

Calculations Using Lattice Gas Techniques

*by Tsutomu Shimomura,
Gary D. Doolen,
Brosl Hasslacher,
and Castor Fu*

Over the last few years the tantalizing prospect of being able to perform hydrodynamic calculations orders-of-magnitude faster than present methods allow has prompted considerable interest in lattice gas techniques. A few dozen published papers have presented both advantages and disadvantages, and several groups have studied the possibilities of building computers specially designed for lattice gas calculations. Yet the hydrodynamics community remains generally skeptical toward this new approach. The question is often asked, "What calculations can be done with lattice gas techniques?" Enthusiasts respond that in principle the techniques are applicable to any calculation, adding cautiously that increased accuracy requires increased computational effort. Indeed, by adding more particle directions, more particles per site, more particle speeds, and more variety in the interparticle scattering rules, lattice gas methods can be tailored to achieve better and better accuracy. So the real problem is one of tradeoff: How much accuracy is gained by making lattice gas methods more complex, and what is the computational price of those complications? That problem has not yet been well studied. This paper and most of the research to date focus on the simplest lattice gas models in the hope that knowledge of them will give some insight into the essential issues.

We begin by examining a few of the features of the simple models. We then display results of some calculations. Finally, we conclude with a discussion of limitations of the simple models.

Features of Simple Lattice Gas Methods

We will discuss in some depth the memory efficiency and the parallelism of lattice gas methods, but first "we will touch on their simplicity, stability, and ability to

model complicated boundaries.

Computer codes for lattice gas methods are enormously simpler than those for other methods. Usually the essential parts of the code are contained in only a few dozen lines of FORTRAN. And those few lines of code are much less complicated than the several hundred lines of code normally required for two- and three-dimensional hydrodynamic calculations.

There are many hydrodynamic problems that cause most standard codes (such as finite-difference codes, spectral codes, and particle-in-cell codes) to crash. That is, the code simply stops running because the algorithm becomes unstable. Stability is not a problem with the codes for lattice gas methods. In addition, such methods conserve energy and momentum exactly, with no roundoff errors.

Boundary conditions are quite easy to implement for lattice gas methods, and they do not require much computer time. One simply chooses the cells to which boundary conditions apply and updates those cells in a slightly different way. One of three boundary conditions is commonly chosen: bounce-back, in which the directions of the reflected particles are simply reversed; specular, in which mirror-like reflection is simulated; or diffusive, in which the directions of the reflected particles are chosen randomly.

We consider next the memory efficiency of the lattice gas method. When the two-dimensional hydrodynamic lattice gas algorithm is programmed on a computer with a word length of, say, 64 bits (such as the Cray X-MP), two impressive efficiencies occur. The first arises because every single bit of memory is used equally effectively. Coined "bit democracy" by von Neumann, such efficient use of memory should be contrasted with that attainable in standard calculations, where each number requires a whole 64-bit word. The lattice gas is "bit democratic" because all that one

needs to know is whether or not a particle with a given velocity direction exists in a given cell. Since the number of possible velocity directions is six and no two particles in the same cell can have the same direction, only six bits of information are needed to completely specify the state of a cell. Each of those six bits corresponds to one of the six directions and is set to 1 if the cell contains a particle with that direction and to 0 otherwise. Suppose we designate the six directions by A,B,C,D,E,F as shown on the next page. We associate each bit in the 64-bit word *A* with a different cell, say the first 64 cells in the first row. If the first cell contains (does not contain) a particle with direction A, we set the first bit in *A* to 1 (0). Similarly, we pack information about particles in the remaining 63 cells with direction A into the remaining 63 bits of *A*. The same scheme is used for the other five directions. Consequently, all the information for the first 64 cells in the first row is contained in the six words *A*, *B*, *C*, *D*, *E*, and *F*. Note that all bits are equally important and all are fully utilized.

To appreciate the significance of such efficient use of memory, consider how many cells can be specified in the solid-state storage device presently used with the Cray X-MP/416 at Los Alamos. That device stores 512,000,000 64-bit words. Since the necessary information for 10^3 cells can be stored in each word, the device can store information for about 5,000,000,000 cells, which corresponds to a two-dimensional lattice with 100,000 cells along one axis and 50,000 cells along the other. That number of cells is a few orders of magnitude greater than the number normally treated when other methods are used. (Although such high resolution may appear to be a significant advantage of the lattice gas method, some averaging over cells is required to obtain smooth results for physical quantities such as velocity and density.)

