\mu}\mathbf{F}_\mu(t+\Delta t)\right]$$

and then

$$\mathbf{p}'^{\mathrm{W}}_{\mu i}(t+\Delta t) = \mathbf{p}'^{\mathrm{WIII}}_{\mu i} a(t+\Delta t)$$

can be calculated.

In order to get

$$\mathbf{p}'_{\mu i}(t+\Delta t) = \mathbf{p}'^{\mathrm{W}}_{\mu i}(t+\Delta t) + \mathbf{p}'^{\mathrm{C}}_{\mu i}(t+\Delta t) = \mathbf{p}'^{\mathrm{W}}_{\mu i}(t+\Delta t) + \frac{\Delta t}{2}\mathbf{G}_{\mu i}(t+\Delta t)a(t+\Delta t)$$

$\mathbf{p}'^{\mathrm{W}}_{\mu i}(t+\Delta t)$ can be plugged in the SHAKE subroutine to obtain $\lambda^{\alpha}_{\mu}(t+\Delta t)$ and then $\mathbf{G}_{\mu i}(t+\Delta t)$. It is what is done in the standard procedure.

In our algorithm instead, $\mathbf{p}'_{\mu i}(t+\Delta t)$ will remain unknown and the new pseudo momentum:

$$\mathbf{p}'^{*}_{\mu i}(t+\Delta t) = \mathbf{p}'^{\mathrm{W}}_{\mu i}(t+\Delta t) = \mathbf{p}'_{\mu i}(t+\Delta t) - \frac{\Delta t}{2}\mathbf{G}_{\mu i}(t+\Delta t)a(t+\Delta t)$$

will be used for the implementation in the next step, until the determination of $\mathbf{r}'_{\mu i}(t+2\Delta t)$ will allow the calculation of $\mathbf{p}'_{\mu i}(t+\Delta t)$.

Steps 25–27. Application of propagators iL_{10}–iL_{12} :

Finally,

$$P_{\xi}(t+\Delta t) = P^{\mathrm{I}}_{\xi} + \left[\frac{P_V(t+\Delta t)^2}{M_V} - k_{\mathrm{B}}T_{\mathrm{EXT}}\right]\frac{\Delta t}{2}$$

$$\xi(t+\Delta t) = \xi^{\mathrm{I}} + \frac{P_{\xi}(t+\Delta t)}{M_{\xi}}\frac{\Delta t}{2}$$

$$\eta(t+\Delta t) = \eta^{\mathrm{I}} + \frac{P_S(t+\Delta t)}{M_S}\frac{\Delta t}{2}$$

It can be useful to note that this algorithm can be adapted for molecular dynamics in (NVT). It is sufficient to eliminate the equations involving the variables ξ, P_{ξ}, V and P_V, while taking $P_V = 0$ in all the other equations. Additionally, for MD in (NVE) ensemble, it is necessary to eliminate also the equations involving η and P_S, while taking $P_S = 0$ in all the other equations.

We have so demonstrated that the standard procedure, based on RATTLE algorithm, can be replaced by a more recursive method, where unknown velocities are carried over the step without calling a second time SHAKE to compute them. In the next section, we check the consistency between RATTLE and our algorithm.

3. Illustration and results

3.1. Bulk water

The algorithm was implemented and tested on a system constituted of 256 SCP/E water molecules [11]. As SPC/E water is rigid, there are three holonomic constraints per molecule of the type: $\sigma_{ij} = (\mathbf{r}_{\mu i} - \mathbf{r}_{\mu j})^2 - d^2_{ij} = 0$ or $\sigma_{ij} = (\mathbf{r}'_{\mu i} - \mathbf{r}'_{\mu j})^2 - d^2_{ij} = 0$.

As a consequence:

$$\dot{\sigma}_{ij} = 2(\mathbf{r}'_{\mu i} - \mathbf{r}'_{\mu j}).\left(\frac{\mathbf{p}'_{\mu i}}{m_{\mu i}} - \frac{\mathbf{p}'_{\mu j}}{m_{\mu i}}\right) = 0$$

The results we got with our algorithm were compared with the standard algorithm detailed before and in which the constraints are applied twice, as it is done in RATTLE.

