Cellular automata and Lattice Boltzmann Methods: a new approach to computational fluid dynamics and particle transport

Bastien Chopard and Alexandre Masselot [1,2]

CUI, University of Geneva CH-1211 Geneva 4, Switzerland

Key words: Simulation and Modeling; Cellular automata and lattice Boltzmann methods; Fluid flows; Particle erosion and deposition; Snow transport; Massively parallel computing;

Cellular automata (CA) and lattice Boltzmann LB approaches are computational methods that offer flexibility, efficiency and outstanding amenability to parallelism when modeling complex phenomena. In this paper, the CA and LB approach are combined in the same model, in order to describe a system where point-particles are transported in a fluid flow. This model is used to simulate the snow transport, erosion and deposition by the wind.

1 Introduction

Cellular automata (CA) are an idealization of a physical system in which space, time and the physical quantities (or state of the automaton) take only a finite set of values. Since they have been invented by von Neumann in the late 1940s, CAs has been applied to a large range of scientific problems [1–4].

A very important feature of CAs is that they provide simple models of complex systems. They exemplify the fact that a collective behavior can emerge from the sum of many simple components. Even if the basic and local interactions are perfectly known, it is possible that the global behavior of a system obeys

[1] Bastien.Chopard@cui.unige.ch
[2] Alexandre.Masselot@cui.unige.ch

new laws that are not obviously extrapolated from the microscopic properties. This fact makes cellular automata very interesting candidates to model physical processes and, in particular, to simulate complex and nonequilibrium systems.

Many phenomena in science can be studied at different levels of description [5]. For instance, at a microscopic level, a fluid can be viewed as a large collection of particles interacting in a complicate way whereas, at a macroscopic level, the same fluid is described by its local density $\rho(\mathbf{r})$ and local velocity field $\mathbf{u}(\mathbf{r})$. Of course, the macroscopic level is linked to the microscopic one by some coarse graining procedure and the explicit link can be worked out using concepts of statistical physics. The interesting point is that, in general, the macroscopic behavior depends very little on the microscopic details of the interactions and the most relevant features that remain after coarse graining are the conservation laws and basic symmetries of the interactions.

Using these symmetries and conservations, it is possible to devise a microscopic universe, made of fictitious components, easily programmed on a computer and which has the same macroscopic behavior as the real system [6]. In this way, a CA model is a fully discrete molecular dynamics where only simple (but essential) interactions are considered among fictitious particles moving on a regular lattice. Figure 1 illustrates this approach in the case of the first known realistic CA fluid, the so-called FHP model [7]. The first image shows a configuration of incoming particles at time t. The arrows indicate the direction of motion of each particle before it collides with the other particles entering the same site. All particles have the same velocity modulus but they may travel in different directions, as allowed by the topology of the lattice. The second image of figure 1 gives the new velocity distribution after collision. Note that mass and momentum are conserved during this step, which is the crucial ingredient of a real interaction. According to their new velocity direction, the particles jump to a neighbor site, and the new the particle configuration at time $t + \tau$ is shown in the last image of figure 1. The figure also illustrates how boundary conditions are easily implemented. The particles simply bounce back on a solid wall, in order to reproduce a no-slip (or viscous) boundary condition.

The LB approach [8] follows the same idea but describes this fictitious universe in terms of the probabilities of presence of the particles. Also, in the LB formalism, one replaces the explicit interaction between particles by a relaxation towards an ideal local equilibrium distribution (which represents the average output of an actual collision). LB models are numerically much more efficient than their CA counterparts. On the other hand, they may be numerically unstable and do not take into account the correlations that may be crucial in some problems [9,10]. The CA models are intrinsically stable and give a full description of a N-body dynamics, but they are statistically very noisy. The best choice between CA or LB depends on the application under study. Here

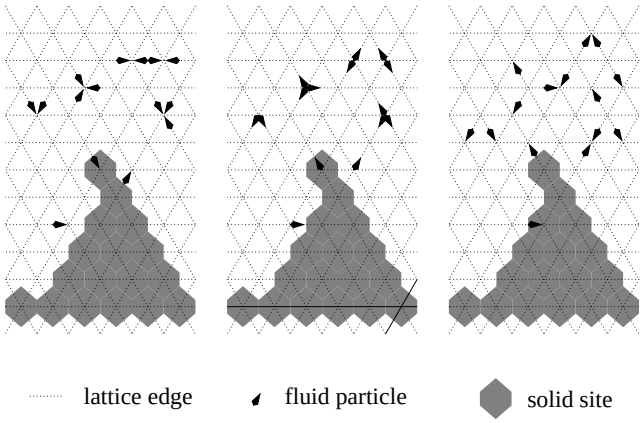

Fig. 1. Evolution of CA fluid particles on a hexagonal lattice (according to the FHP rules).

we shall mix the two approaches in order to model particles transport in a fluid flow.