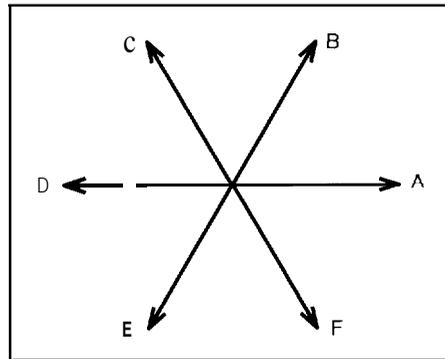
The second efficiency is related to the

fact that lattice gas operations are bit oriented rather than floating-point-number oriented and therefore execute more naturally on a computer. Most computers can carry out logic operations bit by bit. For example, the result of the logic operation AND on the 64-bit words *A* and *B* is a new 64-bit word in which the *i*th bit has a value of 1 only if the *i*th bits of both *A* and *B* have values of 1. Hence in one clock cycle a logic operation can be performed on information for 64 cells. Since a Cray X-MP/416 includes eight logical function units, information for 8 times 64, or 512, cells can be processed during each clock cycle, which lasts about 10 nanoseconds. Thus information for 51,200,000,000 cells can be processed each second. The two-dimensional lattice gas models used so far require from about thirty to one hundred logic operations to implement the scattering rules and about another dozen to move the particles to the next cells. So the number of cells that can be updated each second by logic operations is near 500,000,000. Cells can also be updated by table-lookup methods. The authors have a table-lookup code for three-dimensional hydrodynamics that processes about 30,000,000 cells per second.

A final feature of the lattice gas method is that the algorithm is inherently parallel. The rules for scattering particles within a cell depend only on the combination of particle directions in that cell. The scattering can be done by table lookup, in which one creates and uses a table of scattering results—one for each possible cell configuration. Or it can be done by logic operations.

Using Lattice Gas Methods To Approximate Hydrodynamics

In August 1985 Frisch, Hasslacher, and Pomeau demonstrated that one can approximate solutions to the Navier-Stokes equations by using lattice gas methods,



but their demonstration applied only to low-velocity incompressible flows near equilibrium. No one knew whether more interesting flows could be approximated. Consequently, computer codes were written to determine the region of validity of the lattice gas method. Results of some of the first simulations done at Los Alamos and of some later simulations are shown in Figs. 1 through 6. (Most of the early calculations were done on a Celerity computer, and the displays were done on a Sun workstation.) All the results indicate qualitatively correct fluid behavior.

Figure 1a demonstrates that a stable trailing vortex pattern develops in a two-dimensional lattice gas flowing past a plate. Figure 1b shows that without a three-particle scattering rule, which removes the spurious conservation of momentum along each line of particles, no vortex develops. (Scattering rules are described in Part II of the main text.)

Figure 2 shows that stable vortices develop in a lattice gas at the interface between fluids moving in opposite directions. The Kelvin-Helmholtz instability is known to initiate such vortices. The fact that lattice gas methods could simulate vortex evolution was reassuring and caused several scientists to begin to study the new method.

Figure 3 shows the complicated wake that develops behind a V-shaped wedge in a uniform-velocity flow.

Figure 4 shows the periodic oscillation of a low-velocity wake behind a cylin-

der. With a Reynolds number of 76, the flow has a stable period of oscillation that slowly grows to its asymptotic limit.

Figure 5 shows a flow with a higher Reynolds number past an ellipse. The wake here becomes chaotic and quite sensitive to details of the flow.

Figure 6 shows views of a three-dimensional flow around a square plate, which was one of the first results from Los Alamos in three-dimensional lattice gas hydrodynamic simulations.

Rivet and Frisch and other French scientists have developed a similar code that measures the kinematic shear viscosity numerically; the results compare well with theoretical predictions (see Fig. 8 in the main text).

The lattice gas calculations of a group at the University of Chicago (Kadanoff, McNamara, and Zanetti) for two-dimensional flow through a channel (Fig. 7 of the main text) agree with the known parabolic velocity profile for low-velocity channel flows.

The above calculations, and many others, have established some confidence that qualitative features of hydrodynamic flows are simulated by lattice-gas methods. Problems encountered in detailed comparisons with other types of calculations are discussed in the next section.

