New Gromacs Implementations for Multiscaling Space MD

NICOLAE GOGA[1]*, STEFANIA COSTACHE[2], SIEWERT MARRINK[1]

[1]University of Groningen, Faculty of Natural Science, Nijenborgh 4, 9747 AG Groningen, The Netherlands

[2] Politehnica University of Bucharest, Faculty of Automatic Control and Computers, 313 Splaiul Independentei, Bucharest, 060042, Romania

Among the methods used nowadays for studying the microscopic properties of polymers, proteins, membranes and other bio-materials is the molecular dynamics. One of the widely used software in molecular dynamics is Gromacs developed at the University of Groningen. From a computational perspective, molecular dynamics requires large computational power and an increased storage capacity. Different physical models were recently defined for trying to reduce the complexity and to make more efficient the computational it molecular models. In this paper we present the Gromacs implementation of a new multiscaling MD model with a space dependent multiscaling parameter and we discuss efficiency measurements.

Keywords: polymers, molecular dynamics, Gromacs, multiscaling modeling

Molecular Dynamics is a subdiscipline of Computational Chemistry that focuses on the description and simulation of chemical systems consisting of atoms and molecules. The simulations are used as a kind of virtual experiments, to complement laboratory experiments and to help understanding the underlying chemical processes. Some application areas are: studying the properties of plastic materials [1-3, 9, 12-14], unraveling of protein structures and the design of new materials and pharmaceutical drugs.

Molecular dynamics requires large computational resources [15], typical runs taking a prolonged amount of dedicated multi-processor cluster time, from months to years. Thus, for a typical simulation, the time-scales on which microscopic properties of the materials and cell components are simulated and studied do not match the natural time-scales, being much lower than them. In this respect, sometimes, molecular dynamic simulations are not always realistic in their predictions.

To counteract these restrictions, different physical molecular models were recently developed and used. One common modality nowadays is to use coarse-grained atomistic models, in which an assembly of atoms is replaced with one particle, usually placed in the center of mass of the assembly of correspondent particles. Although this approximation enlarges the time-scales up to an order of 20 and reduces the computational complexity, the disadvantage is that some of the fine-detailed properties are not observed when using coarse graining. One possibility to improve the results obtained from such a simulation is to combine the two models in one multiscaling simulation.

In [6] we presented an implementation of a new multiscaling simulation, where the multiscaling factor λ was constant during the simulations. In this paper we present the Gromacs implementation of a new multiscaling method in which the multiscaling factor λ is space-dependent, in the sense that it depends on the position of the particle. This model is developed at the University of Groningen.

Before going further we will say some words about the software molecular dynamics package GROMACS. GROMACS (GROningen MAchine for Chemical Simulations[1]) is an extensive, well-established and free software package used in Molecular Dynamics simulations, and one of the first of its kind. It has been developed in the early 1990s at Groningen University; the last version GROMACS 4.0.3 contains a high degree of parallelism. GROMACS is a product extensively used in the academic medium (more than 150 universities and research institutions) as well as in industry. It is also known that it achieves a single-processor performance superior to any other similar software [4]. We will detail our implementation in a subsequent section about Gromacs.

One should note that we are not the first in trying to define a multiscaling model that is space-dependent. There are some other efforts in this area but each one of them has different short-comings. The model developed by [10] applies only to two-particle interactions and it is not extended to three or four particle-type interactions. Other proposed models [7] are pure theoretical, without a presentation of any implementation of it.

This paper is organized as follows. The next sections present briefly the theoretical ingredients of the model and its Gromacs implementation. After that efficiency measurements are discussed and at the end we draw the conclusions.

Experimental part

Once we come to the physical model that underlines space dependent multiscaling, one should note that it has similarities with the approach in which the multiscaling factor is constant, approach that was discussed to some degree in [6]. Therefore we will try just to discuss the elements that are specific to space-multiscaling and to refer to the other paper for the common-parts.

