Massively parallel molecular-continuum simulations with the macro-micro-coupling tool

Massively Parallel Molecular-Continuum Simulations with the Macro-Micro-Coupling Tool

Philipp Neumann[1] **and Jens Harting**[2]

[1] Department of Informatics, Technische Universität München, Germany
E-mail: neumanph@in.tum.de

[2] Department of Applied Physics, Technische Universiteit Eindhoven, The Netherlands
E-mail: j.harting@tue.nl

Efficient implementations of hybrid molecular-continuum flow solvers are required to allow for fast and massively parallel simulations of large complex systems. Several coupling strategies have been proposed over the last years for 2D/ 3D, time-dependent/ steady-state or compressible/ incompressible scenarios. Despite their different application areas, most of these schemes comprise the same or similar building blocks. Still, to the authors' knowledge, no common implementation of these building blocks is available yet. In this contribution, the Macro-Micro-Coupling tool is presented which is meant to support developers in coupling mesh-based methods with molecular dynamics. It is written in C++ and supports two- and three-dimensional scenarios. Its design is reviewed, and aspects for massively parallel coupled scenarios are addressed. Afterwards, scaling results are presented for a hybrid simulation which couples a molecular dynamics code to the Lattice Boltzmann application of the Peano framework.

1 Introduction

Hybrid molecular-continuum flow simulations allow to bridge the gap between purely molecular fluid descriptions and coarse-grained flow models such as mesoscopic or continuum models. The typical approach in concurrent molecular-continuum simulations is based on the decomposition of the computational domain into a continuum[a] and a molecular dynamics (MD) region. Within the molecular dynamics region, the fluid is resolved on the atomistic level. This yields a physically accurate description on the one hand, but implies high computational costs on the other hand since every molecule's trajectory needs to be computed. In contrast, a computationally fast, but less accurate flow simulation is carried out in the continuum region based on either particle- or mesh-based simulation methods. Examples for the latter comprise (in-) compressible Navier-Stokes or Lattice Boltzmann methods.

Several strategies for various flow problems have been proposed throughout the last years to coupled MD and mesh-based continuum solvers such as strategies for steady-state coupling of incompressible Navier-Stokes[1] or Lattice Boltzmann methods[2] and MD or compressible flux-based coupling schemes[3] for unsteady flow.

Despite their different application areas, most of these schemes comprise the same or similar building blocks. For example, the sampling of average velocities or fluxes is needed in nearly all coupling schemes; the same holds for particle insertion and removal. Depending on the similarity of two coupling schemes, the same algorithms or slightly modified versions or completely different approaches are required for each building block.

[a] In the following, the term "continuum" shall generally denote the coarse-grained flow description.

Besides, in order to handle large-scale problems from nanoscale engineering or biotechnnology, the simulation on massively parallel systems is of essential importance. Parallel solvers for the continuum and the MD region *as well as* a parallel implementation of the coupling mechanisms are hence necessary.

Within this context, we designed the Macro-Micro-Coupling tool[4,5] which is meant to support developers of new hybrid molecular-continuum schemes and allows for massively parallel coupled simulations. We recently described the parallel USHER-based[6] particle insertion implementation of the tool[4] and the software development of the coupling tool[5] in detail. In the following, the parallel performance of the coupling tool in a hybrid molecular dynamics-Lattice Boltzmann simulation is discussed. The software design with emphasis on the parallel extensions of the coupling tool is reviewed in Sec. 2. We report scaling results on different platforms in Sec. 3 and draw a short conclusion in Sec. 4.