Limitations of Simple Lattice Gas Models

As we discussed earlier, lattice gas methods can be made more accurate by making them more complicated—by, for example, adding more velocity directions and magnitudes. But the added complications degrade the efficiency. We mention in this section some of the difficulties (associated with limited range of speed, velocity dependence of the equation of state, and noisy results) encountered in the simplest lattice-gas models.

The limited range of flow velocities is inherent in a model that assumes a

continued on page 210



Flow Past a Plate

Fig. 1a. Flow past a plate with periodic boundary conditions. This simulation, which was done in September 1985, shows vortices forming behind the plate. The average flow velocity has a magnitude of 0.2 lattice sites per time step and is perpendicular to the plate, pointing to the lower right. The direction of the flow velocity is color coded.

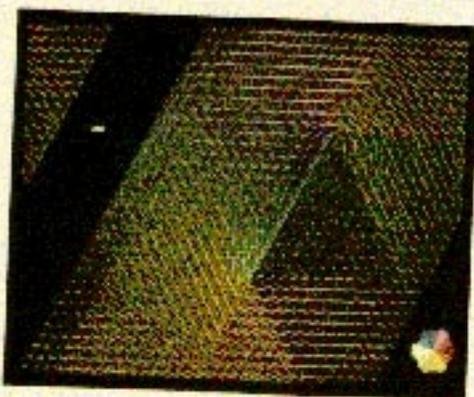
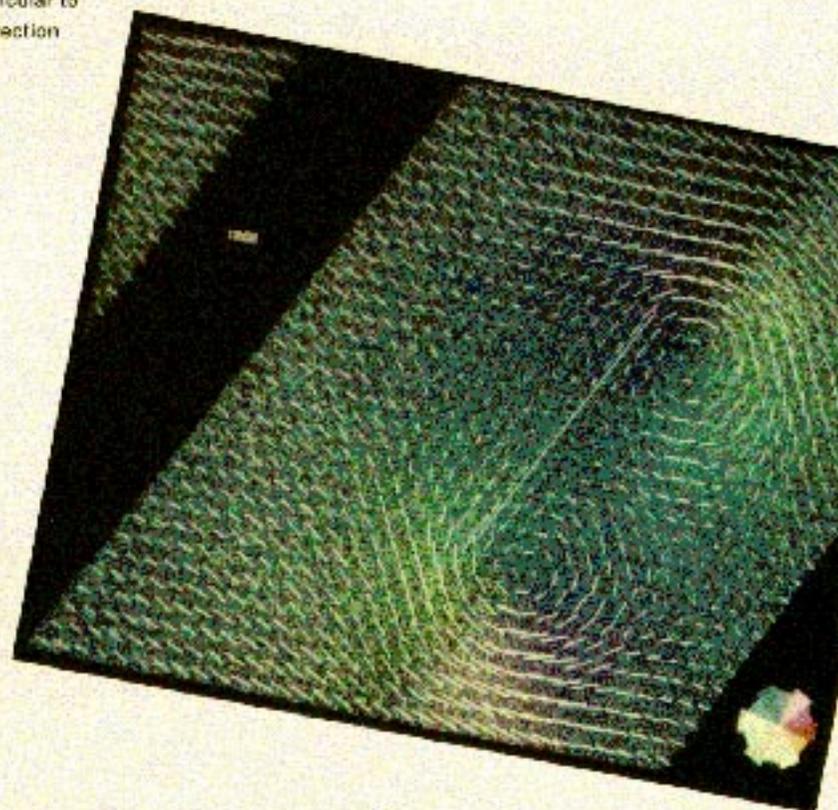
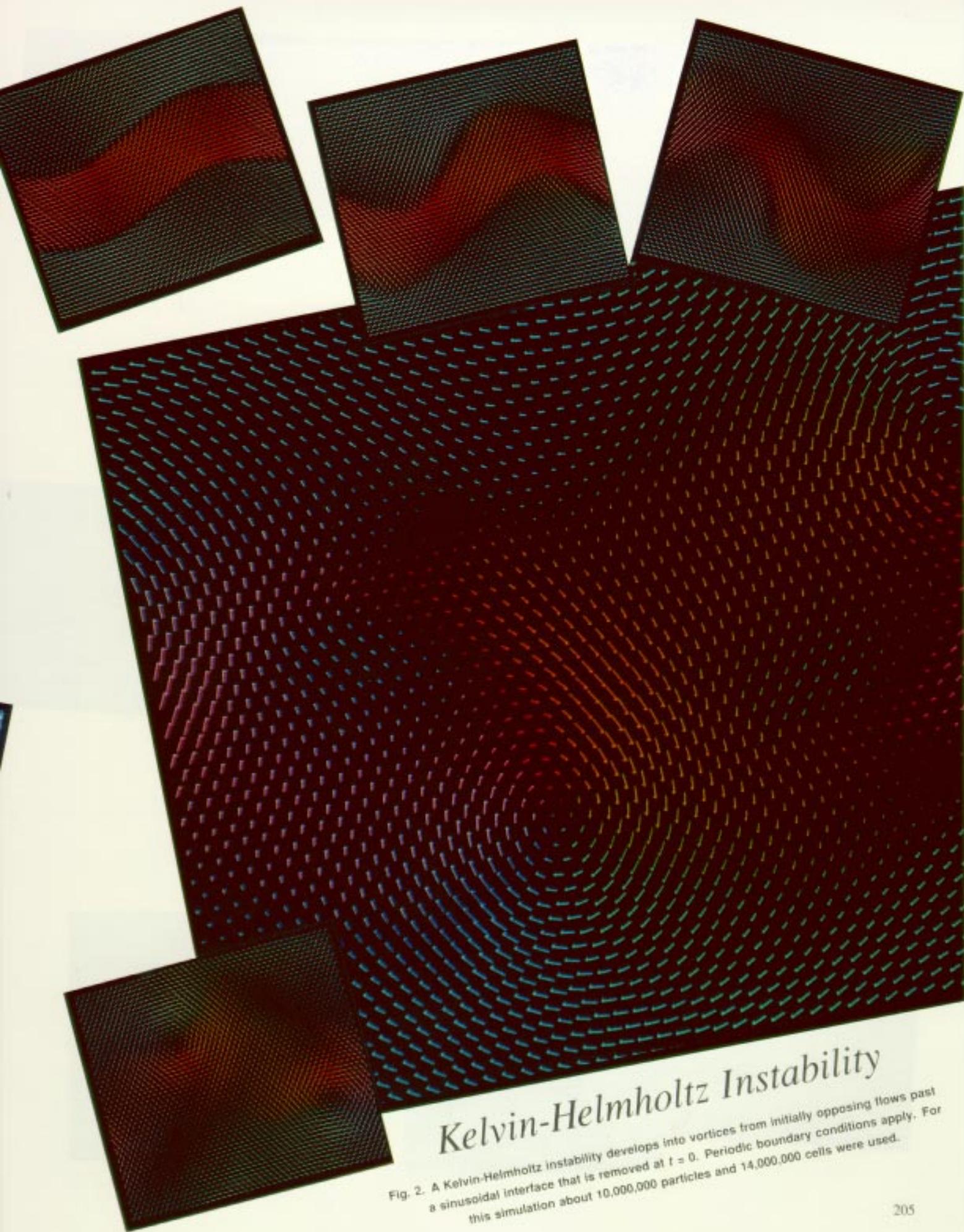


