

Parallel Implementation of the ‘Multiple  
Time Scales’ Algorithm for Molecular  
Dynamics Applied to Flexible Water  
Models \*

Carlo Nardone      Pietro Rossi      Maria Valentini

CRS4,

Center for Advanced Studies, Research and Development in Sardinia

via N.Sauro 10, I-09123 Cagliari, Italy

7th Apr 1993

---

\*submitted to *J. Comp. Chem.*

## **Abstract**

An algorithm to handle multiple time scales (MTS) is applied to a Molecular Dynamics simulation of flexible water. Being explicitly time reversible and area preserving, in the sense of Liouville's theorem, this algorithm ensures extremely good energy conservation with no appreciable drift. It is shown that MTS integration schemes yield the same accuracy as Single Time Scales but allow for a much longer time step. Details of the implementation on a Massively Parallel computer are also discussed.

**Key words:** Molecular Dynamics, Multiple Time Scales, Parallel Algorithms, Water

## Introduction

Molecular Dynamics (MD) is a well-established simulation technique to infer the statistical and dynamical properties of an ensemble of molecules, starting from an appropriate form of the interaction potential. As the computational power has increased, the demands on a MD simulation becomes more and more sophisticated. For example, a typical run would follow the evolution of a macromolecule of several thousand atoms embedded in its solvent of several thousand water molecules, for few hundred picoseconds ( $1 \text{ ps} = 10^{-12} \text{ s}$ ). This run would take tens of days on a vector supercomputer if all the interactions are taken into account (no potential cutoff). A more efficient approach requires taking advantage of the inherent data-parallelism of the problem porting it onto a massively parallel processor (MPP). This, in turn, requires choosing an efficient strategy for treating the intramolecular degrees of freedom.

Some details on the treatment of polyatomic molecules are needed for what follows. Suppose that one does not want to model the intramolecular dynamics because its time scale is smaller than the phenomena of interest, in that case, the covalent bonds and the torsional angles are treated as rigid.

This allows one to fix a time step of the order of 1 femtosecond ( $1 \text{ fs} = 10^{-15} \text{ s}$ ). The two commonly used methods to simulate rigid polyatomic molecules are the rigid body equations of motion and the so-called method of constraints. The former is based on writing Euler equation for a rigid body, whereas the latter relies on adding Lagrange's multipliers to the forces, such that the constraints are fulfilled at each time step [1]. Those forces are found by solving a set of nonlinear equations, whose linearization is a sparse matrix, using an iterative scheme (the SHAKE algorithm or its variations [2]).

A totally different approach would be to consider non-rigid molecules, where atoms follow their own dynamics. In this case, the constraining bond lengths and angles would oscillate within suitably defined intramolecular potentials. As a bonus, one could study the dynamics of those internal modes, although for a proper treatment a quantum mechanical approach should be followed. In the past, it was maintained that because of the smaller intramolecular time scale, one is forced to reduce the integration time step to a small fraction of fs, with a correspondent reduction of the physical time spanned by the simulation. Therefore flexible models were ruled out as an alternative to rigid models whenever one was not interested in intramolecular dynamics *per se*.

A Multiple Time Scales (MTS) approach can ease this problem. With MTS technique the evolution is split into two components: only one “slow” force calculation is required every  $n$  “fast” force calculations. The fast dynamics comprises the intramolecular oscillations and the slow dynamics comprises the intermolecular interactions. The advantage of adopting this approach stems from the fact that while there are  $O(N^2)$  long-range binary intermolecular interactions, there are only  $O(MN)$  intramolecular potential terms (where  $N$  is the number of molecules and  $M$  is the number of bond lengths and angles per molecule). Therefore MTS offers the same computational complexity as rigid models. In a MPP data-parallel implementation, as long as the number of processors  $p \sim O(N)$ , the computation time of the former scales as  $N$ , but the latter scales only as a constant [3]. A possible advantage over the SHAKE scheme implemented on MPP is due to the better load balancing among processors, because the number of SHAKE iterations is dominated by the worst case. Of course, one must consider that the model under study is radically changed. As pointed out in [1], a constrained system is qualitatively different from a system with “hard” degrees of freedom.