The CA and LB methods should be viewed as an alternative to the classical techniques based on solving partial differential equations (PDE). PDEs have been, for a long time, the only and most tractable way to describe dynamical and spatially extended systems. As more difficult problems are considered, PDE may be less adequate and cannot always be formulated when complicated dynamics, involving thresholds or discontinuities, are studied. They may even fail to give to right answer to apparently simple problems [9]. Finally, the sophisticated numerical schemes used to solve PDE often screen out the nature of the process under study and prevent their generalization to new phenomena.

In this paper we consider the complex problem of simulating the formation of snow accumilations resulting form the erosion and transport by wind. Note that a similar approach can be devised to describe the formation of sand dunes. Our approach is based on the CA and LB methods and contrasts with other attempts to solve this problem by a PDE description [11]. A major difficulty when dealing with PDEs (the Navier-Stokes equation, in our case) is to account for the snow-wind binary mixture, one component being granular and the other a continuous media. In addition, boundary conditions are dynamically changing since, as the snow deposits grow, the wind pattern has to follow a new ground profile.

Phenomenologically, snow transport (*i.e.* erosion and deposition) has been divided in three main processes, each corresponding to a different scale [12]: (i) *Creeping:* describing the particles "rolling" on the snow surface or making very little jumps; (ii)*Saltation:* describing the snow particles ejected vertically and following a ballistic trajectory, in the first half meter above the surface [13]; (iii) *Suspension:* accounting for transport over larger scales (often seen as white smoke over mountains crests).

In our approach, these processes need not be explicitly defined. They naturally follow from the basic erosion, transport and deposition rules we shall defined at a microscopic level. Therefore, we end up with a much simpler and unified description of the phenomena, free of many of the technical complications encountered in the PDE approach and giving very promising results when the simulations are compared with real snow deposits.

This paper is organized as follows. Section 2 describes the LB technique used to model a fluid flow. Section 3 explains the rules we propose for adding snow particles on top of a LB fluid. Section 4 presents the results of the simulations and, finally, section 5 discusses some programming and performance issues for a parallel implementation of our application.

2 Lattice Boltzmann fluids

In a LB model, the physical quantities of interest are described in terms of fields $f_i(\mathbf{r}, t)$ defined for each point $\mathbf{r}$ of a lattice and each discrete time step $t = n\tau$. The index i labels the lattice directions (for instance, in a 2D square lattice, i runs from 1 to 8 and labels directions E, NE, N, NW, W, SW, S and SE, respectively).

In a hydrodynamical application, $f_i(\mathbf{r}, t)$ represents the average number of fluid particles entering site $\mathbf{r}$ at time t with velocity $\mathbf{v}_i$. The velocities $\mathbf{v}_i$ corresponds to a motion along lattice direction i so that, in one time step τ, one lattice spacing λ is traveled. A direction $i = 0$, characterized by $\mathbf{v}_0 = 0$ can be introduced, in order to describe a population f_0 of rest particles.

Macroscopic quantities like the density ρ or the momentum $\rho\mathbf{u}$ can be defined using the standard procedure of statistical mechanics, namely

$$\rho = \sum_i f_i \qquad \rho\mathbf{u} = \sum_i f_i \mathbf{v}_i$$

A Boltzmann equation [14] is a balance equation which describes how the populations f_i are re-distributed after an interaction. A general lattice Boltzmann equation reads

$$f_i(\mathbf{r} + \tau\mathbf{v}_i, t + \tau) - f_i(\mathbf{r}, t) = \Omega_i(f)$$

where $\Omega(f)$ is called the collision term. It is usually a nonlinear function of the $f_i(\mathbf{r}, t)$'s.