Fig. 1b. The same simulation as that described in Fig. 1a but with no three-body scattering rule. As a result, spurious laws of conservation of momentum along the lines of the grid prevent the development of hydrodynamics.

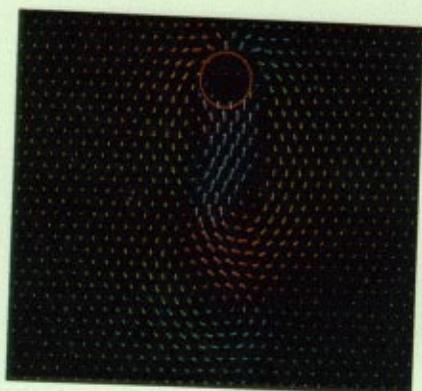


Kelvin-Helmholtz Instability

Fig. 2. A Kelvin-Helmholtz instability develops into vortices from initially opposing flows past a sinusoidal interface that is removed at $t = 0$. Periodic boundary conditions apply. For this simulation about 10,000,000 particles and 14,000,000 cells were used.



Fig. 3. A wake grows behind a wedge. The flow is from left to right with periodic boundary conditions. The flow is initialized as uniform flow to the right. The wedge is inserted at $t = 0$. Then vortices grow and are carried downstream. For this simulation 20 million particles and 16 million cells were used.