We adopted a MTS algorithm independently described by Sexton and Weingarten for Hybrid Monte Carlo calculations on Lattice Field Theory

[4], and by Tuckerman and coworkers [5]. It has the advantage of being inherently time reversible and area-preserving (in the sense of Liouville's theorem), therefore the energy drift is negligible; furthermore the energy is conserved with an accuracy  $O(\Delta t^2)$ , i.e. equivalent to the standard Verlet or leap-frog integration scheme. Higher-order schemes can easily be formulated using the same formalism.

We choose to consider water because it is a simple, non-trivial example of a polyatomic molecule, which is also of obvious scientific interest. Indeed, one of the first MD simulations on polyatomic molecules, were performed on water using the quaternions method [6]. Furthermore, in [7] the properties of some flexible models of water are described, but the authors, there, used a naive MTS approach which has bad energy conservation properties. To our knowledge, this paper shows for the first time a reversible MTS simulation of a polyatomic molecule.

The rest of paper is organized as follows. In Par.1 we describe the algorithm used, in Par.2 we describe the application to the Molecular Dynamics of liquid water. Par.3 is devoted to the parallel implementation, while we show some results of the simulation in Par.4. Finally, conclusions are drawn in Par.5.

# 1 The MTS algorithm

Here we will describe briefly the MTS algorithm [4, 5] for reader's convenience, using mostly Sexton and Weingarten's formalism.

We introduce the classical Hamiltonian

$$\mathcal{H} = \frac{1}{2} \sum_i \frac{p_i^2}{m_i} + V(q). \quad (1)$$

where  $q = (q_1, \dots, q_n)$  is the set of coordinates and  $p = (p_1, \dots, p_n)$  is the set of conjugate momenta (the index runs over the atoms). Introducing the linear operator

$$\mathcal{L}(f)g = -\{f, g\} = \sum_i \left( \frac{\partial f}{\partial p_i} \frac{\partial g}{\partial q_i} - \frac{\partial f}{\partial q_i} \frac{\partial g}{\partial p_i} \right), \quad (2)$$

we can write the time evolution under the Hamilton equations for any observable  $\Phi(t) = \Phi[p(t), q(t)]$  in the compact form

$$\frac{d\Phi(t)}{dt} = \mathcal{L}(\mathcal{H})\Phi, \quad (3)$$

which has the formal solution  $\Phi(t) = \exp[t\mathcal{L}(\mathcal{H})]\Phi(0)$ . In particular

$$\exp \left[ \Delta t \mathcal{L} \left( \frac{1}{2} \sum_i \frac{p_i^2}{m_i} \right) \right] \Phi(p, q) = \Phi \left( p, q + \Delta t \frac{p}{m} \right) \quad (4)$$

and

$$\exp[\Delta t \mathcal{L}(V(q))]\Phi(p, q) = \Phi \left( p - \Delta t \frac{\partial V(q)}{\partial q_i}, q \right). \quad (5)$$

It can be shown that, for a decomposition of the form  $\mathcal{H} = h_0 + 2h_1 + \dots + 2h_n$ , the approximation

$$\begin{aligned} \exp[\Delta t \mathcal{L}(\mathcal{H})] &\cong \exp[\Delta t \mathcal{L}(h_n)] \times \dots \times \exp[\Delta t \mathcal{L}(h_1)] \exp[\Delta t \mathcal{L}(h_0)] \times \\ &\exp[\Delta t \mathcal{L}(h_1)] \times \dots \times \exp[\Delta t \mathcal{L}(h_n)] \end{aligned} \quad (6)$$

holds within an error  $O(\Delta t^3)$ . When one makes the natural choice  $h_0 = 1/2 \sum_i p_i^2/m_i$  and  $h_1 = 1/2V(q)$  the previous formula applied to  $(p, q)$  gives a special form of the well-known leapfrog update:

$$\begin{aligned} p_i \left( t + \frac{\Delta t}{2} \right) &= p_i(t) + \frac{\Delta t}{2m_i} f_i(q(t)) \\ q_i(t + \Delta t) &= q_i(t) + \frac{\Delta t}{m_i} p_i \left( t + \frac{\Delta t}{2} \right) \\ p_i(t + \Delta t) &= p_i \left( t + \frac{\Delta t}{2} \right) + \frac{\Delta t}{2m_i} f_i(q(t + \Delta t)) \end{aligned} \quad (7)$$

where  $f_i(q) = -\partial V(q)/\partial q_i$ . This 2-step form of the leapfrog scheme is equivalent to the ‘‘velocity Verlet’’ scheme [8], as can be seen by substituting the first equation in the third one. The reversibility stems naturally from the symmetry of the product Eq.6; the area  $d\mu(p)d\mu(q)$  is preserved because of the Hamiltonian character of the propagators Eq.4 and Eq.5. The coefficient of the leading term of the error includes the commutators  $[\mathcal{L}(V), [\mathcal{L}(V), \mathcal{L}(\frac{1}{2} \sum_i \frac{p_i^2}{m_i})]]$  and  $[\mathcal{L}(\frac{1}{2} \sum_i \frac{p_i^2}{m_i}), [\mathcal{L}(V), \mathcal{L}(\frac{1}{2} \sum_i \frac{p_i^2}{m_i})]]$ .



In the MTS case, we split the potential term in the “fast” and the “slow” contribution, i.e.  $h_1 = \frac{1}{2}V_f(q)$  and  $h_2 = \frac{1}{2}V_s(q)$ . Then we apply the fast term  $n$  times. The evolution operator

$$\begin{aligned} & \exp\left[\frac{\Delta t}{2}\mathcal{L}(V_s(q))\right] \times \\ & \left[ \exp\left[\frac{\Delta t}{2n}\mathcal{L}(V_f(q))\right] \exp\left[\frac{\Delta t}{n}\mathcal{L}\left(\frac{1}{2}\sum_i \frac{p_i^2}{m_i}\right)\right] \exp\left[\frac{\Delta t}{2n}\mathcal{L}(V_f(q))\right] \right]^n \times \\ & \exp\left[\frac{\Delta t}{2}\mathcal{L}(V_s(q))\right] \end{aligned} \quad (8)$$

is equal to  $\exp[\Delta t\mathcal{L}(\mathcal{H}) + \Delta t^3\epsilon]$  i.e. again approximately equal to the Hamiltonian evolution with an error  $O(\Delta t^3)$  [4]. Now the coefficient of the leading term of the error  $\epsilon$  includes  $1/n^2$  times the “fast” commutators  $[\mathcal{L}(V_f), [\mathcal{L}(V_f), \mathcal{L}(\frac{1}{2}\sum_i \frac{p_i^2}{m_i})]]$  and  $[\mathcal{L}(\frac{1}{2}\sum_i \frac{p_i^2}{m_i}), [\mathcal{L}(V_f), \mathcal{L}(\frac{1}{2}\sum_i \frac{p_i^2}{m_i})]]$ , plus the “slow” commutators  $[\mathcal{L}(V_s), [\mathcal{L}(V_s), \mathcal{L}(\frac{1}{2}\sum_i \frac{p_i^2}{m_i})]]$  and  $[\mathcal{L}(\frac{1}{2}\sum_i \frac{p_i^2}{m_i}), [\mathcal{L}(V_s), \mathcal{L}(\frac{1}{2}\sum_i \frac{p_i^2}{m_i})]]$ . Expanding  $V_s$  and  $V_f$  around their minima, setting  $\omega_f^2 = \frac{1}{m}(\frac{\partial^2 V_f(q)}{\partial q^2})_{(0)}$  and  $\omega_s^2 = \frac{1}{m}(\frac{\partial^2 V_s(q)}{\partial q^2})_{(0)}$  it can be shown that the error operator reduces to

$$\epsilon\Delta t^3 = \Delta t(\omega_f^2\Delta t^2) \left[ \frac{1}{n^2}\mathcal{H}_1 + \frac{\omega_s^2}{\omega_f^2}\mathcal{H}_2 \right] + O(\Delta t^5) \quad (9)$$

where  $\mathcal{H}_1$  and  $\mathcal{H}_2$  are two operators of the same order of the Hamiltonian. Therefore, in the Single Time Scale (STS) approach (which correspond to