2 Software Design

2.1 General Concept and Modularity Aspects

The design of the Macro-Micro-Coupling tool[4,5] is shown in Fig. 1 (a). The modules for momentum and particle insertion can be used, extended or modified by the developer to implement mass and momentum transfer on the MD solver side. In order to use these mechanisms, three interface implementations (`MoleculeWrapper`, `MoleculeIterator`, `MDSolverInterface`) need to be provided by the MD simulation. The `MacroscopicSolverInterface` represents the only required interface on the continuum solver side. All four interface implementations are used by the internal mechanisms of the coupling tool. A direct call to each interface is accomplished via the respective services, cf. the `CouplingMDSolverService` or the `CouplingMacroscopicSolverService`. In order to consistently describe the mapping of flow quantities be-

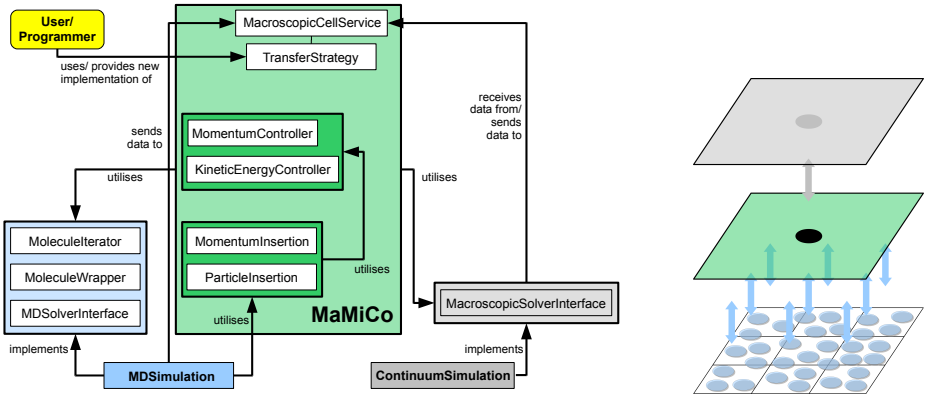

Figure 1. Design and general concept of the Macro-Micro-Coupling tool. (a) Interfaces and module separation. (b) Macroscopic cell-concept: macroscopic cells (green) build a geometrical interface between the mesh-based continuum solver (grey cell) and molecular dynamics (blue-coloured molecules).

tween the continuum and the MD solver, macroscopic cells are introduced, cf. Fig. 1 (b). They represent the discrete control volumes for sampling and exchange of mass and momentum; both two- and three-dimensional scenarios are supported.

2.2 Parallel Extensions

In distributed parallel simulations, the macroscopic cells are always stored on the same process as the respective volume in the MD simulation. Since the cells are strictly tied to the MD simulation, it is only the continuum solver which is left to be linked to the topology of the coupling tool (or the MD solver, respectively). For this purpose, the interface implementation of the `MacroscopicSolverInterface` needs to be provided. The method `accumulateSendReceiveInformation()` is called during the initialisation phase of the coupling. It loops over all continuum cells and calls `addSendReceiveInformation(cellPosition)` of the `CouplingMacroscopicSolverService` on each cell. The latter method uses the two methods `receiveMacroscopicQuantityFromMDSolver(...)` and `sendMacroscopicQuantityToMDSolver(...)` of the `MacroscopicSolverInterface` to determine if the flow data of a particular grid cell are received/ sent from/ to the MD solver. Depending on the coupling strategy and the respective implementation of the `MacroscopicSolverInterface`, an arbitrary subset of the macroscopic cells can thus be chosen in the initialisation phase for the quantity transfer mechanisms. As a consequence, the coupling tool has full knowledge of all required macroscopic cell-based communications after this phase.

In order to exchange quantities between the continuum and the MD solver during the coupled simulation, local macroscopic cell buffers are filled with respective mass and momentum contributions. A call to `receiveMacroscopicQuantitiesFromMacroscopicSolver()` or `sendMacroscopicQuantitiesToMacroscopicSolver()` of the `MacroscopicCellService` triggers the MPI-based communication between the processes.

3 Results

We recently investigated the sequential performance of the coupling tool as well as its parallel performance with respect to the parallel USHER-based particle insertion scheme[4]. In the following, the parallel performance of the tool in molecular dynamics-Lattice Boltzmann simulations of plane channel flow is measured. For this purpose, a single-centred Lennard-Jones MD simulation is coupled to the Lattice Boltzmann solver of the Peano framework[7].