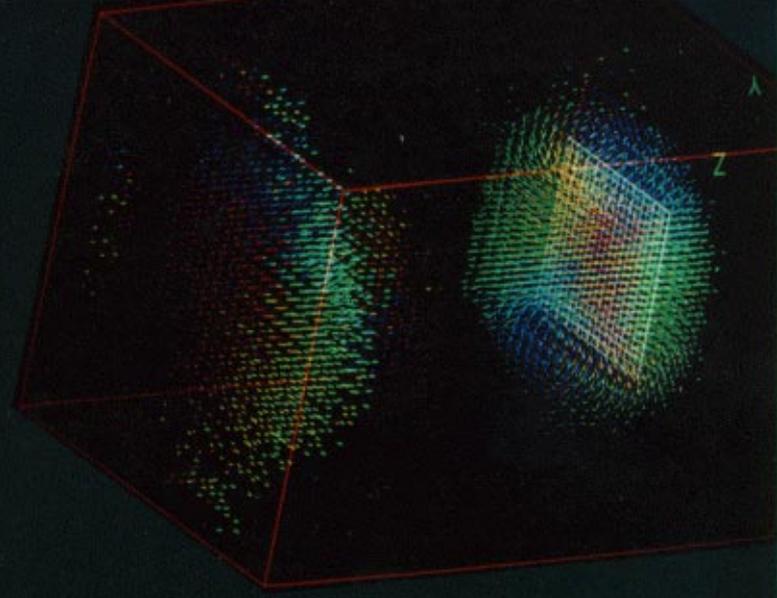
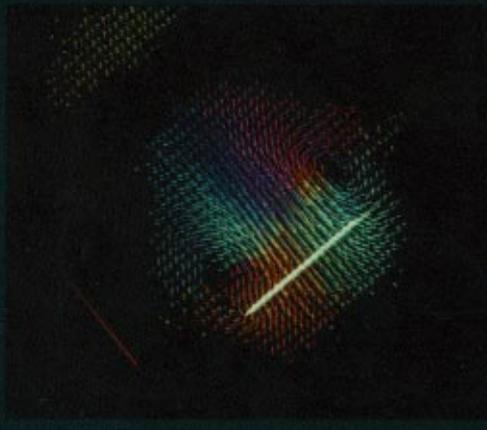
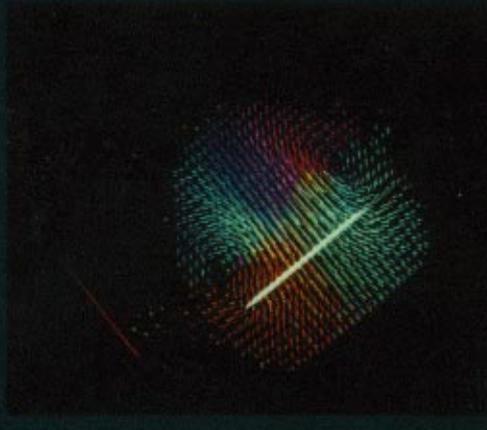
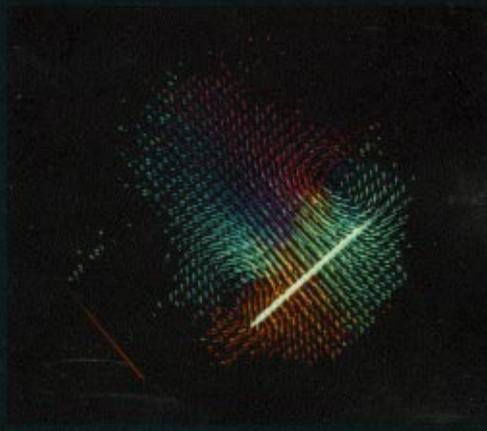
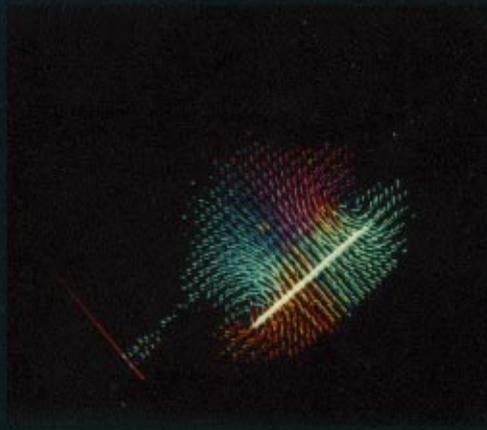
Flow Past a Cylinder

Fig. 4. Low-velocity flow (from top to bottom) past a cylinder creates a periodically oscillating wake. Four snapshots from one period of the oscillation are shown. In this simulation, which has periodic right and left boundaries, 1.4 million particles flowed through 1 million cells. The flow was initially uniform.