When the dynamics is based on specific "microscopic" interactions (as is the

case with FHP or other lattice gas automaton models [15]), an explicit expression for Ω can be obtained. However, the collision term can also be expressed in a more abstract way: in the so-called lattice BGK models[16], the dynamics of the functions f_i is governed by a LB equation where the interaction is simply given by a relaxation term

$$f_i(\mathbf{r} + \tau \mathbf{v}_i, t + \tau) - f_i(\mathbf{r}, t) = \frac{1}{\xi}\left[f_i^{(0)}(\mathbf{r}, t) - f_i(\mathbf{r}, t)\right] \qquad (1)$$

where $f_i^{(0)}(\mathbf{r}, t)$ is the so-called local equilibrium distribution and ξ the relaxation time. The function $f_i^{(0)}$ is the key ingredient of a LB model since it actually contains the properties of the physical process which is studied: this is the distribution to which the dynamics spontaneously relaxes and which is, therefore, intimately related to the nature of the system.

Hydrodynamic models assume a local equilibrium which is polynomial (quadratic in the local velocity field and linear in the local density of particles)

$$f_i^{(0)} = a\rho + \frac{b}{v^2}\rho \mathbf{v}_i \cdot \mathbf{u} + \rho e \frac{u^2}{v^2} + \rho \frac{h}{v^4} v_{i\alpha} v_{i\beta} u_\alpha u_\beta$$

This form is a discrete, truncated version of the Maxwellian distribution used in statistical physics. The coefficient of each term is adjusted so that mass and momentum are conserved and the Navier-Stokes equation reproduced [8,4].

The relaxation time ξ in eq. 1 is related to the fluid viscosity ν as $\nu = c_s^2\,(\xi - 1/2)$, where c_s^2 is the sound speed whose value is model dependent (different in hexagonal, square or cubic lattices [8]). Figure 2 illustrates the behavior of the LB fluid in a simulation of a flow past a plate, leading to a von Karman street pattern.

High Reynolds flows can be achieved if the viscosity is small enough, *i.e.* when $\xi \approx 1/2$. Unfortunately, numerical instabilities appears when ν decreases too much. To solve this problem, a subgrid technique has been proposed [17] in lattice BGK models, which consists of adjusting dynamically and locally the relaxation time ξ according to the local velocity gradients $\xi = \tau_0 + C\left[\partial_\beta u_\alpha + \partial_\alpha u_\beta\right]^2$, where C is the so-called Smagorinski constant and $\tau_0 = 1/2$ the equilibrium relaxation time. The indices α and β label spatial coordinates and summation over repeated indices is assumed. Having an effective relaxation time (and thus an effective viscosity) allows one to extrapolate unresolved scales and simulate high Reynolds number flows (up to 10^6)[17].

In this study we are interested in producing a turbulent flow in a semi-infinite space. The boundary at $z = 0$ is the ground level and the system has, in principle, no limit in the positive z direction (in practice, of course, we have

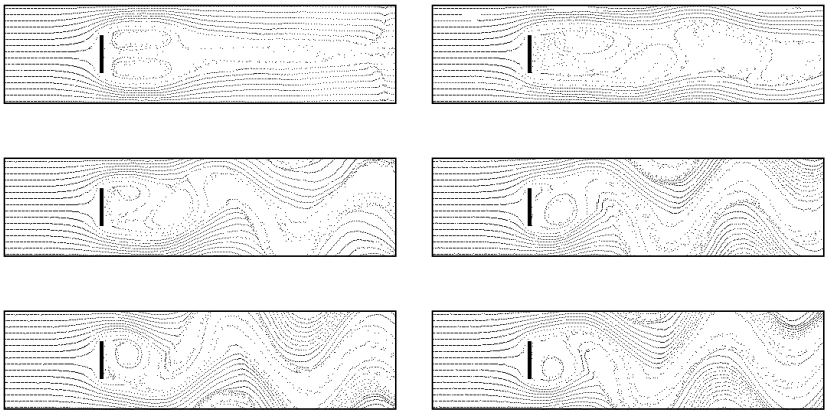

Fig. 2. *Non-stationary flow past a plate obtained from the 2-D lattice Boltzmann model. System size is* 512×128, $\xi = 1$. *and the fluid entry speed is* $u_\infty = 0.025$. *From left to right and top to bottom, the figure shows the different stage of evolution.*

to define a boundary which reproduces an open system). The wind blows parallel to the ground direction (the x-axis) and wind particles are accelerated rightward on the left boundary. The average wind speed profile in such a turbulent flow is described by the relation [18] $u_x(z) = u_* \log(z/z_0)$ up to some height z_{max} where it reaches the unperturbed wind speed u_∞. The quantity z_0 is dependent of the ground roughness, while u_* depends on the viscosity and u_∞.