The coupling is established following the principles of the steady-state based coupling approach by Dupuis *et al.*[2]. In our scenario, a fully three-dimensional domain is considered which consists of $54 \times 54 \times 54$ Lattice Boltzmann cells; each Lattice Boltzmann cell corresponds to one macroscopic cell of the coupling tool. In the middle, the molecular dynamics domain is embedded, cf. Fig. 2. The number density in the MD simulation is chosen as $n = 0.6$, and the Lennard-Jones parameters are scaled to unity. One coupling cycle consists of two Lattice Boltzmann time steps and 100 concurrent molecular dynamics time steps; though significantly more time steps are required to reach steady-state in each

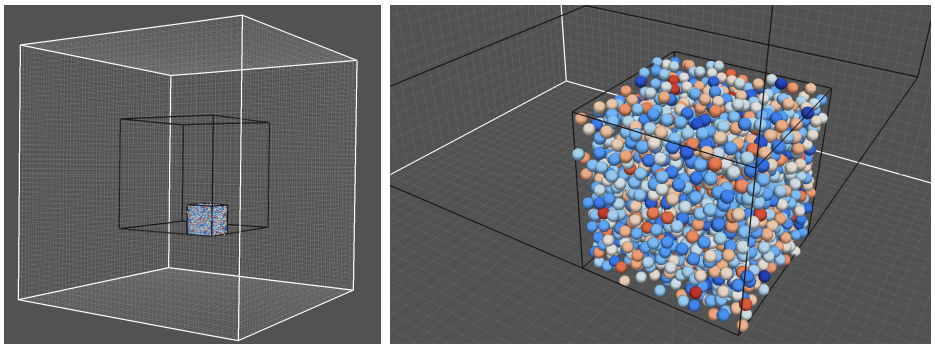

Figure 2. Parallel molecular dynamics-Lattice Boltzmann simulation executed on 64 cores. (a) Complete simulation domain consisting of $54 \times 54 \times 54$ Lattice Boltzmann cells. The molecules that are handled by the process on rank 0 are shown as coloured spheres. (b) Zoom into the molecular sub-domain on rank 0.

cycle, this choice is found to be sufficient for the scaling experiments. It further represents a suitable measure in case of unsteady flow simulations. Within a boundary strip of two Lattice Boltzmann cells, the flow velocity of the Lattice Boltzmann simulation is sent to the MD simulation. The molecules are relaxed towards this target average velocity in each macroscopic cell. In the outermost cell strip, the mass of the molecular system is relaxed towards the reference mass. For this purpose, the average mass is measured over one coupling cycle, and the mass difference between this average and the reference mass $m^{ref} = n \cdot dx^3$ is imposed over the next coupling cycle where dx denotes the cell size of one macroscopic, i.e. Lattice Boltzmann, cell. The removal of molecules is accomplished using a random removal technique whereas the particle insertion is based on the USHER scheme[4]. In the macroscopic cells which are located in the inner region of the molecular dynamics domain, the average velocity is sampled and sent to the Lattice Boltzmann solver.

Two scenarios are evaluated: in scenario A, the cell size is chosen as $dx = 2.5$ using $1.3 \cdot 10^5$ molecules. The scenario B applies cells of size $dx = 5.0$ and holds $1.0 \cdot 10^6$ molecules. This corresponds to a MD simulation which is eight times bigger than in scenario A and yields the same macroscopic cell topology in both scenarios. The strong scaling of a single coupling cycle has been measured on two IBM systems – Shaheen[b](IBM BlueGene/P) and Huygens[c] (IBM pSeries 575). The computationally intensive MD simulation is executed in parallel mode using a standard domain decomposition to distribute the computational load among the processes. The Lattice Boltzmann simulation is executed in sequential mode on rank 0. The initialisation phase including the setup phase for the parallel topology between the solvers is negligible for both scenarios A and B; its contribution to the overall runtime has been found to be of the order of seconds.