$n = 1$  in the previous formula), the error factor for a given  $\Delta t$  is given by the fast time scale:  $\epsilon \Delta t^3 \approx (\Delta t / \tau_f)^2 \mathcal{H}_1 \Delta t$ , having used the inequality  $\omega_f^2 \gg \omega_s^2$ . On the other hand, in the MTS approach, in the limit  $n \rightarrow \infty$ , the error factor is given by the slow time scale:  $\epsilon \Delta t^3 \approx (\Delta t / \tau_s)^2 \mathcal{H}_2 \Delta t$ , i.e. much smaller for fixed  $\Delta t$ . Physically, the different time scale means that the average curvature around the minimum of the intermolecular potentials is smaller than the steeper “fast” intramolecular potentials ( $\omega_s^2 \ll \omega_f^2$ ).

The algorithm which translates Eq.8 in pseudo-code is the following:

```

initialize ( q(i), p(i) )

f_s(i) <- forces_slow( q(i) )

f_f(i) <- forces_fast( q(i) )

# main loop

do iter = 1,max_iter

  p(i) <- p(i) + dt / 2 * f_s(i)

  do count = 1,n

    p(i) <- p(i) + dt / (2*n) * f_f(i)

    q(i) <- q(i) + dt / n * p (i)

    f_f(i) <- forces_fast( q(i) )

```

```

      p(i) <- p(i) + dt / (2*n) * f_f(i)

    end do

    f_s(i) <- forces_slow( q(i) )

    p(i) <- p(i) + dt / 2 * f_s(i)

  end do

```

where  $f\_s$  is equal to  $-\partial V_s(q)/\partial q_i$  and  $f\_f$  is equal to  $-\partial V_f(q)/\partial q_i$ . Note that one “slow” forces calculation and  $n$  “fast” are required per time step.

## 2 MTS applied to water

We have implemented the algorithm described in Par.1 for the SPC flexible water both in Fortran 77 running on IBM RISC/6000 workstations and in CM Fortran (a dialect of Fortran 90) running on the Connection Machine CM-200. The SPC water model [9] is a 3-sites model where the sites coincide with the O and H nuclei and are assigned respectively -0.82e and 0.41e electric charge. Furthermore, the oxygen atoms interact each other through a Lennard–Jones potential

$$V(r) = 4\epsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right] \quad (10)$$

with interaction length  $\sigma = 3.166 \text{ \AA}$  and potential well  $\epsilon = 0.6502 \text{ kJ/mole}$ . We tested as intramolecular potential an harmonic one, with the parameters used in reference [7]. The intramolecular potential is written

$$V(q) = \frac{1}{2}k_r \sum_i \sum_{j=1}^2 (r_i^{(j)} - r^{(0)})^2 + \frac{1}{2}k_\theta \sum_i (\theta_i - \theta^{(0)})^2 \quad (11)$$

where  $\mathbf{r}^{(j)} = |\overline{OH_{(j)}}|$  and  $\theta = \widehat{HOH}$ , with  $r^{(0)} = 1.0 \text{ \AA}$  and  $\theta^{(0)} = 109.47^\circ$  are the SPC equilibrium values. The constants  $k_r$  and  $k_\theta$  have been chosen equal respectively to  $4.637 \cdot 10^5 \text{ kJ/mole nm}^2$  and  $383 \text{ kJ/mole rad}^2$ . The contribution of the term depending on the angle bending is spread over the atoms using formulas (12-14) in [10].

It must be noted that a refinement of the model is necessary, because SPC has been calibrated for a rigid situation while we simply superimposed the intramolecular interactions on top of it. An example of model which takes into account flexibility in a rigorous *ab initio* approach is given in [11]. However the necessity of taking polarizability into account as well in a self-consistent way brought us to demonstrate the MTS algorithm on a simpler model, because the accuracy of the water model used is beyond the scope of the present paper.

