Molecular Simulation of Fluid Flow on a Cluster of Workstations

Martin Bernreuther
Martin.Bernreuther@ipvs.uni-stuttgart.de
Institut für Parallele und Verteilte Systeme, Abteilung Simulation großer Systeme
Universität Stuttgart, Universitätsstr. 38, 70569 Stuttgart

Hans-Joachim Bungartz
bungartz@in.tum.de
Institut für Informatik
Technische Universität München, Boltzmannstr. 3, 85748 Garching

Abstract

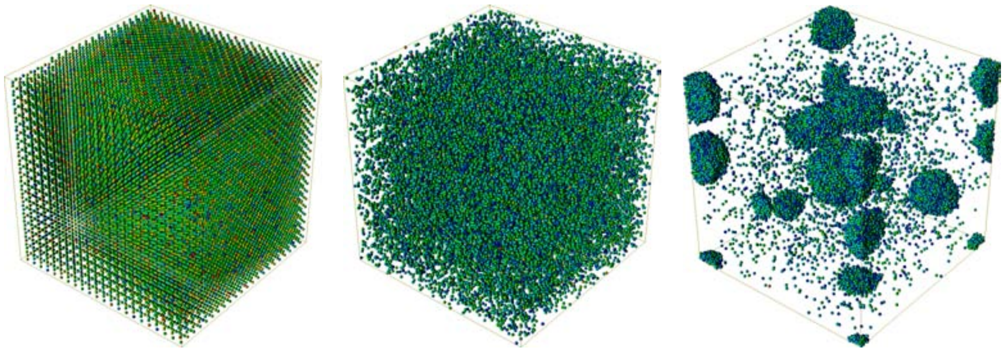
Simulation of fluid properties and flow below a certain length scale, where the continuum assumption does not hold any more, has to be done on a molecular level. Molecular Dynamics (MD) is a proper tool for nanofluidics. The limits of the system sizes manageable today are pushed not only by advances and availability of new hardware. It's even more important to achieve enhancements in the development of fast efficient algorithms and hardware optimized implementations. High Performance Computing systems and especially Clusters of Workstations, which turn out to be very well suited for this task, are the primary target platform for the majority of MD codes today. After a classification of the flow type addressed here, implementation details and parallelization strategies will be discussed for MD simulations based on short-range potentials, suitable for a rich variety of components.

1 Introduction

Most people typically associate the numerical simulation of fluid flow with the Navier-Stokes (NS) equations, which govern the computational fluid dynamics (CFD) world and are derived in the framework of continuum mechanics with a macroscopic Eulerian view on the matter. On a molecular scale the continuum hypothesis is not valid any more. The classical Molecular Dynamics (MD) approach [1, 2, 3] calculates the particle motion of molecules based on Newtonian mechanics with a Lagrangian view. Up to now, MD is only applicable for small domains and time scales due to the immense computational power needed. Fluid flow through nanochannels can be modeled using either a continuum [4] or a molecular approach. For microfluidics the Reynolds number is usually low and the resulting laminar flow is categorized as Stokes flow. This is typically accompanied by a small Péclet number, which reveals the relevance of the diffusion processes. However, if the length scale drops down a certain level in the nanoscale, the continuum approach won't give accurate results any more [5, 7, 6]. A neuralgic point is the boundary condition: the classical slip/no-slip is inaccurate to represent the momentum and temperature accomodation. Introducing a slip length will result in an intermediate behavior between slip and no-slip and lowers the limit to apply a continuum method. Hybrid methods [8] combining MD with a continuum method like the Lattice-Boltzmann method might be an answer to the shortcomings of both approaches.

2 Physical and mathematical basics

In contrast to a macroscopic simulation, a MD simulation has to model the whole phase space. The molecular velocities combine streaming and Brownian motion. It is not reasonable to trace a single molecule due to the Lyapunov instability [9]. MD calculations therefore target averages of ensembles, which provide macroscopic properties. Statistical mechanics provides the theoretical foundation to link molecular dynamics with continuum mechanics. The fundamental equation here is the Boltzmann equation, whereas the Chapman-Enskog procedure shows an equivalence to the Navier-Stokes equations under certain assumptions. It introduces the dimensionless Knudsen number as ratio of the mean free path of a molecule and a macroscopic reference length. This number also classifies the flow type, whereas exceeding a certain limit ($\approx 10^{-2}$) indicates a non-continuum flow.