Nevertheless one should note that the complexity of space dependent multi scaling is larger than in the case of a constant factor because when deriving the Newtonian equations of motion from the Hamiltonian the multiscaling factor should be taken into account in a larger proportion than for the constant case.

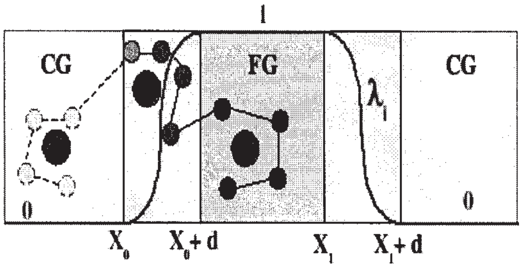


Fig.1 A multiscaling space representation for FG and CG particles

* *e-mail: n.goga@rug.nl; Tel: (+31)-64-3634338*

Before presenting the equations of motion, we will explain in more detail the space-lambda model. This is illustrated in the figure 1. There is presented a typical case of a multiscaling molecular dynamics. The middle region is the one in which the fine-grained detail of interest are computed. The fine grained portion is surrounded by two coarse grained regions in which the computations are done only at the coarse grain level. Between fine and coarse grain parts there are two transitions regions in which the particles start to change from fine grain to the coarse grain nature. For modeling the transitions between the two regions, a multiscaling space depending factor λ is taken into account. In the figure it can be seen that λ has the value 0 in the CG region, 1 in the FG region and between 0 and 1 in the transitions regions.

Movement of the particles

As we did in [6], the movement of the particles is represented in generalized coordinates that are given by the following equations of motions:

$$(R_i, s_{ik}^{FG} = r_{ik}^{FG} - R_i^{CG}) \tag{1}$$

and the constraints

$$\sum_{ik} m_{ik} * s_{ik} = 0 \tag{2}$$

where m_{ik}^{FG} represents the masses of the FG particles and the number of constraints equals the number of CG particles multiplied with 3 (the number of coordinates (x, y, z)).

The equation of motions starts from the Hamiltonian. Without losing of generality in what follows we will discuss the situation of the interaction potentials between two particles A and B. The interactions between 3 particles (angle potentials) and four particles (dihedral potentials) are solved in a similar way as presented for two particles. The integration of energies is done in the following way:

$$\begin{aligned} U_{mult} = {} & \lambda\ (R_a) * \lambda(R_b) * U^{FG}(r_{ik\_a}^{FG}, r_{ik\_b}^{FG}) + \\ & (1 - \lambda(R_a) * \lambda(R_b)) * U^{CG}(R_{i\_a}^{CG}, R_{i\_b}^{CG}) + \\ & (1 - \lambda(R_a) * \lambda(R_b) * Const \end{aligned} \tag{3}$$

One should note that as compared with the case of constant lambda [6] in these equations there is an extra term present, which is the constant *Const*. We did not explicit mention in [6], although it was also correct to put it there, because when deriving the forces, *Const* and λ being constant they were not showing up in the equations of forces that were used further for the computation of the movements. In the case of space multiscaling this constant that corresponds to the difference in chemical potentials cannot be ignored because it will show up further in the equations of motions (see bellow). The disadvantage of this constant is the fact that there should be parameterization work done for establishing the value of the constant for different types of molecular systems. An alternative approach could be to use a special potential in the transitions zones, but this implies also parameterization.