Note that the MTS algorithm described in [12] (and used by the authors in [7]) is neither area preserving nor time reversible and therefore suffers from the problem of energy drift. Those authors introduce a splitting not only in the forces,

$$\ddot{q}_i = a_i = a_{i,f} + a_{i,s} \quad (12)$$

but also on the coordinates, so that

$$q_i = q_{i,f} + q_{i,s} \quad (13)$$

and therefore

$$\begin{aligned} \ddot{q}_{i,f} &= a_{i,f} \\ \ddot{q}_{i,s} &= a_{i,s}. \end{aligned} \quad (14)$$

At each time step the dynamics is evaluated  $n$  times for  $q_{i,f}$  and  $p_{i,f}$ , leaving  $q_{i,s}$  and  $p_{i,s}$  unchanged, then the correction for the “slow” terms is introduced. Translated into the formalism of the previous paragraph, the evolution operator for a time step is the following:

$$\begin{aligned} \exp[\Delta t \mathcal{L}(\mathcal{H})] &\cong \exp[\Delta t \mathcal{L}(V_s(q))] \times \exp[\Delta t \mathcal{L}(\frac{1}{2} \sum_i \frac{p_{i,s}^2}{m_i})] \times \\ &\left[ \exp[\frac{\Delta t}{2n} \mathcal{L}(V_f(q))] \exp[\frac{\Delta t}{n} \mathcal{L}(\frac{1}{2} \sum_i \frac{p_{i,f}^2}{m_i})] \exp[\frac{\Delta t}{2n} \mathcal{L}(V_f(q))] \right]^n \end{aligned} \quad (15)$$

This expression is different from Eq.8 in many respects. First of all, it lacks reversibility, secondly the mixed term containing  $\mathcal{L}(\sum_i p_{i,f} p_{i,s})$  is absent. A poor energy conservation is indeed observed, and the authors suggest swapping  $q_{i,s}$  and  $q_{i,f}$  from time to time to ease the problem [12].

### 3 Parallel implementation

Molecular Dynamics is a typical example of the N-body problem. When the interactions are long-range (which is the case in charged systems like water) one must consider all the  $N(N - 1)/2$  pairs of particles (discarding multiple interactions). Leaving aside known methods to handle the  $O(N^2)$  interactions indirectly, such as multipole expansion and particle-mesh, we are left with a particle-particle treatment. This class of problems displays an intrinsic data-parallelism [13] that can be exploited differently (in the framework of a MPP computational environment) according to the ratio between the number of particles  $N$  and the number of (virtual) processors  $p$  available:

1.  $p \sim O(N^2) \Rightarrow$  one processor per binary interaction;
2.  $p \sim O(N) \Rightarrow$  one processor per particle;

3.  $p \ll N \Rightarrow$  one processor per spatial cell.

In the Fortran 90 environment the solution for case 1 is to use a `spread` of the status array of the  $N$  particles in order to calculate all forces concurrently, but for the number of particles of interest this approach is unfeasible. Case 3 corresponds to a coarse-grained parallelism with few processors; it is feasible only for short-range forces, otherwise the efficiency is severely limited by the communications. Case 2 requires ordering of the particle status in a ring which is `cshifted`  $N - 1$  times in order to pass the status to every other particle, accumulating the forces acting onto each particle (the last `cshift` would correspond to self-interaction, therefore it is skipped); this technique is also called “digital orrery” in the Thinking Machine parlance [3].

In view of the target number of molecules to model, we have chosen the second approach, assigning a molecule to each processor. Therefore the time required by the intermolecular “slow” force calculation scales as  $N$ , including the Ewald sum necessary to add up the contributions of the particles in the infinite replicas of the basic periodic cell. Using Newton’s third law it is possible to cut the  $(N-1)$  `cshift` to  $(N/2)$ , keeping track of the accumulated forces on an auxiliary array `cshifted` positions. On the other hand the

intramolecular “fast” forces involve only calculations between atoms of the same molecule, so this term is immediately parallelised along the molecules and scale as a constant.