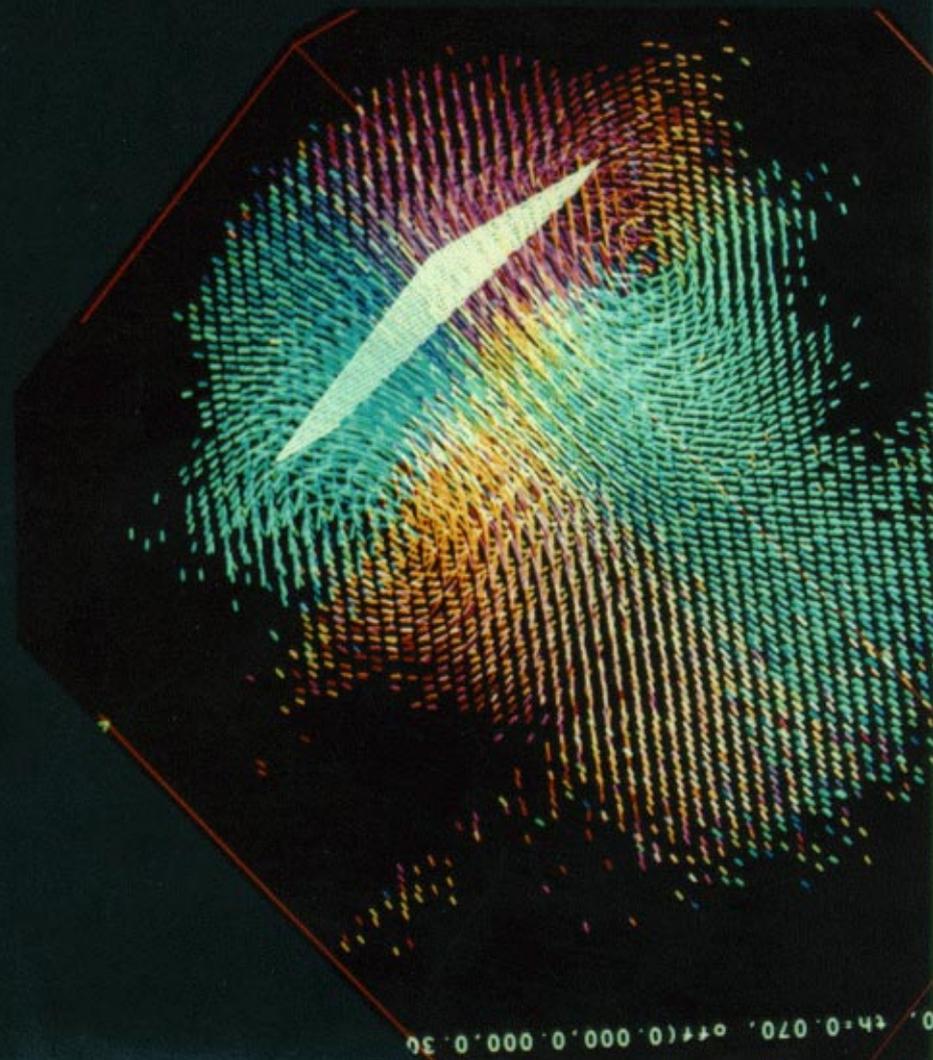
Turbulent Wake

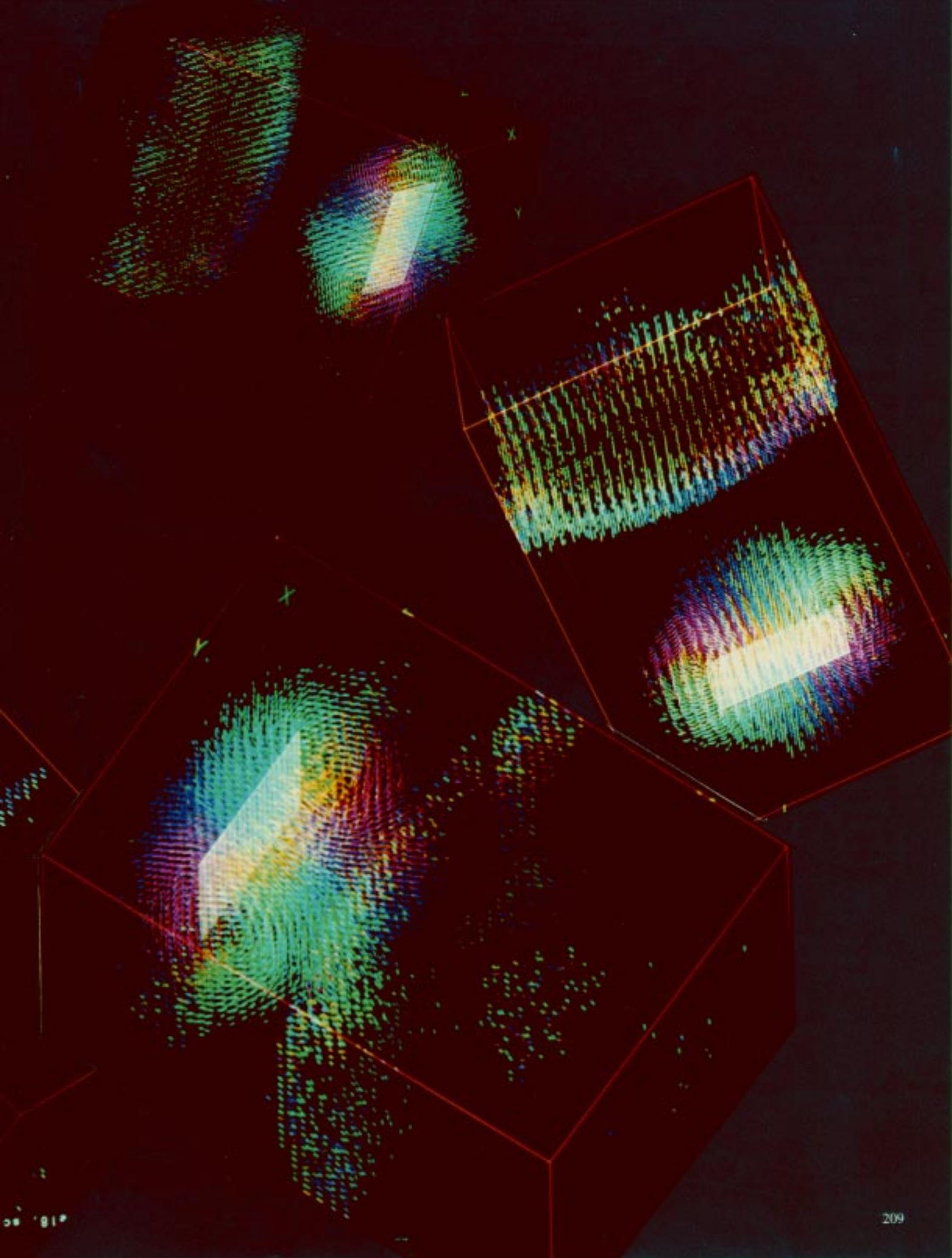
Fig. 5. A turbulent wake grows behind an ellipse being dragged through a fluid consisting of 11 million particles and 8 million cells. The ellipse is composed of about 2400 cells in which the velocity directions of the entering particles are reversed. The flow has periodic right and left boundaries. (An infinite sequence of equivalent ellipses exists to the left and right of the frame shown.) The Reynolds number in the flow is 1021.



3-D Flow Past a Plate

Fig. 6. Cross sections (left) and three-dimensional views show the development of vortices behind a square plate and a spherical sound wave that propagates through the system. Mean flow was subtracted to highlight the dynamics of the flow. For this simulation 80 million particles and 10 million face-centered-cubic cells were used.





continued from page 203

single speed for all particles. The sound speed in such models can be shown to be about two-thirds of the particle speed. Hence flows in which the Mach number (flow speed divided by sound speed) is greater than 1.5 cannot be simulated. This difficulty is avoided by adding particles with a variety of speeds.

