Table 1. Comparison of effectiveness of the load balance algorithms. WB – without balancing, CAVL – centralised algorithm with virtual layers, DA – diffusive algorithm

N of PE	2 PE	3PE	4PE	5PE	6PE	7PE	8PE	20PE
WB	121768	107878	68801	63481	65332	47470	46716	24687
CAVL	122294	81900	61902	49453	41223	35826	30744	13227
DA	122048	83191	62606	50083	41465	34451	31508	13353

7. Conclusion

The described algorithms of the PIC method realization provide high performance of the assembled program execution, its high flexibility in reconstruction of the code and dynamic tunability to available resources of a multicomputer.

High performance of the program execution provides the modelling of big size problems such as the study of a cloud plasma explosion in magnetised background, modelling of interaction of a laser impulse with plasma. For the first problem with a non-uniform magnetic field, the 60x60x100 grid and 14 million particles were used for modelling. Process of modelling made more than 2000 time steps.

Flexibility of program construction with the assembly technology provides assembling a certain program for solution of a certain problem, but not a general program. This also improves the performance of an assembled program.

We apply the assembly technology to realisation of different numerical methods and hope to create the general tool to support realisation of mathematical approximating models.

References

1. V.A.Valkovskii, V.E.Malyshkin.Synthesis of Parallel Programs and Systems on the Basis of Computational Models.// Nauka, Novosibirsk, 1988. (In Russian)
2. V.Malyshkin. Functionality in ASSY System and Language of Functional Programming.// Proc. "The First Aizu International Symposium on Parallel Algorithms/Architecture Synthesis", 1995, Aizu-Wakamatsu, Fukushima, Japan. IEEE Comp. Soc. Press, Los Alamitos, California. P. 92-97.
3. M.Kraeva, V.Malyshkin. Implementation of PIC Method on MIMD Multicomputers with Assembly Technology.// Proc. of HPCN Europe 1997 (High Performance Computing and Networking) Int. Conference, LNCS, Vol.1255, Springer Verlag, 1997. pp. 541-549.
4. Hockney R, Eastwood J. Computer Simulation Using Particles.// McGraw-Hill Inc. 1981.
5. Berezin Yu.A., Vshivkov V.A. The method of particles in rarefied plasma dynamic.// Novosibirsk: Nauka, 1980 (in Russian)
6. M.Kraeva, V.Malyshkin. Dynamic load balancing algorithms for implementation of PIC method on MIMD multicomputers.// Programmirovanie, N 1, 1999 (accepted for publication, in Russian)
7. A.Corradi, L.Leonardi, F.Zambonelli Performance Comparison of Load Balancing Policies based on a Diffusion Scheme// Proc. of the Euro-Par'97 LNCS Vol. 1300, p.882-886.

(64) / (65)

Computational Aspects of Multi-Species Lattice-Gas Automata*

D. Dubbeldam, A.G. Hoekstra and P.M.A. Sloot

University of Amsterdam,
Faculty for Mathematics, Computer Science, Physics, and Astronomy,
Kruislaan 403, 1098 SJ Amsterdam,
The Netherlands,
Tel +31 20 525 7563, Fax +31 20 525 7490,
http://www.wins.uva.nl/research/pscs/,
Email:{dubbelda, alfons, sloot}@wins.uva.nl

Abstract. We present computational aspects of a parallel implementation of a multi-species thermal lattice gas. This model, which can be used to simulate reaction-diffusion phenomena in a mixture of different fluids, is analyzed for a fluid system at global equilibrium. Large system sizes combined with long-time simulation makes parallelization a necessity. We show that the model can be easily parallelized, and possesses good scalability. Profiling information shows the random number generator has become a bottleneck. The model can be statistically analyzed by calculating the dynamic structure factor $S(k,\omega)$. As an illustration, we measure $S(k,\omega)$ for a one-component system, and extract the values of transport coefficients from the spectra. Finally, $S(k,\omega)$ is shown for a two-component thermal model, where the central peak is more complicated, due to the coupled entropy-concentration fluctuations.