The effect of the Smagorinski constant C is to adjust the resolution scale of the flow. A small C is appropriate to describe small developed eddies. Thus, in order to produce the correct velocity profile, which requires a finer resolution near the ground, we propose to make C depend on the height. It decreases linearly from a value C_∞ to 0 as the ground level is approached. This technique should be viewed as a way to define the proper boundary condition (change of the relaxation time due to an obstacle), ensuring a log profile for the velocity field $\mathbf{u}$. The distance over which C varies has been determined to be four lattice spacings in the simulations presented in section 4.

3 The snow model

Snow transport is obtained by adding a probabilistic CA model on top of the LB wind model. This CA model will give the rule of motion of point-like particles representing the snow flakes. As opposed to standard CA rules, we shall not impose here a "exclusion principle" limiting the number of particles on each site.

We define $N_i(\mathbf{r}, t) \in \{0, 1, 2, ..., \infty\}$ as the number of snow particles entering

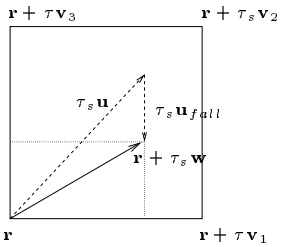

Fig. 3. Transport, on the lattice, of particles in suspension in a fluid. A particle at site $\mathbf{r}$ jump to east with probability $p_1(1-p_3)$, to north-east with p_1p_3, to north with $p_3(1-p_1)$ and stay at rest with probability $(1-p_1)(1-p_3)$; (see text).

site $\mathbf{r}$ at time t with velocity v_i. Snow particles can be injected in the simulation (snowfall) or eroded from the ground, deposited and transported according to the combined effect of gravity and local wind velocity. The rules we consider are:

Transport: an arbitrary number of snow grains may reside at each lattice site. During the updating step, they synchronously move to the nearest neighbor sites. Between times t and $t+\tau_s$, particles at site $\mathbf{r}$ should move to $\mathbf{r}+\tau_s\mathbf{w}$, where τ_s is the time step associated to solid particle motion, $\mathbf{w}=\mathbf{u}+\mathbf{u}_{fall}$, with $\mathbf{u}$ the local wind speed and $\mathbf{u}_{fall}$ the falling speed (accounting for gravity). Usually, $\mathbf{r}+\tau_s\mathbf{w}$ does not correspond to a lattice node (see fig. 3), and the amount of grains that reach each neighbor is computed according to the following randomized algorithm, which ensures that the average motion is correct. One computes $p_i=\max(0,(\tau_s/\tau)(\mathbf{v}_i\cdot\mathbf{w})/|\mathbf{v}_i|^2)$, for $i=1,3,5,7$ (if $p_i>0$, then $p_{i+4}=0$, since $\mathbf{v}_i=-\mathbf{v}_{i+4}$). For efficiency, we choose $\tau_s\geq\tau$, but small enough so that p_i is always less than 1. Then, each particle ℓ jumps to site $\mathbf{r}+\mu_1^\ell\mathbf{v}_1+\mu_3^\ell\mathbf{v}_3+\mu_5^\ell\mathbf{v}_5+\mu_7^\ell\mathbf{v}_7$, where μ_i^ℓ is a Boolean quantity which is 1 with probability p_i. If $N=\sum N_i$ is large enough, this binomial scattering can be approximated by a Gaussian distribution [19]. Note that in this algorithm, there is no attempt to include specific rules for creeping, saltation or suspension.

Deposition: lattice sites can be either free (air) or solid (original landscape or deposited snow). Snow particles on a free site may "freeze" if the neighbor site i they want to jump to is a solid site: $N_{frozen}\to N_{frozen}+N_i$, $N_i\to 0$. When N_{frozen} exceed some pre-assigned threshold N_s, the site becomes solid and subsequent incoming wind particle will bounce back (hence defining a new ground profile). This threshold gives a way to assign some size to the snow flakes. When a site solidifies the wind particles that may be present get trapped until erosion frees them again.