(a) Initial configuration (fcc lattice), $t^* = 0$

(b) $t^* = 10$

(c) $t^* = 3000$

Figure 1: MD simulation of a nucleation process

Molecular simulation programs using the classical Lennard-Jones (LJ) 12-6 potential u to describe binary interactions between atoms usually implement a dimensionless form for the potential and the derived intermolecular force equation:

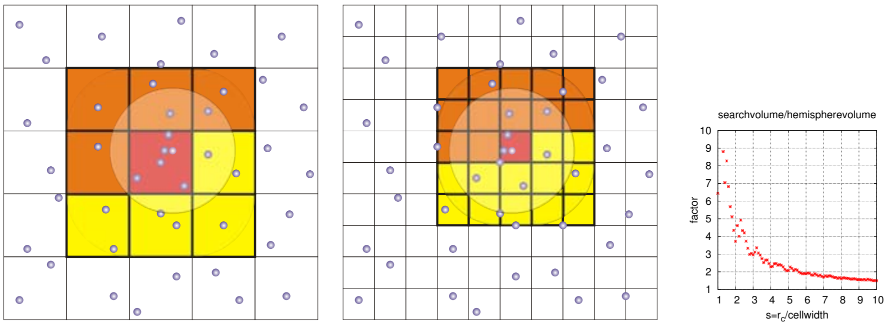
$$\vec{F}^*(r_{ij}) = \frac{-\nabla u(r_{ij})\sigma}{\epsilon} = 24\left(2\left({r_{ij}^*}^2\right)^{-6} - \left({r_{ij}^*}^2\right)^{-3}\right)\frac{\vec{r}_{ij}^*}{{r_{ij}^*}^2} \tag{1}$$

for a given length parameter σ and energy parameter ε with $r^* = r/\sigma$. For mixtures, the modified Lorentz-Berthelot combining rule is applied [10]. Fluids consisting of anisotropic molecules can be modeled by composites of several LJ centers. When polar fluids are considered, additionally polar sites with adequate potential have to be added. The molecular models are considered rigid here and therefore have no internal degrees of freedom. To calculate the interactions between two multicentered molecules, all interactions of the centers are summed up. Newton's equations of motion are solved numerically for N molecules over a period of time. These equations set up a system of ordinary differential equations. This initial value problem can be solved with a time integration scheme like the Velocity-Störmer-Verlet method. In the case of non-spherical molecules, an enhanced time integration procedure which also takes care of orientation and angular velocity is needed [11]. A thermostat controls the temperature, which is related to the adjusted velocities excluding flow.

Fig. 1 shows an example of a canonical ensemble of $N = 40000$ molecules in a domain of volume $V^* = 97^3$ with periodic boundary conditions at temperature $T^* = 0.7$. The observable nucleation process initiates a phase transition, which also has consequences on the runtime and load balance (cf. 1).