We have run our code mostly on a Connection Machine 8k CM-200. Due to the heavy computational burden of the long-range forces, mainly the exp function (which arise from the Ewald sum treatment of the coulombic terms), we have found that when the number of molecules is significant (some hundred to thousands), most of the CPU time is spent for the “slow” forces calculation, while the communications among processors take approximately 20% of the total and the “fast” forces a mere fraction of percent. Therefore the present class of simulations is computationally bound. Table I shows the detailed breakdown of the CM busy time (as measured by CM-timer library calls) for one time step, for a system of 512 molecules and 178 Ewald sum terms.

Fig. 1 shows the scaling of the front-end time (in ms) for one time step, divided by the number of molecules  $N$ , as a function of  $N$ , in log-log plot. For comparison, a similar code in Fortran 77 have been run on a workstation IBM RISC/6000 560. While the serial machine time per molecule scales always as  $N$ , the CM time reaches an optimum for  $N$  equal to 4 times the number



of physical processor available (in our case, 256 floating point coprocessors). When  $N$  increases, finally the time required scales again as  $N$  because each processor has to solve an  $N$ -body problem serially. We have achieved speeds around 5 ms/step/molecule for  $N$  up to 4096.

We have also found convenient to calculate most of the physical quantities of interest “on the fly” rather than producing large files of raw positions and velocities. An example of this is given by the radial pair correlations functions, which need the information on particle distances that comes from the “slow” forces evaluation. Routines are then available to map the parallel array of distances efficiently on a serial array of bins to be histogrammed at the end of the simulation.

## 4 Results

Thermalization of systems at different time-scales is a well known physical problem in itself. In flexible models of water the vibrational kinetic energy is transferred relatively quickly to the translational degrees of freedom (few tenths of ps for a periodic box of 512 molecules), whereas the exchange in the opposite direction is much slower. We start our simulation with zero velocity

of the molecule center-of-mass and a suitable displacement of the atoms from their equilibrium position to give an initial non-zero vibrational energy. If special tricks were not used, tens of ps would be wasted just for equipartition of the kinetic energy (see Fig. 2a, where the translational and the roto-vibrational temperatures are shown for a free run). The order parameter of the starting, regular configuration relax indeed in less than 1 ps. We have found that the Maxwellian shuffle technique thermalise each degree of freedom at room temperature within 2 ps only (see Fig. 2b). This technique consists in updating the velocities of all atoms with a Maxwell-Boltzmann distribution at the desired temperature every few time steps [14, 15] (in our case, every 100 time steps of 1 fs). This corresponds to putting each particle instantaneously in contact with a heat bath, to force ergodicity of the N-body system.

The single most important result presented in this paper is the accuracy of the energy conservation of the algorithm. We have calculated a relative drift  $(1/E_{tot})(dE_{tot}/dt)$  of  $(2.3 \pm 0.1) \cdot 10^{-6} ps^{-1}$  (which means 0.2 K only over 100 ps) over a period of 70000 time steps, for a simulation of 512 molecules including 178 Ewald terms, at  $297 \pm 5$  K. This is to be compared with [7] who have a drift of  $9.3 \cdot 10^{-4} ps^{-1}$ , i.e. 400 times higher, on a system of 216

molecules on a comparable number of steps. The time steps used were  $\Delta t = 1$  fs for the “slow” and  $\Delta t_f = 0.1$  fs for the “fast” forces, which gives a fluctuation of the total energy of approximately 1% of the fluctuation in the total potential energy, as commonly required [16]. Fig. 3 shows the total energy fluctuation observed in our system of 512 molecules at 300 K as a function of  $\Delta t$  in log–log plot. The data is fitted reasonably well by the predicted  $O(\Delta t^2)$  accuracy.

Adding the inner degrees of freedom in the water model induces some interesting effect already pointed out in [7]. We also observe a stretching of the OH bond length by  $0.015 \text{ \AA}$  at 300 K and a concomitant decrease of the HOH bond angle by  $4^\circ$ , which can be explained by the formation of water dimers through hydrogen bonds. The dynamics of the intramolecular motion is given by the cosine transform of the velocity autocorrelation function, which gives the peaks at  $1620 \text{ cm}^{-1}$  and  $3760 \text{ cm}^{-1}$ , corresponding respectively to the angle and bond length vibrations in the case of harmonic intramolecular potential.