Erosion: deposited particles may be eroded under some conditions. For snow, the erosion rate seems to be related to the wind speed above the solid site [20], the concentration of snow being transported, the saturation concentration and the efficiency of the transport [12]. In our model, we express these mechanisms

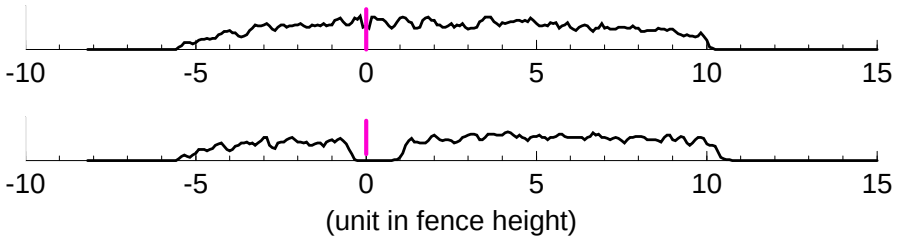

Fig. 4. Snow deposits around a fence without or with a ground clearance. Particles are introduced from the left, and the figures show the steady state of the system. In theses simulations, the lattice spacing is 0.2m.

in a very simple way: erosion means that each frozen snow particle is ejected upwards ($N_3 \rightarrow N_3 + 1$, $N_{frozen} \rightarrow N_{frozen} - 1$) with probability p. When the local wind is fast enough, these ejected snow particles will be transported, otherwise they fall back and freeze again.

4 Simulations

The above simple rules, when combined with the LB wind dynamics, are sufficient to produce realistic deposition patterns at different space scales by varying the Smagorinski constant C_∞, the threshold N_s and the erosion probability p [21,22].

Figure 4 shows the deposition pattern around a fence without or with a ground clearance (small gap at the bottom devised to increase the efficiency of the the fence by allowing the wind to flow through, thus preventing the fence to be rapidly buried by the deposited snow). Fences are commonly used as an obstacle for the wind in order to accumulate snow at some appropriate locations (e.g. windward a road to be protected). The resulting wind screening decreases the snow transport past the fence and makes the deposition possible. The accumulation patterns produced by our simulation are in good agreement with field observations.

Figure 5 illustrates the filling of a trench excavated in a large flat area. We may observe the non-intuitive deposition pattern and its time evolution where several growth regimes may be identified. Good qualitative agreement is observed between the model and reality [13], mainly for the first part of the experiment (growth of two deposition peaks) before the wind has slowed down in the outdoor experiment.

Figure 6 shows small scale patterns known as ripples occurring with both sand and snow transport. Ripples are mainly due to creeping transport. The ratio we find between the height and the spacing of the oscillations (called the

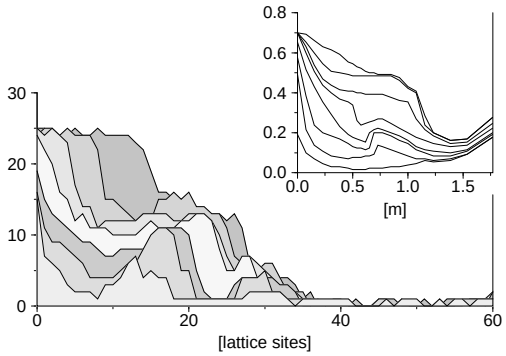

Fig. 5. Deposition pattern of snow in a trench (0.7m×1.7m, lattice spacing 0.03m, $N_s = 10$, $p = 0.04$ and $C_\infty = 0.3$). Snow particles are introduced on the left corner of the simulation; profiles are shown every 1000 iterations. The experimental profiles measured by Kobayashi [13] (given every 1/2 hour for the first layers) are sketched in the inset.

wave index) ranges around 6; this value agrees with the lowest index found for sand [23] in field observations, fits well wind tunnel experiments values [24] and sand ripples in water [23]. Outdoor snow ripples are more complicated since freezing and cohesion have to be taken into account; their wave index has been measured to be around 16 [12,25]. In agreement with real observations, we also see in our simulation that ripples move horizontally. This effect is illustrated in the figure. As observed in [26], our model also shows that large ripples can be built through the merging of smaller ones, traveling faster.

Our description of erosion and deposition uses three parameters: C_∞, N_s and p. Roughly speaking, these parameters give a way to select the characteristic scale of the patterns, but their precise role is still under investigation. If C_∞, N_s and p are badly chosen, the expected depositions pattern does not occur. For instance, ripples are obtained with large p and C_∞, because, as observed outdoor, they require a large snow mobility and weakly turbulent flow.

