

Lattice Boltzmann Simulation and Visualisation of Adsorption Processes in Complex Geometries

J.Bernsdorf¹, U.Jaekel¹, T.Zeiser², T.Takei³, H.Matsumoto⁴, and K.Nishizawa⁴

¹ C&C Research Laboratories, NEC Europe Ltd., Rathausallee 10,
D – 53757 Sankt Augustin, Germany

² Institute of Fluid Mechanics (LSTM), University of Erlangen - Nuremberg,
Cauerstr. 4, D – 91058 Erlangen, Germany

³ Internet Systems Research Laboratories, NEC Corporation, 4-1-1 Miyazaki,
Miyamae-ku, Kawasaki Kanagawa, Japan

⁴ NEC Informatec Systems, Ltd., Kanagawa Science Park, Takatsu-ku,
Kawasaki 213, Japan

Abstract. In this paper, we describe briefly the lattice Boltzmann method, which simulates the flow on a mesoscopic level using a time-, space- and velocity-discrete version of the Boltzmann equation. Further we describe its coupling to the real-time visual simulation library RVSLIB, which allows an efficient client/server based on-line visualisation of the simulation results and/or the image creation on the server.

An extension of the standard lattice Boltzmann model for adsorption processes is given together with simulation results and visualisation examples for adsorption in porous media.

Keywords: Lattice Boltzmann, adsorption, real time visualisation.

1 Outline

The simulation and visualisation of time dependent multi-species flow phenomena in complex geometries is often crucial to understand processes which play an important role in environmental sciences and chemical engineering.

Using the lattice Boltzmann method, a detailed simulation of the time- and space dependent diffusion and adsorption processes can be carried out on a computational domain extending over several million grid points.

An efficient approach for visualising the enormous amount of data being produced at each time step can be achieved with a server sided data reduction carried out in batch mode or interactively with a steering client. Instead of storing the whole dataset, only the relevant information is extracted during the ongoing simulation, thus bypassing the major bottleneck of bandwidth to the harddisk.

2 Lattice Boltzmann Method

The lattice Boltzmann method is based on the numerical simulation of a time-, space- and velocity-discrete Boltzmann-type equation. The propagation and interaction of the particles of an 'artificial computer fluid' is calculated in terms of

the time evolution of a density distribution function, representing an ensemble average of the particle distribution. The flow velocity and the fluid density are derived from the moments of that (time- and space-discrete) density distribution function, while the pressure is linked to the density by the (model specific) speed of sound. It can be shown theoretically [1] and by a detailed quantitative evaluation of simulation results (see, e.g., [2]), that these flow quantities fulfill the time dependent incompressible Navier Stokes equations under certain conditions.

Extensions for locally inhomogeneous diffusion- and reaction processes have been successfully applied to simulate processes in technical chemical engineering devices [3, 4].

A special feature of the lattice Boltzmann method is the efficient and cheap way of handling equidistant Cartesian lattices. In combination with a highly optimised implementation, tens of millions of grid points can be handled on large PCs or workstations. This allows one to use the ‘marker and cell’ approach for representing the geometry with sufficient accuracy by marking single lattice nodes as being occupied or free.

While more advanced models have been recently developed and successfully tested, which get rid of the memory consuming full matrices [5], for several types of geometries the simple approach used here is still competitive with respect to CPU time and memory consumption. Due to the local (cellular automata like) structure of that simple-type lattice Boltzmann algorithm, the vectorisation and parallelisation is easily possible with a very good performance on modern high performance computers [6].

With respect to porous media, highly complex geometries are available as three-dimensional ‘real objects’ which can be digitised using 3D computer tomography (3D CT) [7, 8]. The 3D CT-data can be converted to the marker and cell data format. Numerical simulations using such an explicit representation of the geometry, as presented in this paper, provide important information, which cannot be achieved when using the well known homogenization approaches. A detailed simulation in those geometries is, due to the problems concerning grid generation and efficiency, often not feasible within the framework of Navier Stokes methods but can be done with lattice Boltzmann [9].

