CELLULAR AUTOMATONS APPLIED TO GAS DYNAMIC PROBLEMS

Lyle N. Long
Lockheed - California Company
Burbank, California

Robert M. Coopersmith
Lockheed - California Company
Burbank, California

B. G. McLachlan
NASA - Ames Research Center
Moffett Field, California

AIAA 19th Fluid Dynamics, Plasma Dynamics and Lasers Conference
June 8-10, 1987/Honolulu, Hawaii
CELLULAR AUTOMATONS APPLIED TO GAS DYNAMIC PROBLEMS

Lyle N. Long *
Lockheed - California Company
Burbank, California 91520-7013

Robert M. Coopersmith **
Lockheed - California Company
Burbank, California 91520-7552

B. G. McLachlan ***
NASA - Ames Research Center
Moffett Field, California 94035

Abstract

This paper compares the results of a relatively new computational fluid dynamics method, cellular automatons, with experimental data and analytical results. This technique has been shown to qualitatively predict fluid-like behavior, however, there have been few published comparisons with experiment or other theories. Comparisons are made for a one-dimensional supersonic piston problem, Stokes First Problem, and the flow past a normal flat plate. These comparisons are used to assess the ability of the method to accurately model fluid dynamic behavior and to point out its limitations. Reasonable results were obtained for all three test cases. While this is encouraging, the fundamental limitations of cellular automatons are numerous. In addition, it may be misleading, at this time, to say that cellular automatons are a computationally efficient technique. Other methods, based on continuum or kinetic theory, would also be very efficient if a little of the physics were included.

INTRODUCTION

This paper describes a novel gas dynamic prediction method known as cellular automatons. Since this method treats the gas as a discrete set of particles, it is best described in terms of the kinetic theory of gases. In order to understand how the method compares with more complete theories of gas dynamics and to illustrate its limitations, a brief review of kinetic theory is presented. Those familiar with the kinetic theory of gases may find the reading of the following section unnecessary. Cellular automatons, in their present state, employ very crude approximations to the physics of gas dynamics. This becomes clear when the method is compared to more complete methods. After discussing kinetic theory, the cellular automaton technique is described. While cellular automatons have been shown to qualitatively predict fluid flows, there have been few quantitative comparisons to other methods or experiments. Quantitative comparisons are presented below.

THE KINETIC THEORY OF GASES

Gas dynamics can be modeled as a continuum or as a system of particles. The continuum approach is the most common, primarily because it is mathematically simpler and is valid for many fluid flows. However, the kinetic theory approach
has a wider range of validity than the continuum approach. The main limitations of kinetic theory are due to mathematical and numerical complexities, which cause difficulty in obtaining solutions.

One parameter that can be used to describe the range of validity of continuum theory is the Knudsen number, \( Kn = \frac{\lambda}{L} \), where \( \lambda \) is the mean free path and \( L \) is a characteristic length. If the Reynolds number, \( Re \), is based on the same characteristic length, then one can show that for hard-sphere molecules \( Kn \approx 1.5 \frac{M}{Re} \), where \( M \) is the Mach number.

In these relations, it is best to use a characteristic length based upon the flow gradients (Ref. 1), \( L = \frac{Q}{\left| \nabla Q \right|} \), where \( Q \) is a representative flow variable such as density, temperature, or velocity, and \( \nabla \) is the gradient operator. For Knudsen numbers larger than approximately 0.1, one must resort to the kinetic theory of gases because the continuum assumptions begin to break down. The Knudsen number limitation is only one breakdown criteria for the continuum equations. Other effects such as those due to three-body (or greater) collisions, diatomic or polyatomic gases, ionized flows, and quantum mechanical effects also cannot be modeled by the Navier-Stokes constitutive relations (Ref. 2, p. 275).