The computation of the forces for CG and FG particles are given by:

$$\begin{aligned} F_{mult}^{CG} &= \frac{\partial U_{mult}}{\partial R_i^{CG}} \\ F_{mult}^{FG} &= \frac{\partial U_{mult}}{\partial s_{ik}^{FG}} \end{aligned} \tag{4}$$

From (4) it results that the formulas of the force for particle A (similar to particle B) are computed in the following way:

$$\begin{aligned} F_{mul\_a}^{CG} = {} & \lambda(R_a) * \lambda(R_b) * \sum_{ia} F^{FG}(R_a + s_{ia}) + \\ & (1 - \lambda(R_a) * \lambda(R_b)) * F^{CG}(R_a) + \\ & \frac{\partial \lambda(R_a)}{\partial R_a} * \lambda(R_b) * (U_{mul}^{CG} - U_{mul}^{FG} + Const) \\ F_{mul\_ia}^{FG} = {} & \lambda(R_a) * \lambda(R_b) * F^{FG}(R_a + s_{ia}) \end{aligned} \tag{5}$$

The equations of movement are solved using a Verlet algorithm and constraint solving as discussed also in [6].

Temperature coupling

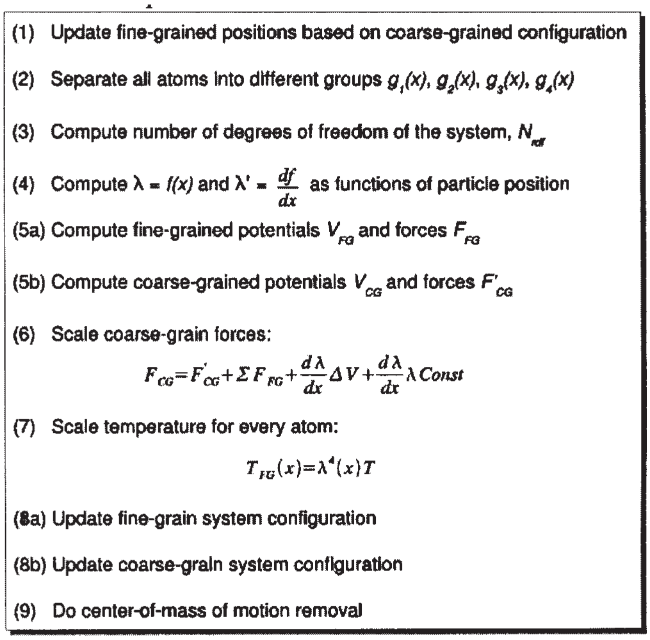
As for constant lambda [6], the temperature is scaled with the parameter λ for the relative fine grained details and kept constant at the given reference temperature for the CG system. This way, in the CG regions the FG temperature is *0* that assure also no relative velocities (the FG details are frozen: the forces are also *0*), in the FG region the FG temperature gets the full value (full FG details) and in the transitions regions the temperature is between *0* and full value, making the FG details to be introduced smoothly into the system.

Implementation in Gromacs 4.0

The implementation of the multiscaling model presented in the previous Section is based on the present structure of the GROMACS 4 main MD simulator. We used the existing topology and configuration of the simulated system, and the parallel and scientific algorithms available in GROMACS [8]. To implement our model, a well suited approach would be to modify the related data structures and functions for supporting a second topology and configuration, making thus possible the existence of fine-grain and coarse-grain descriptions for the same molecular system. However, the existing structure of the application is not entirely modular and implementing our desired algorithm would have imposed a substantial change in the code, with extensive and complex modifications. This would have interfered with constrains arisen from the structure of the program and its data, and would have made the model testing more difficult. Therefore, to allow different configurations for the fine-grain and coarse-grain subsystems, and to analyze the macroscopic properties of both representations, we chose an implementation based on different simulation spaces, leading to two simultaneous simulations. Also, to analyze the properties of the simulated subsystems, we kept the finegrain to coarsegrain mapping, together with the topology, for the complete simulation box.

The main computational steps are described in the figure above. In step (1) the finegrained coordinates and velocities are constrained. Because the λ parameter is now a function dependent of particle position, we must compute it at each MD step, for every atom from the system. Keeping in mind that a CG particle is composed of many FG atoms, it is enough to compute the λ parameter only for the CG particles. The surrounding FG atoms will have the same value for λ . When using multiple processors for an MD simulation, these computed values must be distributed to the neighbor nodes due to implementation of coarse-grained particles by using virtual interaction sites. This is required by the force computation steps (5a) and (5b).