3 Real-Time Visualisation

The real-time visual simulation library (RVSLIB) [10] is a system for visualising computational results concurrently with the on-going simulation for a broad range of scientific computing applications such as computational fluid dynamics and structural analysis. The basic concept is a real-time data reduction to the required image (according to pre-defined scenarios or interactively) instead of writing the whole dataset to the harddisk and carrying out an a-posteriori visualisation.

RVSLIB is a server-client type system consisting of RVSLIB/Server and RVSLIB/Client. These two components can be either on the same machine or on different machines.

The RVSLIB/Server supports a set of FORTRAN subroutines to be incorporated into user applications. It enables the user to display an image created by the RVSLIB/Server and to change simulation parameters through a GUI (Graphical User Interface). Alternatively, the generation of movies in batch mode is possible by using a script steering the camera position etc.

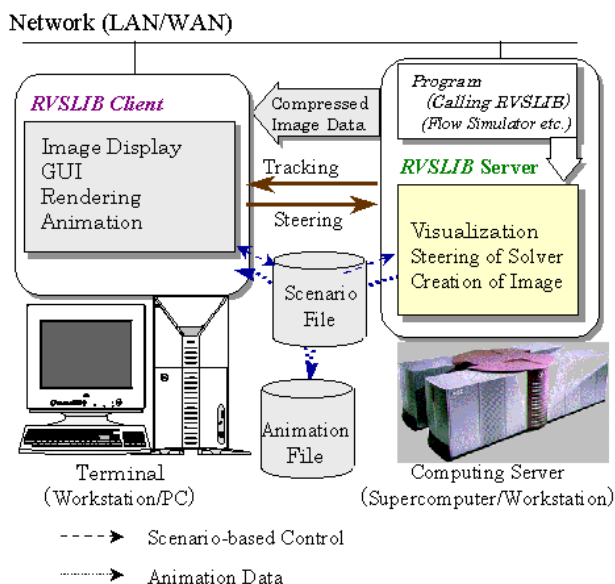


Fig. 1. The RVSLIB client/server concept.

3.1 Programming Interface and Coupling

RVSLIB visualisation routines can be integrated into existing simulation software by adding a few subroutine calls. The additional CPU-time needed for the image generation and data compression during the simulation is, depending on what has to be visualised, usually only a few percent of the total CPU-time.

Within the lattice Boltzmann algorithm, all local flow quantities (velocity, pressure, species concentration) are calculated from the density distributions during the simulation procedure (once per iteration). For a visualisation, additional arrays have to be defined to store and later provide these quantities for the RVSLIB routine calls at the end of the main loop. So, the additional overhead is restricted to the relatively small memory size necessary to store these arrays and the CPU-time due to the image rendering.

A snapshot from a movie visualising a time dependent catalytic reaction-diffusion process in a porous media produced with LBA/RVSLIB is shown in figure 3.

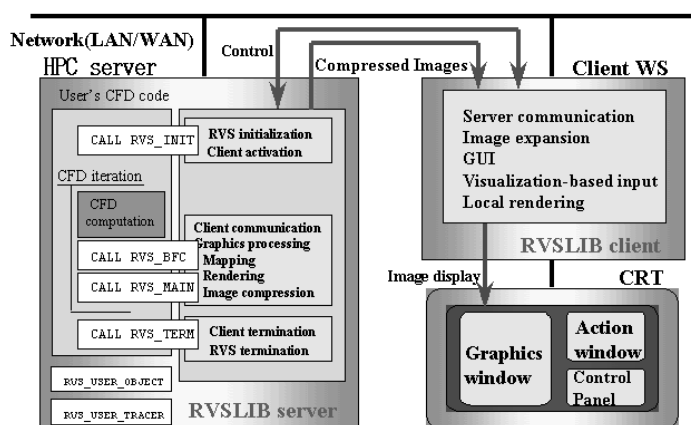


Fig. 2. Implementation scheme for RVSLIB.