The intermolecular structure is described traditionally by the radial pair correlation functions, which we show in Fig. 4. We have also used 3D visualization tools to study 3D pair correlation functions for Oxygen and Hydrogen,

getting a more vivid understanding of the structure of the liquid water at least in the first coordination shell. To obtain this 3D functions, we transform the coordinates of each molecule in the local frame of each other. The frame is defined by the vectors  $H_1\vec{H}_2$ ,  $O\vec{H}_1 + O\vec{H}_2$  and the one orthogonal to the HOH plane. The frequency of occurrence of any triplet of “local” coordinates is binned in cubes. Fig. 5 shows a perspective view of a representative isosurface of the 3D scalar field of the oxygen (darker) and hydrogen (lighter) atoms in the first coordination shell of the central, reference molecule. While a general tetrahedral order is apparent, a remarkable deviation from it is given by the fact that the hydrogen bonds with the reference oxygen atom are placed on a large angle arc (almost a semicircle) behind which there is an arc of oxygen atoms. On the other hand, the oxygen atoms bonded to the reference hydrogen atoms are placed on a cone with a narrower aperture, whose axis is approximately given by the OH direction.

Finally, as an example of the transport property of the system, we have measured diffusivity both through the Einstein and the Kubo relationships, obtaining values around  $2 \cdot 10^{-9} m^2/s$  at 300 K.

## 5 Conclusions

In this paper, we have shown that one can exploit the presence of separate time-scales to simulate flexible polyatomic molecules in a straightforward and efficient way, by using a Multiple Time Scale algorithm. We have demonstrated good energy conservation for a real, important system, i.e. water, spending the same CPU time as for conventional rigid models. MTS allows a reduction of the computation time over STS by exploiting the different numerical complexity of the intra- and inter- molecular interactions.

It is well known that in some cases rigid models can suffer either from the complexity of modelling the intramolecular bonds (rigid body motion approach), or from problems in the convergence of SHAKE (method of constraint approach) for some classes of molecules which have a singular matrix for the bonds between atoms [17]. Therefore we maintain that, wherever possible, a flexible model treated with MTS should be considered instead, which is far easier to implement. This also ensures good load balancing among processors in a MPP computer, while SHAKE performance is bounded to the worst case, being an iterative scheme. Finally, it is true that “hard” degrees of freedom are difficult to thermalise [1], but there are suitable and

reliable tools like Maxwellian shuffling, to circumvent this difficulty, as we have shown.

In conclusion, we have shown that a proper MTS approach is an effective method in the Molecular Dynamics simulation of water and other polyatomic molecules, allowing a natural move to parallel, efficient code for running on MPP machines.

## **Acknowledgments**

We are pleased to acknowledge the work of Paolo Marenzoni for porting the Flexible Water code to CM Fortran. We also thank Ernesto Bonomi and Francesco Sciortino for valuable discussions and suggestions, and Alan Scheinine for improvements to the manuscript.

This work has been carried out with the support of the Sardinian government.

## References

- [1] G. Ciccotti and J.-P. Ryckaert, “Molecular dynamics simulation of rigid molecules”, *Computer Phys. Rep.*, **4**, 345–392 (1986).
- [2] W. F. van Gunsteren and H. J. C. Berendsen, *Molec. Phys.*, **34**, 1311 (1977).
- [3] B. G. J. P. T. Murray, P. A. Bash, and M. Karplus, “Molecular dynamics on the connection machine system”, Technical Report TMC-31 CB88-3, Thinking Machines Corporation, 1988.
- [4] J.C. Sexton and D.H. Weingarten, “Hamiltonian evolution for the hybrid monte carlo algorithm”, Technical Report RC 17668, IBM Research Division, 1992.
- [5] M.E. Tuckerman, B.J. Berne, and G. J. Martyna, “Reversible multiple time scale molecular dynamics”, *J. Chemical Phys.*, **97**, 1990–2001 (1992).
- [6] A. Rahman and F. H. Stillinger, “Molecular dynamics study of liquid water”, *J. Chemical Phys.*, **55**, 3336–3359 (1971).