1 Introduction

Lattice-gas automata (LGA) are a relative novel method to simulate the hydrodynamics of incompressible fluids [1]. The flow is modeled by particles which reside on nodes of a regular lattice. The extremely simplified dynamics consists of a streaming step where all particles move to a neighboring lattice site in the direction of its velocity, followed by a collision step, where different particles arriving at the same node interact and possibly change their velocity according to collision rules. The main features of the model are exact conservation laws, unconditional stability, a large number of degrees of freedom, intrinsic spontaneous fluctuations, low memory consumption, and the inherent spatial locality of the update rules, making it ideal for parallel processing.

Different LGA models exist, both in two and three dimensions, where the models differ in the number of used velocities and the exact definition of the collision rules (see [2–4]). The basic LGA model proposed by Frisch, Hasslacher,

* presenting author: D. Dubbeldam

and Pomeau (the FHP I model) is a two dimensional model, based on a triangular grid, where up to six particles (hexagonal symmetry) may reside at any of the sites [5]. The model was extended with rest-particles (FHP II model) and the maximization of the number of collision rules (FHP III model), resulting in a higher maximum Reynolds number. The model of Grosfills, Boon, and Lallemand (GBL-model) introduced non-trivial energy conservation to the LGA's, allowing the simulation of temperature, temperature gradients, and heat conduction [6, 7].

The models can be extended with multiple species, where each species is tagged with a different color, but differs in no other way [1]. In this paper we focus on such an extended LGA model, which can be used to simulate reaction-diffusion phenomena in a mixture of different fluids. The only four properties that are conserved in this model are mass, momentum, energy, and color, in case there is only diffusion and no reaction.

The Boolean microscopic nature, combined with stochastic micro-dynamics, results in intrinsic spontaneous fluctuations in LGA [6–8]. Such fluctuations can be described by the dynamic structure factor $S(k,\omega)$, the space and time Fourier transform of the density autocorrelation function [9]. The fluctuations extend over a broad range of wavenumbers $\mathbf{k}$ and frequencies ω. $S(k,\omega)$ of real fluids can be measured by light-scattering experiments, where the measured quantity is the power spectrum of density fluctuations. In the hydrodynamic limit, one observes two Doppler-shifted Brillouin peaks and a central Rayleigh peak. Such spectra are also observed in GBL [6, 7]. A measured spectrum from a GBL simulation contains transport coefficient information. The Brillouin peaks correspond to the sound modes, the Rayleigh peak corresponds to energy density fluctuations and diffusion. We are interested in fluctuations in two-species GBL models, and therefore have extended our multiple-species GBL with routines to measure $S(k,\omega)$. The main goal in the present work is to discuss the computational aspects of this model.

2 The GBL model

The two dimensional GBL lattice gas model is based on a triangular lattice with hexagonal symmetry for isotropy reasons. The particles have unitary mass with no spatial extension. The model evolves according to the following dynamical rule:

$$n_i(\mathbf{x}+\mathbf{c_i}, t+1) = n_i(\mathbf{x}, t) + \Delta(\mathbf{n}(\mathbf{x}, t)) \qquad i = 1, \ldots, 19 \qquad (1)$$

where the n_i are Boolean variables representing the presence/absence of a particle at site x at time t, and $\Delta(\mathbf{n}(\mathbf{x},t))$ is the collision operator, and $\mathbf{c_i}$ is the velocity vector. The exclusion principle (no more than one particle is allowed at a given time in a certain channel) garantues the convenient specification of a state of a node as a 19-bits integer.

The introduction of temperature requires a multiple-speed model (ideally a velocity distribution). The GBL-model has four different speeds, $0, 1, \sqrt{3}$, and 2, which corresponds to a kinetic energy of $\frac{1}{2}mv^2 = 0, \frac{1}{2}, \frac{3}{2}$, and 2, respectively.