The simulations on water were performed with $P = 1$ bar, $T = 298$ K. The inertial factors M_S, M_{ξ} and M_V are linked to the characteristic times of the response of the variables η, ξ and V. Possible expressions for these times, τ_{η}, τ_{ξ} and τ_V, can be obtained by rewriting the equations of motion as follows:

$$\tau^2_V\frac{\ddot{V}}{V} = \left[\frac{\Pi}{\Pi_{\mathrm{EXT}}} - 1\right] - \tau^2_V\dot{\xi}\left(\frac{\dot{V}}{V}\right)$$

$$\tau^2_S\ddot{\eta} = \frac{T}{T_{\mathrm{EXT}}} - 1$$

$$\tau^2_{\xi}\ddot{\xi} = \tau^2_V\frac{\Pi_{\mathrm{EXT}}V}{k_{\mathrm{B}}T_{\mathrm{EXT}}}\left(\frac{\dot{V}}{V}\right)^2 - 1$$

where we have defined $\tau^2_V = \frac{M_V V}{\Pi_{\mathrm{EXT}}}$, $\tau^2_S = \frac{M_S}{gk_{\mathrm{B}}T_{\mathrm{EXT}}}$, and $\tau^2_{\xi} = \frac{M_{\xi}}{k_{\mathrm{B}}T_{\mathrm{EXT}}}$.

Several time constants were tested for both algorithms. A choice of a too short time constant can lead to bad dynamical properties of the system and make it unstable. This is the case when τ_V is chosen smaller than 1 ps. For this value of τ_V, the average period of volume fluctuations is much too fast and the algorithms are unstable. Values of τ_V were tested from 1 ps to 2000 ps. Volume fluctuations were very slow with the latter value, with a period of about 100 ps. The algorithms were less sensitive to τ_S and τ_ξ.

The simulations whose results are given here were undertaken with values for τ_V, τ_S and τ_ξ equal respectively to 6 ps, 60 fs and 60 fs. The time step was $\Delta t = 1$ fs. The SHAKE tolerance was chosen to be 10^{-12} for $\dot{\sigma}_{ij}$ and 10^{-10} for σ_{ij}. The system was first equilibrated and simulations of 1 million steps (1000 ps) were undertaken in order to get equilibrium properties. The average values are given in Table 1 for both algorithms.

As shown in Fig. 1 and in Table 1, the pseudo-hamiltonian H' is well conserved and the average quantities are very close in both algorithms.

We also checked that the pair correlation functions g(O–O), g(O–H) and g(H–H) described well the structure of SPC/E water [11,12].

Table 1
Average quantities of 1 ns simulations obtained by both algorithms.

	SPC/E		Polymer
	RATTLE	This article	This article
H'	−37.8514	−37.8532	−5.4061
$\delta H'$	0.0059	0.0052	0.0015
E_{TOT}	−39.029	−39.026	−5.8526
δE_{TOT}	0.496	0.492	0.1534
E_P	−46.450	−46.447	−28.628
δE_P	0.420	0.416	0.094
T	297.46	297.45	134.99
δT	10.77	10.69	2.65
P	−0.0018	−0.0021	4.30
δP	0.8438	0.8417	0.43
V	7701	7709	28328
δV	122	120	102

All the energies are given in kJ/mol, the temperature in K and the pressure in kbar.

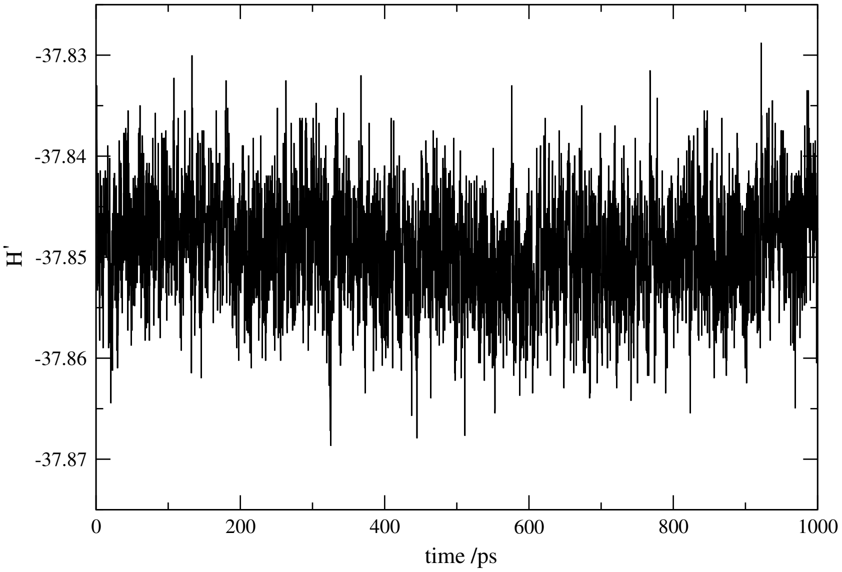

Fig. 1. NPT behavior of the conserved quantity H' for a system constituted of 256 SPC/E water molecules.