The equation that governs the kinetic theory of binary collisions is the Boltzmann equation (Ref. 3). Assuming no external forces, this equation is:

\[
\frac{\partial (nf)}{\partial t} + \vec{\nabla} \cdot (nf \vec{v}) = \int \int \int \int \int \int \int n^2 [f(\vec{v}^*) f(\vec{v}^*) - f(\vec{v}) f(\vec{v})] v \sigma d\Omega d\vec{v}
\]

where \( \vec{v} \) and \( \vec{v}^* \) are the pre-collision molecular velocities, \( v_r = |\vec{v} - \vec{v}^*| \) is the relative speed, \( \vec{v}^* \) and \( \vec{v}^* \) are the post-collision molecular velocities, \( \sigma d\Omega \) is the differential collision cross-section, and \( n \) is the number density. The right-hand-side of this equation is called the collision integral. For hard-sphere molecules of diameter \( d \), \( \int \sigma d\Omega = \pi d^2 \). The Boltzmann equation has only one unknown, the velocity distribution function \( f \), but for a single-species monatomic gas this is a function of seven variables \( f(\vec{r}, t, \vec{v}) \), where \( \vec{r} \) describes the spatial position, \( t \) is time, and \( \vec{v} \) is the molecular velocity. This distribution function represents the fraction of molecules in the volume between \( \vec{r} \) and \( \vec{r} + d\vec{r} \) that have a velocity between \( \vec{v} \) and \( \vec{v} + d\vec{v} \). The integral of \( f \) over all velocity space is unity.

One must remember that the Boltzmann equation does have limitations. It is limited to monatomic dilute gases, binary collisions, and molecular chaos. Thus the mean free path must be large compared to the molecular spacing, and the molecular spacing must be large compared to the molecular diameter. Phenomena such as recombination and catalytic reactions which involve the simultaneous collision of three bodies are also beyond the range of validity of the Boltzmann equation. In order to rigorously model these phenomena one must use more complete theories, such as the Liouville equation (Ref. 4), from which one can derive the Boltzmann equation.

The continuum conservation equations (mass, momentum, and energy) are obtained by multiplying the Boltzmann equation by \( m \), \( m \vec{v} \), and \( m \vec{v}^2/2 \), respectively (where \( m \) is the molecule mass), and then integrating over all velocity space. However, these contain more unknowns than equations, therefore the real difficulty comes in evaluating the relationships between the stress tensor, the heat transfer, and the other flow quantities. To derive these relationships one expands the distribution function in an asymptotic series in the Knudsen number:

\[
f = f_0 + Kn f_1 + Kn^2 f_2 + \ldots
\]

The Euler equations are the zeroth-order approximation (\( f_0 \)) to the Boltzmann equation, and are only valid in the limit as \( Kn \to 0 \). The Navier-Stokes equations
are the first-order approximation and are only valid for small Knudsen numbers because terms of order $Kn^2$ and above are neglected. Evaluating higher order terms of this expansion, as done by Burnett (Ref. 4), in order to obtain equations valid at higher Knudsen numbers is of limited usefulness since the above expansion is not uniformly valid.

Kinetic theory is more difficult mathematically than continuum theory primarily because of the complicated collision integral and the infinite degrees of freedom of the velocity distribution function. Numerical simulation requires an enormous number of computations. Another major difficulty is the amount of memory required to store all the discretized values of the seven-dimensional velocity distribution function $f(x,t,v)$. If one required 500,000 cells in physical space and 50 in each of the three molecular velocities directions, this would require $6.25 \times 10^{10}$ floating point numbers for each temporal value of $f$. At the limit of zero Knudsen number (translational equilibrium) the solution to the Boltzmann equation is the Maxwellian distribution:

$$f_0(x, t, v) = \left(\frac{m}{2\pi KT}\right)^{1.5} e^{-\frac{(v - \bar{v})^2}{2KT}}$$

where $\bar{v} = \bar{u}(x,t)$, $T = T(x,t)$, and $K$ are the macroscopic velocity and temperature, and Boltzmann constant, respectively. Figure 1 schematically illustrates three velocity distribution functions: equilibrium, nonequilibrium, and free molecular flow. They are intended to show how the distribution function varies as a function of Knudsen number. Only one dimension of $f$ is shown here, it actually depends on three molecular velocities. Actual computed distribution functions are presented in Reference 5. Quadrature techniques have a very difficult time integrating these functions because of the rapid growth and decay of the curves. The integrand in the collision integral is the product of two distribution functions and is extremely difficult to integrate accurately.