This model not only produces quantitatively realistic deposits, it also provides, through simple and intuitive rules, a better understanding of the basic (and quite controversial) mechanisms that occur in particle transport. Various patterns of deposition result from the emergence of a collective effect rather than from mechanisms that have not yet been identified. In a CA type of approach, creeping, saltation or suspension are no longer three phenomena requiring each a special treatment: they are all captured by the same erosion/transport mechanisms. Therefore, a unified view of the basic laws governing the formation of particle deposition pattern is gained.

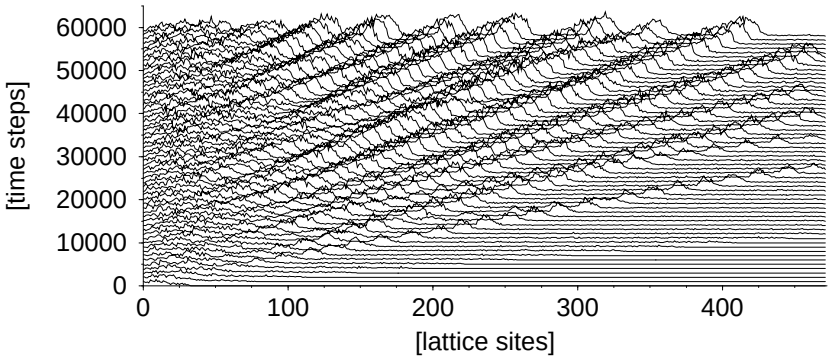

Fig. 6. Formation of ripples, as obtained from our model. Particles are continuously injected in the lower left corner of the simulation and the ripples grow spontaneously. The deposition profile is given every 1000 time steps, which makes the horizontal ripple motion quite clear (as well as the higher speed of the smaller ripples "escaping" rightward). The lattice spacing is around 0.03m, $N_s = 10$, $p = 0.02$ and $C_\infty = 5.0$.

5 Parallel implementation

The implementation of LB or CA models on a massively parallel computer is straightforward due to the intrinsic parallel and synchronous nature of these approaches. The motion of the fictitious particles is obtained by regular and local communications. The computation itself is local, uniform and rather simple, although the number of variables used to describe the system is larger than in a PDE approach. Finally, LB model lead to rather short codes, which improves the programmer's productivity and the ease the adapt a given model to a new problem.

The simulations described in this paper are two-dimensional and have been performed on a Connection Machine CM-200 (8192 1-bit processors and 256 FPU). The code has been written in CM-Fortran, a precursor of HPF. Data parallelism turns out to be a well suited programming model to implement the wind model. There is a good adequation between the language constructs and the numerical algorithm. For the snow algorithms, on the other hand, more flexibility (as that offered by a message passing programming model) would

be sometimes desirable.

Due to the good performance offered by parallelization, an interactive environment, with on-line visualization and dialog boxes, has been developed to tune on-line the simulation parameters and obtain a better understanding of the relevant processes occurring in this application. A batch mode is also possible, which allows a complete description of the various events needed to perform a full numerical experiment.

The simulation domain (typically of 512×64 lattice sites) is mapped on a square lattice, making 2D arrays a natural and efficient data structure. On each cell, we need (i) 9 floats for the wind densities (eight neighbors plus a rest quantity), (ii) 9 integers for the snow particles and (iii) 9 bits to get a local map of the surrounding solid cells.

The program execution can be summarized by:

```
Initialization
While t < nb.of.iterations
  Forall lattice sites do in parallel
    1. update the local map (9 bits) of the surroundings (as
       some neighboring cells may have frozen)
    2. compute the wind density and velocity
    3. compute the snow particles evolution, deposition and
       erosion
    4. compute the new wind particles distribution
    5. propagate snow and wind densities according to their
       direction of motion
  End Forall
  t=t+1
End While
```

A standard performance measure in CA simulations is the number of site updates per second: the value we obtain is roughly 8×10^4 sites per second for our current implementation on the Connection Machine. A scalability analysis (speed-up and efficiency) is difficult on such a machine because the number of processors is fixed. Note however that the communication time is quite small in comparison with the local computations.

Currently, 3D simulations are performed on a 14-node IBM SP2, using the MPI library. This machine offers much more memory space, which is obviously critical in 3D applications and, also, gives more flexibility than a SIMD architecture to equilibrate the work on each processor (as only a few lattice sites contain solid particles, the efficiency is less optimal for the snow step unless some kind of load balancing is performed or more clever data decomposition is considered).