The force calculation for the space λ model must be done in a different way than before, because the mixing



```
(1)  Update fine-grained positions based on coarse-grained configuration
(2)  Separate all atoms into different groups $g_1(x)$, $g_2(x)$, $g_3(x)$, $g_4(x)$
(3)  Compute number of degrees of freedom of the system, $N_{rdf}$
(4)  Compute $\lambda = f(x)$ and $\lambda' = \frac{df}{dx}$ as functions of particle position
(5a) Compute fine-grained potentials $V_{FG}$ and forces $F_{FG}$
(5b) Compute coarse-grained potentials $V_{CG}$ and forces $F'_{CG}$
(6)  Scale coarse-grain forces:
         $F_{CG} = F'_{CG} + \Sigma F_{FG} + \frac{d\lambda}{dx}\Delta V + \frac{d\lambda}{dx}\lambda Const$
(7)  Scale temperature for every atom:
         $T_{FG}(x) = \lambda^4(x) T$
(8a) Update fine-grain system configuration
(8b) Update coarse-grain system configuration
(9)  Do center-of-mass of motion removal
```

parameter is position dependent. Usually, in the forces computation both bonded and non-bonded interactions bring their contributions. Following this model, we excluded the non-bonded fine grain interactions from the coarse-grain region when building the neighbour lists. For the bonded interactions, we avoided the unnecessary computations in the cases in which the forces will be zero. For the computation of non-bonded interactions, the modifications were added to the C version of the non-bonded kernels, using the existing kernel generator (a tool that is used to generate the C code for the dozens of kernels existent in GROMACS). This generator was modified to create separate versions of kernels for the space- λ force computation.

After force calculation, the update of the system configuration (coordinates, velocities, temperature coupling) is done. For the space-λ model, the temperature is not scaled with a constant factor, but is a function dependent on the atom coordinates. Therefore, we modified the original update algorithm from step (8), to correct the temperature scaling for the fine grain subsystem.

To keep the simulated fine-grain system more stable, the atoms from the FG topology were distributed into four temperature groups according to their position, or the character of the region in which they were included. These groups must be rebuilt at each step, because the temperature groups are related to spatial regions. In the original implementation, the information related to system topology (group, masses, etc.) is initialized and the number of degrees of freedom is computed only once. However, in our case, this number fluctuates, as atoms move between the four regions. To correctly update the groups for all the atoms from the system, they are computed at each step. If parallelization is performed, this computation will be made only for the home atoms. After that, the information will be gathered and broadcasted, so each node will know the groups for the entire system.

Results and discussions

In this section we will discuss the performance impact and the scaling of our multiscale algorithm. The measurements were performed on an IBM Blue Gene/L architecture with all the simulations ran in the Co-processor mode. We simulated a system of 750 HD molecules in a box of 15x4x8 nm. The fine grain part of the system was centrated and 3nm wide, with a transition buffer of 3nm wide also.

The algorithm implies the existence of two simulation spaces, with exchange of information between them, when needed (e.g. forces, positions). The number of nodes implied in the simulation of each box (fine grained and coarse grained) can be parameterized by the user. For choosing the appropriate distribution of processors, we have taken in consideration the design of the coarse grain model that we used.

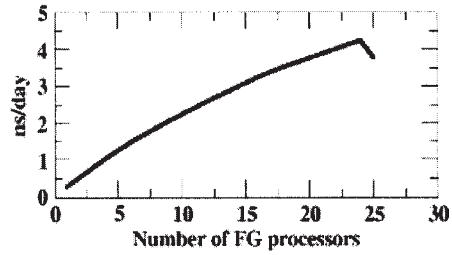


Fig. 2. Performance function of the number of FG processors

The results obtained regarding the correct distribution and the scaling of the algorithm can be seen in table 1. In this case, the most appropriate the fine-grain/coarse-grain ratio is 6:1, meaning that for one processor that handles the coarse grain simulation, six processors are allocated for the fine grain part. However, even with a proper ratio, having two simultaneous leads to an overhead imposed by the global communication between the two simulations and the two decomposition domains for the same box.