Ideally one would like to use spectral methods to represent the distribution function. Polynomial approximations are very poor even for Maxwellian distributions and even worse for nonequilibrium distributions with several maxima. Series expansions in exponential functions may be better representations of the behavior of the distribution function. Or one may be able to use a series expansion of different Maxwellian distributions, taking the Mott-Smith bi-modal distribution function idea (Ref. 6) to the limit. These approaches do not seem to have been attempted however.

Several solution procedures for the Boltzmann equation have been attempted with varying degrees of success. References 1, 3, and 7-14 include descriptions of the various methods. Numerous articles may also be found in the semi-annual Proceedings of the Rarefied Gas Dynamics conferences. There are basically two types of numerical approaches: particle simulations (Ref. 9) or numerical solutions to the Boltzmann equation (Ref. 10). The attempts to numerically solve the Boltzmann equation have not been as successful for a number of reasons, as described in References 9 and 10.

The most widely used particle simulation method is the Direct Simulation Monte Carlo (DSMC) method of Bird (Ref. 3, 7, 9, and 14). This method has had years of development and validation. The DSMC method does not actually solve the Boltzmann equation directly. However, this technique and the Boltzmann equation are derived in a similar manner. In fact, the DSMC as implemented is more general than the Boltzmann equation. It has been used to model chemical reactions, three-body collisions, catalytic reactions, radiation effects, and ionized flows. In the DSMC technique simulated gas molecules are followed through a series of motion and collisions. The technique has been very successful despite its simplicity, and the fact that one does not model the actual number of molecules. Although the
DSMC method has typically been applied to transitional or rarefied flows, with the ever-increasing power of computers, it may be used more and more for continuum flows. The primary disadvantage of this technique is the large amount of computer time required by it. Therefore it is of interest to pursue alternative methods that may offer advantages in terms of computational efficiency.

CELLULAR AUTOMATONS

A relatively new class of computational fluid dynamics methods known as cellular automatons (also as lattice gases or discrete velocity methods) have been receiving more attention recently (Ref. 15-18). These methods also treat the gas as a system of particles; so they are best classified as kinetic-theory-based methods. These methods were originally pursued several years ago (Ref. 19 and 20).

Cellular automatons are simply point-mass particles that can assume only a limited number of speeds and spatial locations. Usually, they travel at the same speed directly from one node to another node on a uniform grid. Whereas in an actual gas the particles have an infinite range of possible speeds (0 to \(\infty\)), the speed of cellular automatons is quantized. The model essentially tries to replace the velocity distribution by a few Dirac delta functions. It is similar to the DSMC method in this regard, but the DSMC method uses the full infinite (within the constraints of floating point numbers) range of molecular speeds and directions.

The particles in an actual gas are free to travel to any point in the flow field as long as they do not pass through a solid. The cellular automaton particles can only travel down certain paths in space. This is analogous to a billiard table where the balls can only travel along a crisscrossed array of tracks. The curious thing about this model is that it can be reduced to a set of rules such that very few floating point operations need to be performed. That is, given the particles coming into a node, there is only a small number of outgoing states (after their collision) that conserve mass, momentum, and energy.

Cellular automatons do not have to be limited to single speeds or uniform grids, but their complexity grows rapidly as additional degrees of freedom are added. As the number of paths going into every node and the number of molecular speeds goes to infinity, the method would approach modeling the Boltzmann equation for hard-sphere point-mass molecules and a 2-D gas. Modeling molecules other than hard spheres may not be possible without reducing the advantages of the method.