The limited range of velocities also restricts the allowed range of Reynolds numbers. For small Reynolds numbers (0 to 1000) the flow is smooth, for moderate Reynolds numbers (2000 to 6000) some turbulence is observed, and for high Reynolds numbers (10,000 to 10,000,000) extreme turbulence occurs. Since the effective viscosity, ν , is typically about 0.2 in two-dimensional problems, the Reynolds number scales with the characteristic length, l , allowed by computer memory. Currently the upper bound on l is of the order of 100,000.

The velocity dependence of the equation of state is unusual and is a consequence of the inherent Fermi-Dirac distribution of the lattice gas (see the section on Theoretical Analysis of the Discrete Lattice Gas in the main text). The low-velocity equation of state for a lattice gas can be written as $p = \frac{1}{2}\rho(1 - \frac{1}{2}v^2)$, where p is the pressure, ρ is the density, and v is the flow speed. Thus, for constant-pressure flows, regions of higher velocity flows have higher densities.

The velocity dependence of the equation of state is related to the fact that lattice gas models lack Galilean invariance. The standard Navier-Stokes equation for incompressible fluids is

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\nabla p + \nu \nabla^2 \mathbf{v}.$$

But in the incompressible, low-velocity limit the single-speed hexagonal lattice gas follows the equation

$$\frac{\partial \mathbf{v}}{\partial t} + g(\rho) \mathbf{v} \cdot \nabla \mathbf{v} = -\nabla p + \nu \nabla^2 \mathbf{v},$$

where

$$g(\rho) = \frac{3 - \rho}{6 - \rho}$$

and ρ is the average number of particles per cell. The extra factor $g(\rho)$ requires special treatment. The conventional way to adjust for the fact that $g(\rho)$ does not equal unity (as it does in the Navier-Stokes equation) is to simply scale the time, t , and the viscosity, ν , by the factor $g(\rho)$ as follows: $t' = g(\rho)t$ and $\nu' = \nu/g(\rho)$. (The pressure must also be scaled.) Hence a density-dependent scaling of the time, the viscosity, and the pressure is required to bring the lattice gas model into a form that closely approximates the hydrodynamics of incompressible fluids in the low-velocity limit.

Finally, the discreteness of the lattice gas approximation introduces noise into the results. One method of smoothing the results for comparison with other methods is to average in space and time. In practice, spatial averages are taken over 64, 256, 512, or 1024 neighboring cells for time-dependent flows in two dimensions. For steady-state flows, time averaging is done. The details of noise reduction are complicated, but they must be addressed in each comparison calculation. The presence of noise is both a virtue and a defect. Noise ensures that only robust (that is, physical) singularities survive, whereas in standard codes, which are subject to less noise, mathematical artifacts can produce singularities. On the other hand, the noise in the model can trigger instabilities.

Conclusion

In the last few years lattice gas methods have been shown to simulate the qualitative features of hydrodynamic flows in two and three dimensions. Precise comparisons with other methods of calculation remain to be done, but it is believed that the accuracy of the lattice gas method

can be increased by making the models more complicated. But how complicated they have to be to obtain the desired accuracy is an unanswered question.

Calculations based on the simple models are extremely fast and can be made several orders-of-magnitude faster by using special-purpose computers, but the models must be extended to get quantitative results with an accuracy greater than 1 percent. Significant research remains to be done to determine the accuracy of a given lattice gas method for a given flow problem. ■

Note added in proof: Recently Kadanoff, McNamara, and Zanetti reported precise comparisons between theoretical predictions and lattice gas simulations (University of Chicago preprint, October 1987). They used a seven-bit hexagonal model on a small automaton universe to simulate forced two-dimensional channel flow for long times. Three tests were used to probe the hydrodynamic and statistical mechanical behavior of the model. The tests determined (1) the profile of momentum density in the channel, (2) the equation of state given by the statistical mechanics of the system, and (3) the logarithmic divergence in the viscosity (a famous effect in two-dimensional hydrodynamics and a deep test of the accuracy of the model in the strong nonlinear regime).

The results were impressive. First, to within the accuracy of the simulation, there is no discrepancy between the parabolic velocity profile predicted by macroscopic theory and the lattice gas simulation data. Second, the equation of state derived from theory fits the simulation data to better than 1 percent. Finally, the measured logarithmic divergence in the viscosity as a function of channel width agrees with prediction. These results are at least one order of magnitude more accurate than any previously reported calculations.