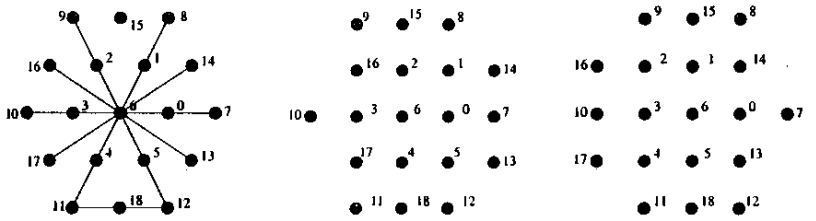
The model has one rest particle, 6 velocities of speed 1, 6 velocities of speed $\sqrt{3}$, and 6 velocities of speed 2. The mapping of these velocities to bits are as shown in figure 1(a). As an example, in the streaming step bit 7 will be propagated two lattice nodes to the right.

The collisions redistribute mass, momentum, and energy among the 19 channels of each node at every time step such that mass, momentum, and energy are conserved. The collision outcome is chosen randomly among all states belonging to the same class as the input state (including the input state itself). This procedure is equivalent to the *random algorithm* [10]

Our implementation of the GBL model is such that it contains all other known models based on a triangular grid, FHP I (bit 0, ... ,5), FHP II/III (bit 0, ... ,6) etc. Hence, to simulate another lattice gas model we stream only a subset of the 19 bits, and we use a different collision lookup-table. To simulate a square grid, the streaming has to be slightly adjusted.

3 Implementation Aspects

The triangular grid To denote each node in a set of integer coordinates (x, y) we multiply the coordinates by the factor $(2, \frac{2}{\sqrt{3}})$. Thus, each node of a triangular lattice can be uniquely mapped to a node of a square lattice (figure 1). Note that when working with rectangular space, the lengths along the vertical axes should be multiplied by $\frac{\sqrt{3}}{2}$. To avoid an awkward diamond shaped grid, the streaming step is different for even and odd parity of the lattice (see figure 1(b) and 1(c)). The conversion of a rectangular shaped triangular lattice now remains rectangular.



(a) channel labeling (b) even line (c) odd line

Fig. 1. In (a) the channel labeling per lattice node is shown. The conversion from the triangular lattice to the square lattice is based on a the parity of the lattice. On even lines we propagate using (b), for odd lines we propagate using (c).

Parallelization Parallelization of grid based algorithms like the lattice-gas automata is done by means of a data decomposition strategy, where the computational domain is decomposed into sub-domains. Each processor performs computations on a certain sub-domain and exchanges information with other nodes in order to resolve dependencies. For that purpose we use a ghost-boundary of two lattice nodes (the GBL model has a highest velocity of 2, i.e. each lattice site streams its particles of velocity 2 to its next-nearest neighbors). The Fast Fourier Transform (FFT) used to measure the $S(k,\omega)$ is a parallel version of FFTW [11] for in-place, multi-dimensional transforms on machines with MPI [12]. It has the requirement that the decomposition is a slice-decomposition (equal sub-volumes in one dimension). In this paper we restrict ourselfs to the dynamical structure factor for a fluid system at global equilibrium. We have periodic - or no-slip boundary conditions and no obstacles in the fluid. Hence, we have near-optimal load-balancing. If the lattice grid is not rectangular and/or contains obstacles in the fluid the decomposition can be done with the ORB-method [13, 14]. In this case, the parallel FFT routine needs to be adjusted.

Initialization The GBL model can be shown to be free of known spurious invariants, which have plagued other LGA's. These models must be correctly initialized to avoid such invariants. The best known spurious invariant is the conservation of total transverse momentum on even and odd lines every two time steps. This spurious invariant can be eliminated by choosing initial conditions such that the total transverse momentum is zero [8]. We accomplish this by initializing the density only pairs-wise with opposite velocities.

Data structures The GBL model uses 19 velocities and can be conveniently represented as a 19-bit integer. The complete lattice is allocated as a single array of a structure containing $n+1$ longs (a long is 4 bytes), where n is the number of species (colors). One state denotes the presence/absence of the particle, and the n longs denotes the presence/absence of the particular colors. This generic approach allows a variable number of species (decided at compile-time) at the cost of using more memory. An index to a lattice node can be found in the array by the relation:

$$\text{lattice}(x,y) = x + y \times \text{ lattice width} \tag{2}$$

i.e. the local data is expected to be stored in row-major order (C order).