- [7] Anders Wallqvist and Olle Teleman, “Properties of flexible water models”, *Molec. Phys.*, **74**, 515–533 (1991).
- [8] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Clarendon Press, 1987.
- [9] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, and J. Hermans, in B. Pullman (ed.), *Intermolecular Forces*, Reidel, 1981, p.331.
- [10] J. H. Dunn, S. G. Lambrakos, P. G. Moore, and M. Nagumo, “An algorithm for calculating intramolecular angle-dependent forces on vector computers”, *J. Comput. Phys.*, **100**, 17–24 (1992).
- [11] G. Corongiu, “Molecular dynamics simulations for liquid water using a polarizable and flexible potential”, *Int. J. Quantum Chem.*, **42**, 1209–1235 (1992).
- [12] O. Teleman and B. Jonsson, “Vectorizing a general purpose molecular dynamics simulation program”, *J. Comput. Chem.*, **7**, 59 (1986).
- [13] Ernesto Bonomi and Marco Tomassini, “Massively data-parallel molecular dynamics”, *Int. J. Modern Phys. C*, **3**, 709–731 (1992).



- [14] Ernesto Bonomi, “Molecular dynamical simulation of the canonical ensemble”, *J. Stat. Phys.*, **39**, 167–180 (1985).
- [15] J. L. Fattebert and E. Bonomi, “Isothermal molecular dynamics: a practical study”, *Int. J. Modern Phys. C*, **4** (1993).
- [16] Dieter W. Heermann, *Computer Simulation Methods in Theoretical Physics*, Springer-Verlag (1986).
- [17] Kenton D. Hammonds and Jean-Paul Ryckaert, “On the convergence of the SHAKE algorithm”, *Computer Phys. Comm.*, **62**, 336–351 (1991).

## Figure captions

1. Log-log plot of CPU time per step per molecule (ms) vs number of molecules  $N$  for SPC-flexible model of water running on a workstation IBM RISC/6000 (open circles) and on a Connection Machine 8k CM-200 (full circles). While the serial code time per molecule scales as  $N$ , the parallel code reaches an optimum dependent on the number of physical processors, then starts to scale as  $N$ , with a much smaller factor compared to the workstation.
2. **a:** Time plot of the translational (dotted line), roto-vibrational (dashed line) and global temperatures (solid line) for 2 ps of simulation of 512 molecules starting from an ordered configuration and zero velocity;  
**b:** same as **a** with the Maxwellian shuffle each 0.1 ps from  $t = 0.5$  ps.
3. Log-log plot of the total energy fluctuations in J/mole vs “slow” time step  $\Delta t$  in fs (stars). The line shows the best fit which has slope  $2.3 \pm 0.2$ . The “fast” time step was fixed at 0.1 fs. The observations were made on configurations of 512 molecules equilibrated at 300 K, followed for 10000 steps (apart from case  $\Delta t = 1.0$  fs, followed for 70000

steps).

4. Radial pair correlation functions  $g_{OO}(r)$  (solid line),  $g_{OH}(r)$  (dashed line) and  $g_{HH}(r)$  (dotted line) vs  $r$  (nm) for a system of 512 molecules equilibrated at 300 K,  $\Delta t = 1.0$  fs,  $\Delta t_{fast} = 0.1$  fs, followed for 70 ps; average over 63 time frames. The  $g(r)$  are shifted by one vertical unit for the sake of clarity. The “self” contribution is included for  $g_{OH}(r)$  and  $g_{HH}(r)$ ; its spread is due to the intermolecular vibration.
5. Perspective view of an isosurface of the full 3D pair correlation functions for Oxygen (darker) and Hydrogen (lighter). Reference frame  $\vec{x} = \vec{H}_1\vec{H}_2/|\vec{H}_1\vec{H}_2|$ ,  $\vec{y} = (\vec{O}\vec{H}_1 - \vec{O}\vec{H}_2)/|\vec{O}\vec{H}_1 - \vec{O}\vec{H}_2|$  and  $\vec{z} = \vec{x} \times \vec{y}$ . Box sizes are equal to 0.8 nm.