As momenta at t are calculated indirectly from Lagrange multipliers coming from SHAKE procedure on atomic positions, constraints on them may not be well conserved. Indeed, for each couple of atoms (i, j) in a molecule, we should have: $\dot{\sigma}_{ij} = 2(\mathbf{r}'_{\mu i} - \mathbf{r}'_{\mu j}).\left(\frac{\mathbf{p}'_{\mu i}}{m_{\mu i}} - \frac{\mathbf{p}'_{\mu j}}{m_{\mu i}}\right) = 0$. Because our algorithm is not exact in the constraints on momenta, $\dot{\sigma}_{ij}$ cannot be as small as if a SHAKE procedure were applied directly to momenta at t.

As a test, we made an histogram of $\frac{\dot{\sigma}_{ij}}{2|\mathbf{r}'_{\mu i}-\mathbf{r}'_{\mu j}|\left|\frac{\mathbf{p}'_{\mu i}}{m_{\mu i}}-\frac{\mathbf{p}'_{\mu j}}{m_{\mu i}}\right|}$. It represents the cosinus of the angle between $\mathbf{r}'_{\mu i} - \mathbf{r}'_{\mu j}$ and $\frac{\mathbf{p}'_{\mu i}}{m_{\mu i}} - \frac{\mathbf{p}'_{\mu j}}{m_{\mu i}}$. The histogram of the corresponding angle is given in Fig. 2. It looks like a sum of a Dirac peak, and a gaussian distribution, around 90°, which means that the vectors are, in average, orthogonal. The error on the angle can be then evaluated at about 0.1%, which does not disturb neither the average of the kinetic energy, nor the dynamical properties of the system : self-diffusion coefficients calculated from vacf and mean-squared displacements agreed. To be completely convinced, we also checked that the spectra of O and H velocity auto-correlation functions, obtained from trajectories of 1 ns in NVE ensemble with a time step of $\Delta t = 1$ fs with both algorithms, exhibited the same characteristics of water short time dynamics.

The equivalence of the two algorithms is not surprising since in our approach the (numerical) error in the momenta constraints is not propagated in time. In both algorithms, SHAKE calculates Lagrange multipliers which differ from the analytical ones by a term which has the same order of error as the Verlet algorithm. Therefore the two trajectories obtained by the two algorithms starting with the same initial configuration should differ by the same order. In Fig. 3 we show the distance in configuration space of two trajectories, i.e the sum $\sum_i(\mathbf{r}_{i1} - \mathbf{r}_{i2})^2$, in function of time. We see that the two trajectories become completely uncorrelated after 1.8–2.2 ps and that this time does not depend drastically on the time step (to improve further the behavior with decreasing time step, one should also increase the precision of the calculations beyond double precision). We can conclude that the two algorithms are rigorously equivalent on the timescale of the characteristic times of the various time correlation functions of physical interest (of the order of few ps). After this time, the two algorithms will equivalently differ from the real trajectory. The only way to reduce numerical error on larger characteristic times is to increase the numerical precision of the calculations and decrease the time step for both algorithms.