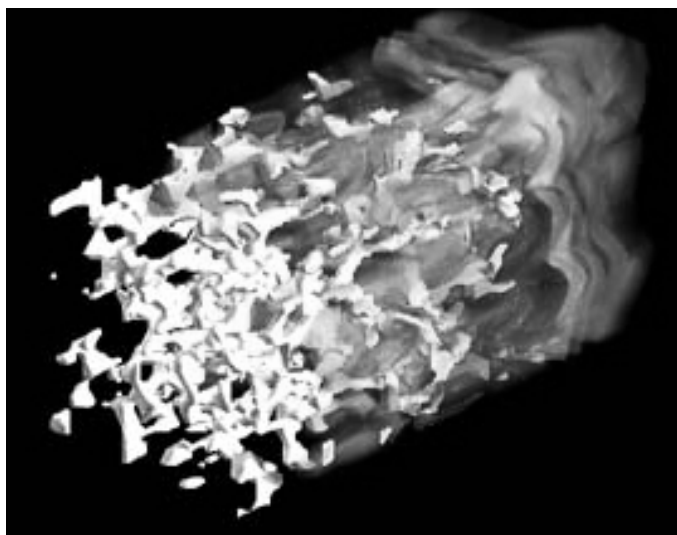


Fig. 3. Snapshot from a time dependent visualisation of the chemical reaction $A + B \rightarrow C$ (flow from the left to the right). Displayed is the geometry and several isosurfaces of the product concentration $[C]$ in grey, simulated with LBA/RVSLIB.

4 Nonlinear Adsorption/Desorption

Adsorption/desorption can be introduced to the lattice Boltzmann algorithm by applying a local rule describing the temporal change of the adsorption rate $\partial s/\partial t$ on the solvent concentration c and the adsorbed mass s (i.e. the immobile mass deposited per unit volume of the porous media matrix) [11]:

$$\frac{\partial s}{\partial t} = r(k_p c^p - s^q) \quad (1)$$

with parameters $k_p > 0$ and exponents p, q fulfilling $p/q \leq 1$. In equilibrium, i.e. for $\partial s/\partial t = 0$, this model reduces to a Freundlich isotherm

$$s = k c^n \quad (2)$$

with $k = k_p^{1/q}$ and $n = p/q$.

Such a relationship between s and c frequently describes the adsorption of substances like pesticides, polycyclic aromatic hydrocarbons and heavy metals in soil and aquifer sediments (e.g. [12], [13], [14]) over several orders of magnitude in the concentrations.

4.1 LB Implementation

Using a diffusion scheme for the solvent (e.g. based on the algorithm described by Flekkøy [15]), adsorption is simulated within our LB code by introducing arrays for the adsorbed species concentration on the surface of the solid phase.

Each time step, the concentrations of the adsorbed species and the solvent are locally updated by applying equation (1).

4.2 Simulations

For (effectively) one-dimensional media, [16] and [17] showed by asymptotic analysis that in the quasi-equilibrium case, after pulse-type injection the concentrations at a fixed position (breakthrough curves) should approach power laws in time:

$$\begin{aligned} c &\propto t^{-\alpha} \\ s &\propto t^{-\beta} \end{aligned} \quad (3)$$

where the exponents $\alpha = 1/(1 - n)$ and $\beta = n/(1 - n)$ are determined by the Freundlich exponent n alone. This is in contrast to the exponential decay of breakthrough curves observed for chemically inert substances without adsorption.

For the rate equation (1), the concentrations cannot always approach quasi-equilibrium. However, asymptotic analysis predicts that the quasi-equilibrium asymptotics holds for the case $q = 1$ [11].

For our simulations we exploited equation (1) with the factors $r = 0.1$ and $k_p = 0.001$. We carried out simulations for the exponents $n = p = 0.5$ and $n = p = 0.8$ for a porous media generated from a 3D-CT scan of a SiC-matrix (see figure 3).

After establishing a steady laminar flow, for a short time a small amount of the solvent species was introduced near the inlet and further on transported downstream through the porous media by advection/diffusion.