3 Algorithms and data structures

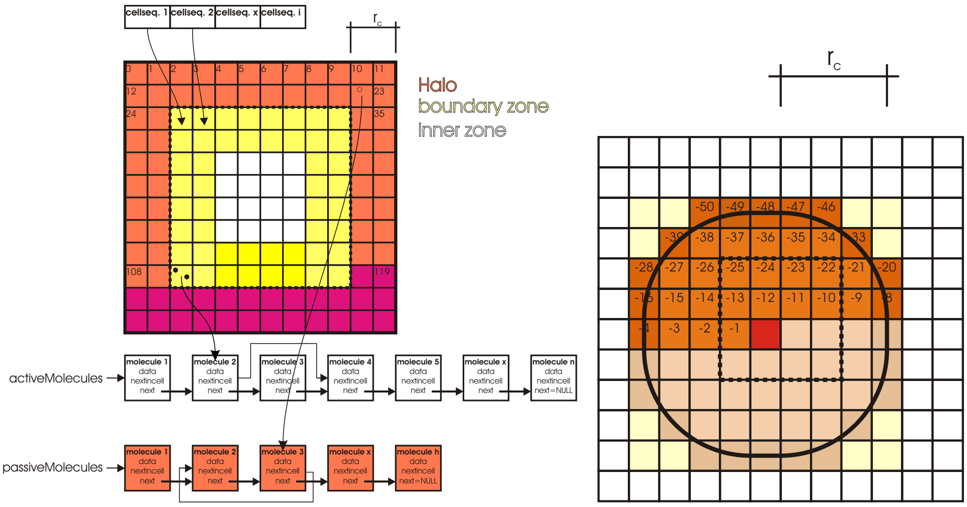
Assuming pairwise additivity, there are $\binom{N}{2} = \frac{1}{2}N(N-1)$ interactions for N molecules. Since LJ forces decay very fast with increasing distance, there are many small entries in the force matrix which may be neglected for distances $r > r_c$. Assuming an homogeneous molecular distribution, for this approximation the force matrix gets sparse with $\mathcal{O}(N)$ nonzero elements. The Link-Cell algorithm gains a linear run time complexity for these finite short-range potentials. The main idea is to decompose the domain into cuboid cells (cf. fig. 2) and to assign molecules to the cells they are located in. The classical implementation uses cells of width r_c (cf. fig. 2(a)). The cell interaction volume is the union of all spheres with radius r_c whose centers are located inside the cell. This is a superset of the union of interaction volumes for all molecules inside the cell. There is a direct volume representation



(a) $s = 1$ (b) $s = 2$ (c) examined volume

Figure 2: Link-Cell interaction volume

of the interaction volume, where the voxels correspond to the cells. This concept is generalized using cells of length r_c/s with $s \in \mathbb{R}^+$. The advantage is a higher flexibility and the possiblity to increase the resolution. For $s \to \infty$ the examined volume will converge to the optimal Euclidean sphere and for $s \in \mathbb{N}^+$ a local optimum is obtained (cf. fig. 2(c)). Cells are also used to calculate time averaged local densities and velocities out of cell ensembles to get an Eulerian view. The cell data structure used here (cf. fig. 3(a)) is comparable to a hash table where a molecule location dependent hash function maps each molecule to an array entry and hash collisions are handled by lists. All atoms are additionally kept in a separate list. The drawback using this data structure with large s is the increasing runtime



(a) Link-Cells

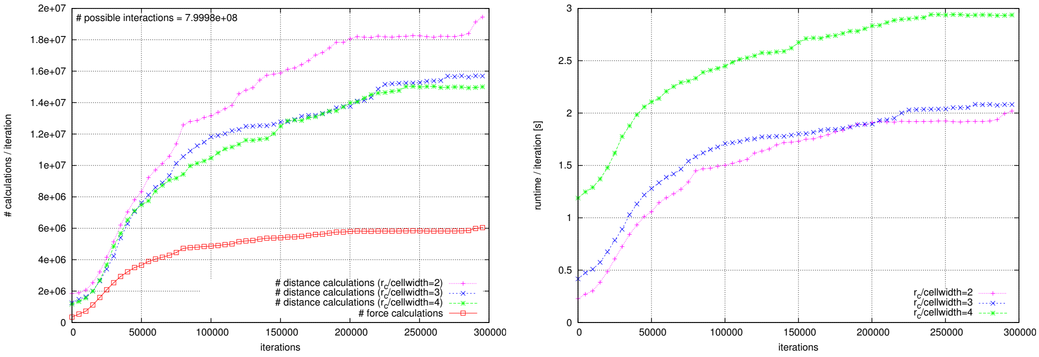
(b) offsets

Figure 3: Data structure

overhead, since a lot of empty cells have to be tested. In practice $s = 2$ is a good choice for fluids [6]. The implementation uses a one-dimensional array of pointers to molecules, which are heads of single linked intrusive lists. For multiple components, multiple lists are used. The domain is enlarged with a border halo-region of width r_c, which takes care of the periodic boundary condition for a sequential and contains virtual molecules of other processes for a parallel version.

Neighbor cells are determined with the help of an offset vector (cf. fig. 3(b)): the sum of the cell address and the offset leads to the neighbor cell address. The neighbor cell offsets are initialized once and cover only half of the cells' interaction volume to take advantage of Newton's third law (*actio* $=$ *reactio*). As a result neigbor cells considered and left out within this region are point symmetric to the cell itself.