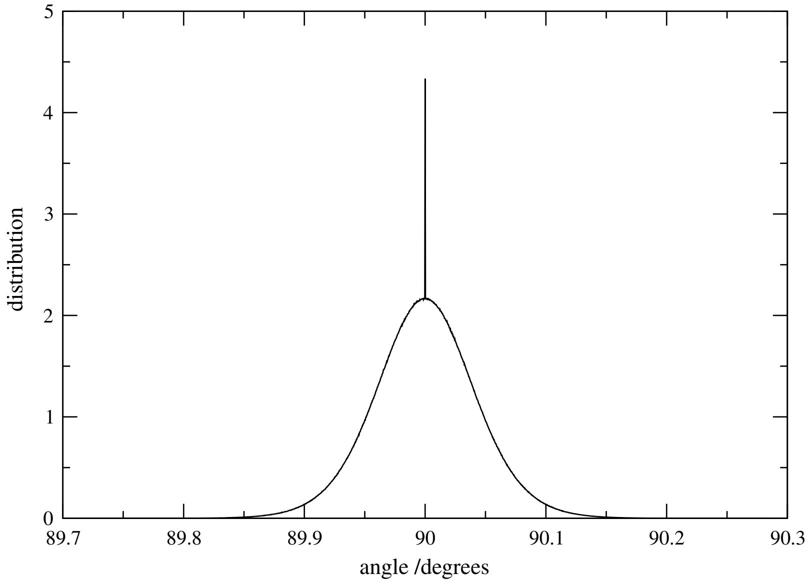

Fig. 2. Distribution of angles between $\mathbf{r}'_{\mu i} - \mathbf{r}'_{\mu j}$ and $\frac{\mathbf{p}'_{\mu i}}{m_{\mu i}} - \frac{\mathbf{p}'_{\mu j}}{m_{\mu i}}$ for water atoms.

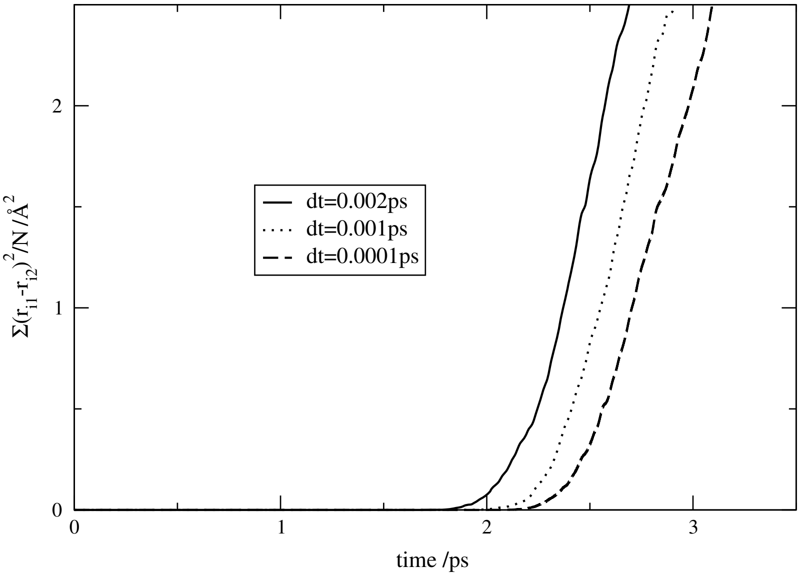

Fig. 3. The cartesian distance, $\sum_i(\mathbf{r}_{i1} - \mathbf{r}_{i2})^2/N$, in time, for different timesteps, of the two configurational trajectories computed using the standard RATTLE algorithm and the one introduced in this paper. The trajectories remain very close until 1.8–2.2 ps.

3.2. Polymer

In the water example, no CPU time is gained by our algorithm since the overhead of the SHAKE procedure is negligible compared to the calculation of the interactions.

However, it can be interesting to see what happens when systems with many holonomic constraints are considered. Tests were done on a system containing a linear polymer of 2000 atoms immersed in an Argon fluid of 400 atoms. The distance between two adjacent atoms in the polymer was constrained to be equal to 1.41 Å. LJ parameters for Argon were taken from Levesque and Verlet [13]: $\epsilon_{\mathrm{Ar}} = 0.996$ kJ/mol and $\sigma_{\mathrm{Ar}} = 3.405$ Å. For the atoms of the polymer, we took: $\epsilon = 0.485$ kJ/mol and $\sigma = 2.20$ Å. LJ parameters between Ar and polymer atoms were obtained by the Lorentz–Berthelot rules:

$$\sigma_{ij} = \frac{\sigma_i + \sigma_j}{2} \tag{2}$$

$$\epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j} \tag{3}$$

In order to simulate a quite dense system, the pressure and the temperature were chosen to be: $P = 4.3$ kbar and $T = 135$ K. The time step and the masses of barostats and thermostats were the same as in the previous section.

Average quantities obtained after stabilization of the box volume are given in Table 1 for our algorithm. Here again, the pseudo-hamiltonian is well conserved.