After the peak of the concentration passed the porous media an order of 10^6 more iterations where necessary to achieve the asymptotic behaviour, which took several CPU-hours on a NEC single vector-CPU SX 6i.

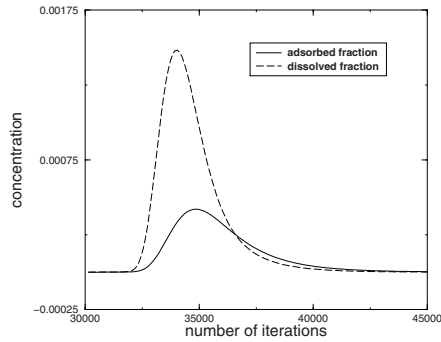


Fig. 4. Linear plot of the time evolution (first 15000 iterations) of the concentration for the adsorbed and dissolved fraction for an exponent $n = p = 0.8$.

Figure 4 shows the first 15000 iterations of the time evolution of the solvent concentration and the adsorbed mass at a surface point inside the porous media in a linear plot for the exponent $n = p = 0.8$. It can be seen that the maximum of the solvent concentration is reached earlier and decays faster compared to that of the adsorbed mass.

Figures 5 show the long term behavior for the two exponents $n = p = 0.5$ (left) and $n = p = 0.8$ (right), respectively. In both cases, one observes approach towards the power laws predicted in equation (3). More detailed simulations with more iterations will be necessary to investigate the actual long term behaviour and show whether a proper power law behaviour was already achieved.

Keeping this in mind, with a regression in the linear part over the last 200000 iterations the slopes were determined to be $t = 1.03$ (adsorbed species) and $t = 2.05$ (solvent species) for the exponent $n = p = 0.5$ as well as $t = 4.36$ (adsorbed species) and $t = 5.45$ (solvent species) for the exponent $n = p = 0.8$. This preliminary result is in good agreement with the predicted $t = 1.0$ (adsorbed) and $t = 2.0$ (solvent) for the exponent $n = p = 0.5$ and in the correct order for

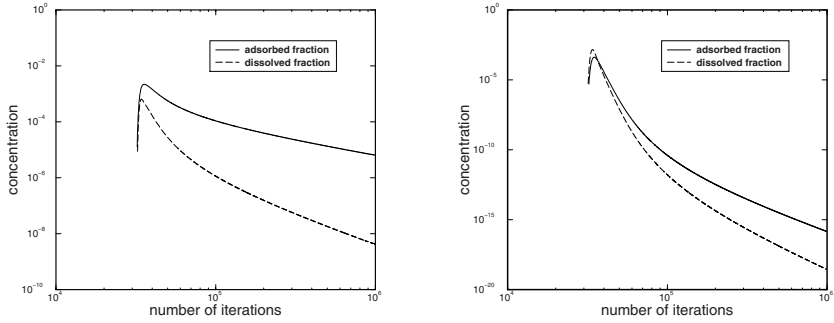


Fig. 5. Logarithmic plot of the time evolution of the concentration for the adsorbed and dissolved fraction for the exponents $n=p=0.5$ (left) and $n=p=0.8$ (right).

the exponent $n=p=0.8$, where the predicted slopes are $t=4.0$ and $t=5.0$ respectively. Due to the slower decay resulting from the exponent $n=p=0.8$ one should expect the result to approach the theoretical value even better with more than the actual 10^6 iterations.

4.3 Visualisation

Using the previously described coupling of our LB code with the RVSLIB routines, we were able to produce movies giving a detailed insight in the time dependent concentrations of the adsorbed and solvent species.