The streaming-step The streaming step is accomplished by using two different lattice grids, named current and new. The new lattice is used to calculate the new state at the next time-step from the values of the current lattice. After updating the lattice, we swap the pointers to the lattices and the new lattice becomes the current lattice. If memory size is a problem it would be possible to do the streaming step in-place, at the cost of accessing each lattice point several times instead of just once. Since our models are all two-dimensional and using domain decomposition, memory should not be a real problem here.

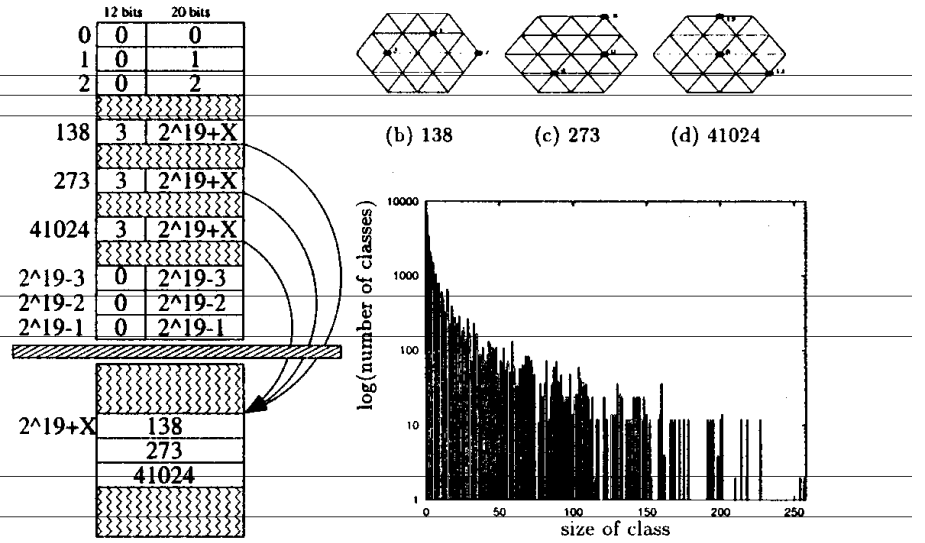


Fig. 2. Collision table. The first 2^{19} indices are divided into 12 and 20 bits. The left 12 bits denotes the number of collision outcomes, the right most 20 bits denotes a index-number from where the collision outcomes are stored in the table. These collision outcomes start from index 2^{19}. The figure shows an example for an equivalence class of size 3. The three configurations are shown in (b),(c), and (d); (e) shows the histogram of the 29925 equivalence classes of the GBL model (largest class 257).

Collision tables The collision step consists of updating the lattice node with its collision outcome. In the 6 and 7 bits models, we use a collision table of words (16 bits) of size 2^6 and 2^7 respectively. In these models we have at most two collision outcomes, hence we store two 8 bits outcomes in each table entry and choose one randomly. If there is only one outcome, both values are equal.

The models which use more than 7 bits are implemented as equivalence classes. An equivalence class is formed by all states having the same mass, momentum, and (for thermal models) energy. For the 19-bits model this means a collision table of 2^{19} indices, followed by the equivalence classes, see figure 2(a). Every index of an element in a class points to the start of the class (138, 273 and 41024 all point to $2^{19}+X$). The left 12 bits are used to indicate the number of collision outcomes. If the number is zero, the outcome is equal to the input state. Otherwise the value of the right 20 bits is an index pointing to the first outcome possibility, followed by the other possible outcomes, of which we choose one at random. It is clear that the input state is also among them (meaning

no collision), but since most classes are quite large this has little influence (see figure 2(e)).