The force calculations are done cell-wise considering the determined neighbor cells. The sequence order influences the cache performance, due to a temporal locality of the data. Using the adjoining cell as next cell, most of the interaction volume is overlapping the previous one and most of the molecules are reused. A vector containing the sequence of cells to be calculated simplifies the implementation of different strategies. The force calculation is the computationally most intensive part of the whole simulation with approximately 95% of the overall cost [12]. Whereas the number of force calculations is dependent on the molecule density and distribution, the distance calculations performed are reduced to less than 2.5% of all interactions for the example of section 2 (cf. fig. 4(a)). The runtime per iteration for $s = 2$ is superior to e.g. $s = 3$ or $s = 4$ here (cf. fig. 4(b)).



(a) Number of calculcations

(b) Runtime per iteration

Figure 4: Calculations for fig. 1

4 Parallelization

The primary target platform is the department Linux Cluster of Workstations “Mozart”, which consists of 64 dual-Xeon nodes with InfiniBand interconnection. In contrast to the Spatial Decomposition method described later, the Atom and Force Decomposition method [13] both do not depend on molecular motion. The core algorithm of the Atom Decomposition (AD), also called Replicated Data, is similar to a shared memory approach. For both, each PE calculates the forces and new positions for one part of the molecules. AD requires each processing element (PE) to store relevant data of all molecules, since it has to be accessible for the force calculation. This means that after each time step a synchronization of the redundant data is necessary, which will inflate the communication effort particularly for a larger number of PEs. The Force Decomposition (FD) method leaves a block of the force matrix and a part of the molecule positions for each PE to calculate. A sophisticated reordering will result in less dependencies between PEs and an improved communication effort compared to the AD approach. The memory requirements are decreased in the same order. However, the number of PEs itself plays a role, e.g. prime numbers will result in force matrix slices for each PE and the FD will degenerate to an AD approach. The Spatial Decomposition (SD) method subdivides the domain and assigns one subdomain to each PE. The subdomains with cuboid shape will be placed in a cartesian topology here. The PE needs access to data of neighbor PEs in the range of r_c. A halo-region will accomodate copies of these molecules, which have to be synchronized. The shape of a subdomain should have minimal surface, since this is directly related to the halo size. But the halo-region is of lower dimension and contains only a few molecules relative to the subdomain. Therefore the communication costs are less than the ones of the AD and FD method. For nucleation processes however, a load balancing technique is favorable, which is not needed for the non-spatial methods. Compared to these also the memory requirements for each PE are lower. The maximal displacement for a molecule per iteration step is limited and the flow induced migrations are of inferior relevance and SD also works well for nanoflow. To make use of Newton's third law, additional communication is needed for all these methods, since the calculated force has to be transported to the associated PE. The SD method implemented uses a full halo (cf. fig. 3(a)) and doesn't make use of Newton's third law within the boundary region. Only the molecule positions of the virtual halo molecules have to be communicated, which is done in three consecutive steps: first x, then y and finally the z direction. The diagonal directions are done implicitly through multiple transportations. Runtime tests on "Mozart" confirm the superiority of the SD method to the AD and FD method (cf. fig. 5) for another example with $1.6 \cdot 10^6$ LJ-molecules, where also a uniform flow was applied. As expected this doesn't show much impact on the runtime for the SD method. It is to mention that the FD method implemented doesn't make use of Newton's third law. The fast InfiniBand connection also gives good results here, whereas AD does not scale up very well. Overall, SD is superior here.