Acknowledgments

We acknowledge financial support from the Swiss National Science Foundation and we would like to thank D. Issler and P. Gauer from IFENA (Davos) for useful discussions.

References

[1] G. Doolen, editor. *Lattice Gas Method for Partial Differential Equations.* Addison-Wesley, 1990.

[2] T. Toffoli and N. Margolus. *Cellular Automata Machines: a New Environment for Modeling.* The MIT Press, 1987.

[3] S. Wolfram. *Cellular Automata and Complexity.* Addison-Wesley, Reading MA, 1994.

[4] B. Chopard and M. Droz. *Cellular Automata Modeling of Physical Systems.* Cambridge University Press, 1998. to appear.

[5] Leo Kadanoff. On two levels. *Physics Today*, 39(9):7–9, September 1986.

[6] W. Daniel Hillis. Richard Feynman and the Connection Machine. *Physics Today*, 42(2):78, February 1989.

[7] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the Navier–Stokes equation. *Phys. Rev. Lett.*, 56:1505, 1986.

[8] Y.H. Qian, S. Succi, and S.A. Orszag. Recent advances in lattice Boltzmann computing. In D. Stauffer, editor, *Annual Reviews of Computational Physics III*, pages 195–242. World Scientific, 1996.

[9] S. Cornell, M. Droz, and B. Chopard. Some properties of the diffusion-limited reaction $nA + mB \to C$ with homogeneous and inhomogeneous initial conditions. *Physica A*, 188:322–336, 1992.

[10] B. Chopard, A. Masselot, and M. Droz. A multiparticle lattice gas model for a fluid. Application to ballistic annihilation. *Phys. Rev. (submitted)*, 1998.

[11] P.-A. Sundsbø. *Numerical modelling and simulation of snow drift.* PhD thesis, Narvik Institute of Technology, 1997. ISBN 82-471-0047-9.

[12] T. Castelle. *Transport de la neige par le vent en montagne: approche expérimentale du site du col du Lac Blanc.* PhD thesis, EPF Lausanne, Switzerland, 1995.

[13] D. Kobayashi. Studies of snow transport in low-level drifting snow. *Contributions from the Institute of Low Temperature Science*, Series A(24):1–58, 1972.

[14] P. Résibois and M. Leener. *Classical Kinetic Theory of Fluids*. John Wiley, 1977.

[15] U. Frisch, 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. Reprinted in *Lattice Gas Methods for Partial Differential Equations*, ed. G. Doolen, p.77, Addison-Wesley, 1990.

[16] Y.H. Qian, D. d'Humières, and P. Lallemand. Lattice BGK models for Navier–Stokes equation. *Europhys. Lett*, 17(6):470–84, 1992.

[17] S. Hou, J. Sterling, S. Chen, and G.D. Doolen. A lattice Boltzmann subgrid model for high Reynolds number flows. *Fields Institute Communications*, 6:151–166, 1996.

[18] D.J. Tritton. *Physical Fluid Dynamics*. Clarendon Press, 1988.

[19] B. Chopard, L. Frachebourg, and M. Droz. Multiparticle lattice gas automata for reaction-diffusion systems. *Int. J. of Mod. Phys. C*, 5:47–63, 1994.

[20] M. Takeuchi. Vertical profile and horizontal increase of drift snow transport. *Journal of Glaciology*, 26(94):481–492, 1980.

[21] A. Masselot and B. Chopard. A lattice Boltzmann model for particle transport and deposition. *Europhy. Lett.*, 1998. to appear.

[22] A. Masselot and B. Chopard. Snow transport and deposition: a numerical model. *J. of Glaciology*, 1998. submitted.

[23] R.P. Sharp. Wind ripples. *Journal of Geology*, 71:617–636, 1963.

[24] Martinez H. *Contribution à la modélisation du transportéolien de particules. Mesures de profiles de concentration en soufflerie diphasique.* PhD thesis, Université Joseph Fourier - Grenoble I, february 1996.

[25] V. Cornish. *Waves of sand and snow*. T. Fisher Unwin, London, 1914.

[26] B.Y. Werner and D.T. Gillespie. Fundamentally discrete stochastic model for wind ripple dynamics. *Phys. Rev. Lett.*, 71:3230–3233, 1993.