The speed-up factors for one coupling cycle are shown in Tab. 1 and 2 for scenarios A and B. Besides the speed-ups for the hybrid molecular dynamics-Lattice Boltzmann simulations, the speed-ups for a pure MD simulation of the same MD setting are depicted

[b]See `http://www.hpc.kaust.edu.sa/documentation/user_guide/resources/shaheen/` for details.
[c]See `http://sara.nl/systems/huygens/description` for details.

Proc.	Shaheen		Huygens	
	MD-LB	MD	MD-LB	MD
1	1.0	1.0	1.0	1.0
8	6.6	6.8	6.5	6.4
64	36.0	44.0	34.6	37.6
512	105.4	206.6	98.0	122.5

Table 1. Strong scaling for scenario A. The first column shows the number of processor cores. The speed-ups obtained on Shaheen and Huygens are listed in the second and third major column. For both machines, the speed-up of the hybrid molecular dynamics-Lattice Boltzmann (MD-LB) simulation is compared to a pure MD simulation.

Proc.	Shaheen		Huygens	
	MD-LB	MD	MD-LB	MD
1	1.0	1.0	1.0	1.0
8	7.2	7.3	7.3	6.9
64	46.0	49.6	45.7	45.3
512	244.0	321.0	235.5	249.7
1728	484.4	814.5	456.7	494.7

Table 2. Strong scaling for scenario B.

for each scenario and platform. Especially for the lower core counts, the sequential Lattice Boltzmann simulation plays a negligible role, and similar speed-ups as in the pure MD simulations can be reached.

4 Conclusion

We presented the parallel extension of our Macro-Micro-Coupling tool which is meant to support developers of massively parallel molecular-continuum simulations. The strong scaling measurements indicate good scaling behaviour on moderate core counts. In these scenarios, a parallelisation of the computationally intensive MD simulation was found to be sufficient whereas the Lattice Boltzmann simulation was executed sequentially. In order to obtain speed-ups on bigger core counts, a parallel continuum solver is required as well. First steps towards a spatially adaptive parallel Lattice Boltzmann solver within the Peano framework are already taken. The realisation of a fully parallel molecular dynamics-Lattice Boltzmann simulation is therefore expected in near future.

Acknowledgements

This publication is based on work supported by the HPC-EUROPA2 project (project number: 228398) with the support of the European Commission Capacities Area - Research Infrastructures Initiative and by Award No. UK-C0020, made by King Abdullah University of Science and Technology (KAUST). We further gratefully acknowledge the support of the Faculty Graduate Center CeDoSIA at the Technische Universität München.

References

1. T. Werder, J. H. Walther, and P. Koumoutsakos, Hybrid atomistic-continuum method for the simulation of dense fluid flows, J. Comput. Phys. 205, 373–390, 2005.
2. A. Dupuis, E. M. Kotsalis, and P. Koumoutsakos, Coupling lattice Boltzmann and molecular dynamics models for dense fluids, Phys. Rev. E 75(046704), 2007.
3. R. Delgado-Buscalioni and P. V. Coveney, Continuum-particle hybrid coupling for mass,momentum and energy transfers in unsteady flows, Phys. Rev. E 67(046704), 2003.
4. P. Neumann and N. Tchipev, A Coupling Tool for Parallel Molecular Dynamics–Continuum Simulations, Proceedings of the International Symposium on Distributed an Parallel Computing 2012, 2012.
5. P. Neumann, W. Eckhardt. and H.-J. Bungartz, Hybrid Molecular-Continuum Methods: From Prototypes to Coupling Software, submitted.
6. R. Delgado-Buscalioni and P. V. Coveney, USHER: An algorithm for particle insertion in dense fluids, J. Chem. Phys. 119(2), 2003.
7. P. Neumann and T. Neckel, A Dynamic Mesh Refinement Technique for Lattice Boltzmann Simulations on Octree-Like Grids, Computational Mechanics, DOI 10.1007/s00466-012-0721-y, 2012. published online.