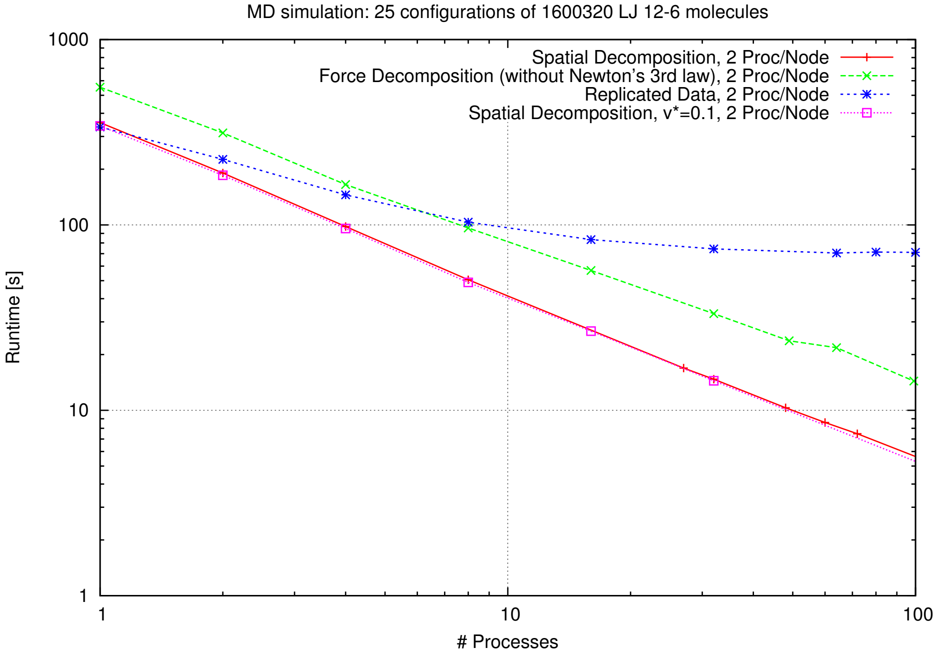


Figure 5: Runtime results for parallel code

5 Summary

The underlying algorithm and data structures of a molecular dynamics based nanofluidic simulation was presented. The underlying Link-Cell data structure is not only indispensable for a fast force calculation with linear run time complexity, but also is useful to obtain an Eulerian view or to couple with lattice based macroscopic continuum simulation techniques in the future. The runtime was not influenced by the flow for the velocities used. The overall performance however might be further improved by overlapping communication-calculation or hybrid methods, introducing shared memory techniques on the nodes, which is one topic of our current MD work.

References

[1] *Allen, M.P., Tildesley, D.J.:* Computer Simulation of Liquids. Oxford University Press, 2003 (reprint).

[2] *Sadus, R.J.:* Molecular Simulation of Fluids - Theory, Algorithms and Object-Orientation. Elsevier, 1999.

[3] *Frenkel, D., Smit, B.:* Understanding Molecular simulation - From Algorithms to Applications. Academic Press, 2002 (2nd ed.).

[4] *Roy, S., Raju, R.:* Modeling gas flow through microchannels and nanopores. J. Appl. Phys. 93 (2003), p. 4870-4879.

[5] *Mi, X.-B., Chwang, A.T.:* Molecular Dynamics Simulations of Nanochannel Flows at Low Reynolds Numbers. Molecules 8 (2003), p. 193-206.

[6] *Mader, D.*: Molekulardynamische Simulation nanoskaliger Strömungsvorgänge. Master thesis, ITT, Universität Stuttgart, 2004.

[7] *Qian, T., Wang, X.-P.:* Driven cavity flow: from molecular dynamics to continuum hydrodynamics. Multiscale Model. Simul. 3 (2005), p. 749-763.

[8] *O'Connell, S.T., Thompson, P.A.:* Molecular dynamics-continuum hybrid computations: A tool for studying complex fluid flows. Phys. Rev. E 52 (1995), R5792.

[9] *Hoover, Wm.G.:* Molecular dynamics. Lecture Notes in Physics 258, Springer, 1986.

[10] *Vrabec, J., Stoll, J., Hasse, H.:* Molecular models of unlike interactions in fluid mixtures. Molec. Sim. 31 (2005), p. 215–221.

[11] *Fincham, D.:* Leapfrog rotational algorithms. Molec. Sim. 8 (1992), p. 165-178.

[12] *Miropolskiy, E.:* Implementation of parallel Algorithms for short-range molecular dynamics simulations. Student research project, IPVS, Universität Stuttgart, 2004.

[13] *Plimpton, S.:* Fast parallel algorithms for short-range molecular dynamics. J. Comp. Phys. 117 (1995), p. 1-19.