Table 1
TIMINGS OBTAINED FOR SPACE-MULTISCALING WITH TWO TOPOLOGIES

Nr proc fg	Nr proc cg	ns/day
1	1	0.28
4	1	1.02
6	1	1.47
8	2	1.86
12	2	2.57
12	3	2.58
18	3	3.51
16	4	3.23
24	4	4.23
25	5	3.76

The performance of the algorithm is close to a linear scaling at the beginning, but as the number of processors increases, we can notice degradation due to the increased volume of communicated data. But considering the additional computation and communication steps required by our multiscale simulation, this is quite expected. Also, the domain decomposition algorithm from Gromacs does not perform well in our particular case, due to the fact that the distribution of particles between the processors is still done for the original box of simulation.

Conclusions

We presented the implementation of the multiscale space- λ in Gromacs 4, a well-known molecular dynamics package. For analyzing better the properties of the simulated systems and the correctness of our algorithm we kept the fine grained particles for the entire box of simulation. As a future work we plan to unify the two domains of simulation (fine grain and coarse-grain) into one that can be split seamlessly on all the processors. In this way we will avoid the overhead of communication between the two spaces of simulation. Also, we are considering modifying the original domain decomposition algorithm for taking in consideration a better redistribution of atoms between processors in our particular case. As a large part of the box is at a coarse grain level, fewer computations are being done and so, the load-balancing between the processors is not fair in this moment.

References

1.BHARADWAJ R.K., BERRY R.J., FARMER B.L., POLYMER, 41, 19, 2000 , p. 720

2.BOLCU D., STANESCU, M.M CIUCA, I., JIGA, G.,GHEORGHIU, H., IANCULUI, D., Mat. Plast., **45**, nr. 2, 2008, p. 137

3.BOWERS, J. K., CHOW, E., XU, H., DROR, R.O., EASTWOOD, M. P., GREGERSEN, B. A., KLEPEIS, J. L., KOLOSSVARY, I., MORAES, M. A., SACERDOTI, F. D., SALMON, J. K., SHAN, Y., SHAW, D. E., Scalable algorithms for molecular dynamics simulations on commodity clusters. In SC'06: Proceedings of the 2006 ACM/IEEE conference on Supercomputing, 2006, p. 84

4.BULACU, M. Molecular dynamics studies of polymer chains. Phd thesis, University of Groningen, 2007

5. GOGA, N., COSTACHE S., MARRINK, S. A multiscaling constant lambda molecular dynamics Gromacs implementation, Mat. Plast., **46**, nr. 1, 2009

6. HEYDEN A., Truhlar D.G. , J. Chem. Theory Comput. 4, 217 (2008).

7. *** Gromacs manual, available online at www.gromacs.org

8. PAUN, V.-P., Mat. Plast. , **43**, nr.1, 2006. p.57

9. PRAPOTNIK M., SITE L.D., KREMER K., J. Comput. Chem. 123, 224106 (2005)

10. SMITH, J.S., BEDROV, D., SMITH D., Composites Science and Technology, 63, 11, 2003, p.1599

11. PAUN, V.-P., Mat. Plast., **40**, nr.1, 2003. p.25

12. YANG,H., SHENG, L.S., QIAN, H., YANG, Y., ZHANG, X., SUN, C., Polymer, **45**, 2, 2004, p. 453

13. YU, K., LI , Z., SUN, J. Polymer Structures and Glass Transition, Macromolecular Theory and Simulations, 10, 6, 2001, p. 624

14. PAUN, V.-P., Mat. Plast., **44**, nr.4, 2007. p.393

Manuscript received: 25.05.2009