Color redistribution (diffusion) The problem at hand is how to distribute several color particles random over the collision outcome. We start with the collision outcome in which we find n ones, denoting the presence of the particles. This value is being used for the distribution of the colors. Now we have the first m particles of a certain color. We start from right to left, look for a one in the collision outcome, and set the particle at this color if a random number between 0 and 1 is smaller than m divided by n. If this is indeed the case we set the bit from 1 to 0 (the original collision outcome was stored for further use), and we continue with $n-1$ and $m-1$ (we now have to distribute $m-1$ over $n-1$). If the random number was larger than or equal to m divided by n we do not set the particle color and continue with $n-1$ and m (we now have to distribute m over $n-1$). Note that the procedure is correct for $m \geq n$, since than the particles will always be set to the color. The other colors are treated similarly, with one change: we now have to distribute r color particles over $n-m$ ones. Note that the number of random generated numbers depends on the number of ones in the collision outcome, i.e. the density.

Calculating $S(k,\omega)$ The dynamic structure factor $S(k,\omega)$ is treated as a spectral function (a function of ω at fixed values of the wave number k). Calculation of $S(k,\omega)$ first requires a spatial two dimensional Fourier transform of the density, which is done at simulation time, followed by a one-dimensional Fourier transform in time of the spatial Fourier transformed data, done after simulation time.

The spatial transform is performed every time step, and the wave numbers we are interested in are stored. The spatial transform results in a two-dimensional array, scattered over the processors. The first row of the array of the first process contains the wavenumbers $k_x = \text{index} \times \frac{2\pi}{L_x}$, the first column of the array (scattered over the processors) represents the wave numbers $k_y = \text{index} \times \frac{2\pi}{L_y} \times \frac{2}{\sqrt{3}}$, where L_x is the lattice size in x-direction, and L_y is the lattice size in the y-direction.

After the simulation we Fast Fourier Transform the stored wave numbers in time. To reduce the initially large variance we apply the technique discussed in [15]. We partition the data into K segments each of 16384 consecutive data points. Each segment is separately FFT'd to produce an estimate. We let the segments overlap by one half their length. Finally, the K estimates are averaged at each frequency, reducing the initial variance by $\sqrt{K}$. Data windowing is used to reduce "leakage". The $S(k,\omega)$ is scaled as $\int_{-\pi}^{\pi} S(\mathbf{k},\omega)\,d\omega = 2\pi S(\mathbf{k})$ [9]. Thus, the area under $S(k,\omega)/S(k)$ is 2π and the spectra shown in the next section are scaled accordingly.

4 Computational aspects

4.1 Profile analysis

A (sequential) profile of the code can be used to provide inside in the fraction of the execution time spent in a function, divided in four categories: collision step (13%), streaming step (6%), the Fast Fourier Transform (18%), and color redistribution (63%) (for a simulation of the GBL model at lattice size 512×512, reduced density 0.3, periodic boundary conditions in both directions, 50% red particles, 50% blue particles). We note the large fraction of the execution time spent in the color redistribution step, which consists almost entirely of random number generation.

The random number generator used is the *ran2* recommended in Numerical Recepices [15]. During this particular simulation we generate more than 10^{13} random numbers. This means that to prevent correlations we have to use a random number generator with a large period (the cycle of *ran2* is 10^{18}).

A solution for the low b-bit two-species LGA models is to construct one collision table of size 2^{2b} where the configuration consists of both place *and* color information. However, for multiple species and/or the GBL model this would result in a much too large collision table. Further improvements in speed should thus focus on a faster random number generator. A first optimization is the use of antithetic variables, i.e. if u is a random number (uniform $[0,\ldots,1]$) then so is $1-u$ [16].

4.2 Scalability results

To reduce the initially large variance in a spectral measurement, a large number of time steps is needed. For measurements at small values of $\mathbf{k}$ we have to use large lattice sizes. Parallel computing is exploited here to facilitate efficient simulation. The two factors controlling the efficiency of parallelization are the ratio between the communication time and calculation time, and the balance of workload among the processors. Since we have near-optimal load-balancing (square lattice grid and absence of obstacles in the fluid) only the first factor applies here. To measure scalability we have performed measurements of the execution time per iteration for a different number of lattice sizes at various number of processors. The LGA simulation data is a two-species GBL simulation of a fluid at global equilibrium, reduced density 0.3, periodic boundary conditions in both directions, 50% red particles, 50% blue particles. A spatial Fast Fourier Transformation is performed every time-step.