The speed of both algorithms were compared. Because the polymer is quite long, the evaluation of Lagrange multipliers at the beginning of the SHAKE procedure is not done by the matrix method which would require the inversion of a matrix of large dimensions. The constraints were then calculated with the iterative procedure of Ciccotti and Ryckaert [14]. The tolerance was chosen to be 10^{-12} for $\dot{\sigma}_{ij}$ and 10^{-6} for σ_{ij}. In the standard algorithm, an average of 56 iterations were necessary for SHAKE on positions to converge (step 14) and 80 iterations for SHAKE on velocities (step 24). In our algorithm, 63 iterations were needed for SHAKE on positions to converge. A decrease of the SHAKE-positions tolerance to 10^{-7} leads to an approximate increase of the number of iterations by 5. Let us note that except for very small timesteps, the number of iterations remains in the same order of magnitude.

In both algorithms, about 99% of the CPU time was spent in the subroutine calculating the interactions between the atoms (step 17) and in SHAKE (steps 14 and 24). The usual RATTLE algorithm spent between

67% and 70% of the CPU time in the calculation of the interactions, 17–18% in the SHAKE procedure for the positions and 12–13% in the SHAKE procedure for the velocities. In our algorithm, these percentages became 77–78%, 21–22% and 0, since SHAKE for velocities is not called, with a global CPU time gain of 10–11%. Simulations were carried out with polymers of various sizes, with the same number of gas atomes. For polymers with 500 and 1000 atoms, the percentages spent in each subroutine were almost the same, leading to a global CPU time gain between 9% and 11%. No obvious trend could be found in the gain in CPU time versus the number of polymer atoms. The gain should depend on the number of interactions to be calculated, as well as on the number of constraints (and the number of iterations necessary to converge in SHAKE), so that the previous estimations can be just useful to give an idea. Note, however, that we used a simple algorithm to calculate the interactions between atoms. Every algorithm which will increase the speed of this calculation (neighbour-list method or linked cell algorithm for example) should increase also the relative percentage of CPU time spent in the SHAKE procedure, implying a better gain in CPU time.

In conclusion, the algorithm we present here is not only without inconsistencies but spends less time in the constraints calculation than the standard RATTLE algorithm. The saving can be non negligible, especially for systems with a large number of constraints.

4. Conclusion

We have introduced a variant of the standard RATTLE algorithm used when performing NVE, NVT or NPT molecular dynamics simulations of molecular systems with holonomic constraints. We showed that SHAKE does not need to be applied twice in a step since momenta at t can be kept unknown until the middle of the following step. The procedure eliminates the inconsistency present in Andersen's velocity algorithm which computes twice, and with different procedure, the values of the Lagrange multipliers at the same time. This makes the algorithm faster, although not dramatically, in particular for large (e.g biological) molecules, connected by a large number of constraints. For this purpose, we employed the recursive method based on the procedure developed by Tuckerman et al. [9], using Trotter formula, and extended to the case of systems subjected to constraints in Refs. [5,10]. This method allows to calculate properly the constraint forces. An identical recursive procedure could be applied to the more interesting case of coexistence of constraints and multiple time step algorithm. Work is in progress in that direction.

References

[1] J. Ryckaert, G. Ciccotti, H. Berendsen, J. Comput. Phys. 23 (1977) 327.
[2] G. Ciccotti, M. Ferrario, J.-P. Ryckaert, Mol. Phys. 47 (1982) 1253.
[3] H. Andersen, J. Comput. Phys. 52 (1983) 24.
[4] M. Tuckerman, Y. Liu, G. Ciccotti, G. Martyna, J. Chem. Phys. 115 (2001) 1678.
[5] G. Kalibaeva, M. Ferrario, G. Ciccotti, Mol. Phys. 101 (2003) 765.
[6] G. Martyna, D. Tobias, M. Klein, J. Chem. Phys. 101 (1994) 4177.
[7] G. Martyna, M. Klein, M. Tuckerman, J. Chem. Phys. 97 (1992) 2635.
[8] G. Martyna, M. Tuckerman, D. Tobias, M. Klein, Mol. Phys. 87 (1996) 1117.
[9] M. Tuckerman, B. Berne, G. Martyna, J. Chem. Phys. 97 (1992) 1990.
[10] G. Ciccotti, G. Kalibaeva, Phil. Trans. R. Soc. A 362 (2004) 1583.
[11] H. Berendsen, J. Grigera, T. Straatsma, J. Phys. Chem. 91 (1987) 6269.
[12] D. van der Spoel, P. van Maaren, H. Berendsen, J. Chem. Phys. 108 (1998) 10220.
[13] D. Levesque, L. Verlet, Phys. Rev. 182 (1969) 307.
[14] G. Ciccotti, J.-P. Ryckaert, Comput. Phys. Rep. 4 (1986) 347.