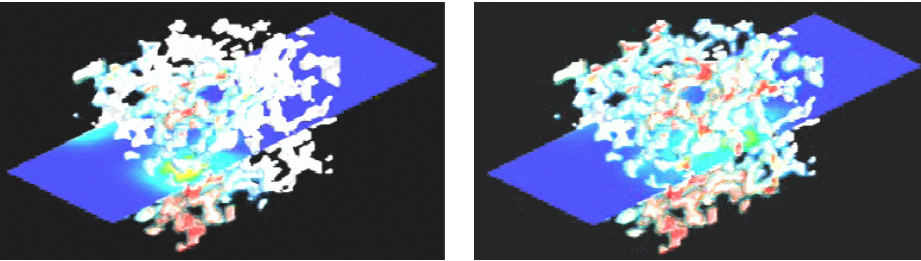


Fig. 6. Screen-snapshot from the visualisation (flow from the left to the right). Displayed is the concentration of the solvent (x-y plane) and the adsorbed species (on the surface of the porous media) at two different time steps. One can observe the peak concentration of the solvent go downstream through the porous media, followed by the peak concentration of the adsorbed species.

Figure (6) shows two screen-snapshots taken from the mpeg-movie at two different time-steps. The concentration of the solvent is displayed on a cut in

the x-y plane, while the concentration of the adsorbed species colours the surface of the porous media.

The simulation was carried out on a single CPU of a NEC SX 6i as a server and a NEC PowerMate PC as a client, both connected by a LAN.

5 Conclusion

The simulation of complex flows with concurrent visualisation using NEC's real-time visual simulation library was shown to be a helpful tool in computational fluid dynamics. We further demonstrated how the lattice Boltzmann approach can be extended for the simulation of adsorption. For a non-linear adsorption/desorption model the preliminary simulation results were found to indicate good accordance with the theoretical predicted power law model.

References

1. Frisch U., d'Humières D., Hasslacher B., Lallemand P., Poke Y. and Rivet J.-P., *Complex Systems* **1**: 649-707 (1986).
2. Bernsdorf J., Zeiser Th., Brenner G. and Durst F., *Int. J. Mod. Phys. C*, **9**(8): 1129-1141 (1998).
3. Chen S., Dawson P., Doolen G.D., Janecky D.R. and Lawiczak A., *Comp. & Chem. Eng.* **19**(6-7): 617-646 (1995).
4. Zeiser T., Lammers P., Klemm E., Li Y.W., Bernsdorf J. and Brenner G., *Chem. Eng. Sci.*, **56**(4): 1697-1704 (2001).
5. Schulz M., Krafczyk M., Tölke J. and Rank E., *Lecture Notes in Computational Science and Engineering* **21**:115-122 (2002).
6. Zeiser T., Brenner G., Lammers P., Bernsdorf J., in: C.B. Jenssen et al: *Proceedings of the Parallel CFD 2000 Conference*, May 22-25, Trondheim, Norway), Elsevier: 407-414 (2001).
7. Rothman D.H., *Geophysics* **53**: 509-518 (1988).
8. Bernsdorf J., Günnewig O., Hamm W. and Munker, O., *GIT Labor-Fachzeitschrift* **4/99**: 387-390 (1999).
9. Bernsdorf J., Schäfer M. and Durst F., *Int. J. Numer. Meth. Fluids* **29**: 251-264 (1999).
10. Takei T., Matsumoto H., Muramatsu K. and Doi S., *Parallel vector performance of concurrent visualization system RVSLIB on SX-4*, *Proc. 3rd Pacific Symposium on Flow Visualization and Image Processing* (2001).
11. Jaekel U. and Vereecken H., *Phys.Rev. E*, **65**: 041402(1-9) (2002).
12. Carmo A. M., Hundal L. S. and Thompson M. L., *Environ. Sci. Technol.*, **34**: 5184-5188 (2000).
13. Xing B., Pignatello J. and Gigliotti B., *Environ. Sci. Technol.*, **30**: 2432-2440 (1996).
14. Vereecken H., Jaekel U., Esser O. and Nitzsche O., *Contaminant Hydrology*, **39**: 7-34 (1999).
15. Flekkøy E.G., *Phys. Rev. E* **47**(6): 4247-4257 (1993).
16. Grundy R., van Duijn C. and Dawson C., *Q. J. Mech. Appl. Math.*, **47**: 69 (1994).
17. Jaekel U., Georgescu A. and Vereecken H., *Water Resour. Res.*, **36**: 3093-3098 (1996).