The results of the scalability measurements are shown in figure 3. For small lattice size the scalability is bad, due to the large communication overheads. However, for a more realistic choice of lattice size we see an almost perfect scaling.

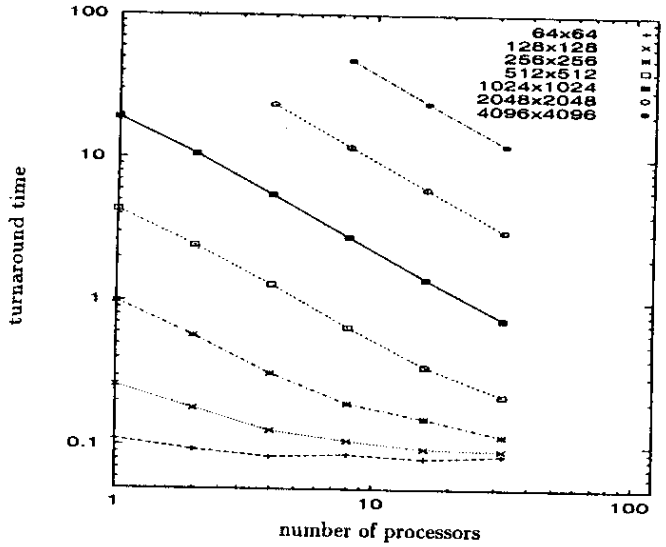


Fig. 3. The execution time in seconds per iteration for different lattice sizes (two-species GBL simulation of a fluid at global equilibrium, reduced density 0.3, periodic boundary conditions in both directions, 50% red particles, a spatial Fast Fourier Transformation is performed every time-step).

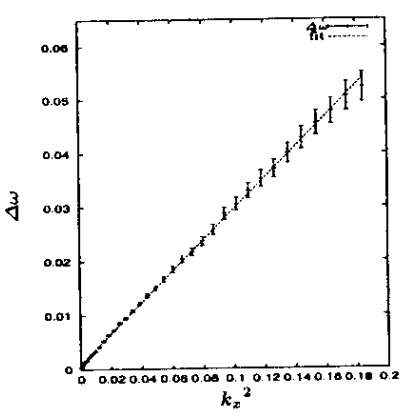
5 Case study

5.1 Transport coefficient measurements

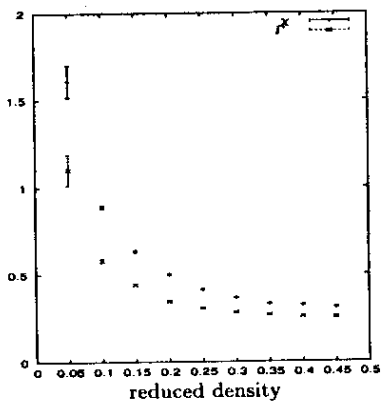
The dynamic structure factor $S(k,\omega)$ for real fluids in the Landau-Placzek approximation (small values of k and ω) reads [9]:

$$\frac{S(\mathbf{k},\omega)}{S(\mathbf{k})} = \left(\frac{\gamma-1}{\gamma}\right)\frac{2\chi k^2}{\omega^2+(\chi k^2)^2} + \frac{1}{\gamma}\sum_{\pm}\frac{\Gamma k^2}{(\omega \pm c_s k)^2 + (\Gamma k^2)^2} + \frac{1}{\gamma}[\Gamma + (\gamma-1)\chi]\frac{k}{c_s}\sum_{\pm}\frac{c_s k \pm \omega}{(\omega \pm c_s k)^2 + (\Gamma k^2)^2} \quad (3)$$

where $S(k)$ is the static structure factor. Here χ is the thermal diffusivity, $\Gamma = \frac{1}{2}[\nu + (\gamma-1)\chi]$ the sound damping, where ν is the longitudinal viscosity; c_s is the adiabatic sound velocity; and γ is the ratio of specific heats. A typical spectrum as found in real fluids consists of three spectral lines, a central peak (Rayleigh peak), which arises from fluctuations at constant pressure and corresponds to the thermal diffusivity mode, and two shifted peaks (Brillouin peaks), which arise from fluctuations at constant entropy and correspond to the acoustic modes.



(a) measuring Γ from a GBL spectrum.



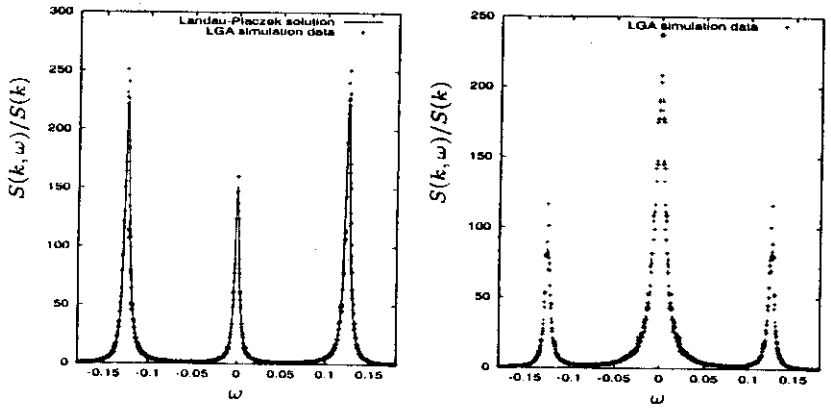
(b) χ and Γ for different reduced densities.

Fig. 4. Figure (a) plots $\Delta\omega$ of the Brillouin peaks (with error bars) as a function of k^2. This provides an experimental measurement of $\Gamma 0.288 \pm 0.001$ (value of the slope) at a reduced density of 0.3. Figure (b) shows the measurements (with error bars) of χ and Γ for different reduced densities.

Equation (3) for $S(k,\omega)$ is known as the Landau-Placzek approximation and also holds for the GBL model in the limit for small k and small ω [6,7]. A measured spectrum from a GBL simulation contains transport coefficient information. We can extract the sound damping Γ as the half-width of the Brillouin peaks: $\Delta\omega = \Gamma k^2$. We accomplish this by plotting $\Delta\omega$ as a function of k^2. A least-square fit provides the value of the slope, which is an experimental value of Γ. Figure 4(a) shows we find $\Gamma = 0.288 \pm 0.001$ for a reduced density of 0.3. The thermal diffusivity coefficient χ is obtained as the half-width of the central peak: $\Delta\omega = \chi k^2$. The experimental adiabatic sound velocity c_s. The ratio of the Brillouin peaks is a measurement of the specific heats γ is obtained as the ratio of the integrated intensity of the Rayleigh peak to those of the Brillouin peaks. Our experimental measurements of the transport coefficients from a GBL simulation are in agreement with other results [7]: $c_s = 1.275 \pm 0.001$, and $\gamma = 1.32 \pm 0.02$, and the dependence of Γ, χ on the reduced density is shown in figure 4(b).

5.2 The spectra of a two component thermal lattice gas

The $S(k,\omega)$ for a two-component system has a spectrum structure where it is difficult to separate the contributions from concentration fluctuations and



(a) 100% red particles

(b) 50% red particles

Fig. 5. The spectral function $S(k,\omega)$ for a (a) 100% and (b) 50% red GBL lattice gas ($k_x = 8 \times \frac{2\pi}{512}$, reduced density 0.3), data obtained from 1000000 timesteps.

entropy fluctuations [9]. The two contributions are independent if one of the two components is in trace amounts, but in general the two are not decoupled.

Figure 5(a) shows $S(k,\omega)$ for the standard GBL model. The simulation data obtained from the lattice gas simulation is fitted with the analytical solution, providing the transport coefficients in lattice units. Figure 5(b) shows $S(k,\omega)$ for the two-species GBL model. We notice a significant contribution from concentrations fluctuations to the amplitude of the Rayleigh peak.

6 Concluding comments

The parallel implementation of a lattice gas automata is straight forward, due to the inherent spatial locality. The GBL model uses velocities of two lattice spacing, hence we now have to use ghost-layers consisting of two lattice nodes for the implementation of the boundary conditions. The implementation of the collision step can simply be done using a collision table, where we obtain the collision outcome in at most two memory references and one random number generation.

Our extension of the model with multi-species, where the number of species is decided at compile time, requires a large amount of random numbers. The most widely used generators (like *random* and *drand48*) are not adequate. We recommend the use of the random number generator *ran2* as found in numerical recipes. A profile shows the random generator has become a large bottleneck. A first optimization is the use of antithetic variables.

The computation of the dynamic structure factor $S(k,\omega)$ requires a spatial Fourier transform at every time step. The parallel FFT expects a slice-decomposition of the lattice grid, but has otherwise no restrictions to the implementation. As an illustration we have shown how to calculate two transport coefficients, Γ (sound damping) and χ (thermal diffusivity), from the Rayleigh-Brillouin spectrum. These measurements were performed for different reduced densities. Finally, we presented a new result, namely the spectrum of a two-species GBL LGA. The spectrum clearly shows the added contribution of concentration fluctuations to the Rayleigh peak. It is in general not easily possible to derive the independent values of χ and D (the diffusion coefficient) from this spectrum.

References

1. D H. Rothman and S. Zaleski. *Lattice gas cellular automata.* Cambridge University Press, 1997.
2. D. d'Humières, P. Lallemand, and U. Frisch. Lattice gas models for 3-d hydrodynamics. *EuroPhys. Lett.*, 2:291–297, 1986.
3. K. Diemer, K. Hunt, S. Chen, T. Shimomura, and G. Doolen. Velocity dependence of reynolds numbers for several lattice gas models. In Doolen, editor, *Lattice Gas Methods for Partial Differential Equations*, pages 137–178. Addison-Wesley, 1990.
4. U. Frish, D. d'Humières, B. Hasslacher P. Lallemand, Y. Pomeau, and J.P. Rivet. Lattice gas hydrodynamics in two and three dimensions. *Complex systems*, 1:649–707, 1987.
5. U. Frish, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the navier-stokes equation. *Physical Review Letters*, 56:1505, April 1986.
6. P. grosfils, J.P. Boon, and P. Lallemand. Spontaneous fluctuation correlations in thermal lattice-gas automata. *Physical Review Letters*, 68:1077–1080, 1992.
7. P. grosfils, J.P. Boon, R. Brito, and M.H. Ernst. Statistical hydrodynamics of lattice-gas automata. *Physical Review E*, 48:2655–2668, 1993.
8. D. Hanon and J.P. Boon. Diffusion and correlations in lattice-gas automata. *Physical Review E*, 56:6331–6339, 1997.
9. J.P. Boon and S. Yip. *Molecular Hydrodynamics.* McGraw-Hill, New York, 1980.
10. M. Hénon. Isometric collision rules for the four-dimensional fchc lattice gas. *Complex Systems 1*, pages 475–494, 1987.
11. Copyright (C) 1998 Massachusetts Institute of Technology. The fastest fourier transform in the west. *http://theory.lcs.mit.edu.*
12. University of Chicago and Mississippi State University. Mpich. *http://www.mcs.anl.gov/mpich/.*
13. D. Kandhai, A. Koponen, A. Hoekstra, M. Kataja, J. Timonen, and P.M.A. Sloot. Lattice-boltzmann hydrodynamics on parallel systems. *Computer Physics Communications*, 111:14–26, 1998.
14. D. Kandhai, D. Dubbeldam, A.G. Hoekstra, and P.M.A Sloot. Parallel lattice-boltzmann simulation of fluid flow in centrifugal elutriation chambers. In *Lecture Notes in Computer Science*, number 1401, pages 173–182. Springer-Verlag, 1998.
15. W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1988.
16. S.M. Ross. *A course in Simulation.* Maxwell Macmillian, New York, 1991.