If one uses a large number of cellular automatons, the macroscopic flow field appears qualitatively similar to a real fluid flow field. However, this does not necessarily mean that the method can produce accurate quantitative predictions of gas dynamics because the details of the intermolecular collisions can dramatically affect the structure of real flow fields (in particular the transport coefficients). In a real gas the molecules are surrounded by force fields that depend upon their molecular structure. The particles do not collide like billiard balls, they exert forces on each other without coming in contact. These include attractive as well as repulsive forces in a real gas.

While cellular automatons can simulate the stress tensors of the Navier-Stokes equations, there are several other aspects of gas dynamics that cannot be modeled well at the present time. In particular doing work on the gas or adding heat may not be possible with the present model. In a single-speed gas, the particles cannot gain or lose energy at a solid boundary. Also, the method will have an unrealistic viscosity-temperature relationship, which is very sensitive to the type of collision model used, for example:
\[
\begin{align*}
\mu &\propto \sqrt{T} & \text{Hard-Sphere Molecules} \\
\mu &\propto T & \text{Maxwell molecules} \\
\mu &\propto T^{1.5}/(T + \text{Constant}) & \text{Sutherland Molecules}
\end{align*}
\]

While the Sutherland formula (Ref. 4) models the behavior of air very well over a limited range of temperatures, it is not a good model for low temperatures or for gases such as helium or hydrogen (Ref. 21, p. 223). The hard-sphere model has an unrealistic viscosity-temperature relation for most gases.

The cellular automaton used here is based upon the hexagonal lattice proposed by Frisch et al (Ref. 17), with a few modifications. Reference 17 uses only 2 or 3 particle collision rules. This is a justifiable assumption for many conditions encountered in gas dynamics. However, the lattice gas model could have up to 7-particle collisions because of its design. In fact, if all the nodes have 7 particles (1 stationary), the system could represent a solid. Therefore in the present technique no collision types are ruled out. The outcome of a collision is randomly chosen from all the possible outcomes that conserve mass, momentum, and energy. These collision rules should not be confused with the complicated collision models (e.g. inverse power law, Leonard-Jones, etc.) used in kinetic theory. The collision rules only dictate the outcome of a collision of point-mass particles.

A typical grid is shown in Figure 2. The flow field is divided into cells as in finite difference techniques, but within each cell is a finer hexagonal lattice. Regular rectangular lattices cannot simulate the Navier-Stokes stress tensors using a single-speed gas (Ref. 17). The particles at a given node are allowed to travel down any of the six paths connected to that node. This lattice (system of nodes) is contained within a single cell of the grid. A cell is quite arbitrary and is only used as a region over which the macroscopic flow quantities are calculated. In fact, the solution can be restarted at any time and the region that are averaged over can be changed. This is unlike the DSMC method which stores the location and velocity of each particle in the computer memory. In this approach the information concerning a particular point in space is stored in memory, and that node can be queried at any time to determine how many particles are located there and what their velocities are.

A macroscopic fluid quantity such as pressure, density, or velocity is obtained by averaging over a large number of nodal values (e.g. 10x10 or 100x100). This averaging is somewhat similar to the Monte Carlo procedures where the molecular properties in a cell are averaged to give the macroscopic flow quantities within that cell. The statistical scatter in the macroscopic quantities is inversely proportional to the square root of the sample size. However, the more nodes one averages over, the more the flow gradients are smeared out. Ideally one should ensemble average the flow properties over small regions by averaging over several different runs of the same numerical experiment. Or for steady-state problems one could average the flow quantities over a long period of time. As in Monte Carlo methods, the cellular automaton approach will not be effective for small-perturbation problems because the perturbations to the flow field will be lost in the statistical noise.

The only types of collisions that change the incoming and outgoing states are shown in Figure 3, which shows the six paths into a node. There is another set identical to these except that they also have stationary particles. Each box in the figure shows a group of nodal states with the same mass, momentum, and energy. Thus after each time step the outgoing state is simply chosen randomly from the group which contains the incoming state. If five or six particles enter a node at the same time, the only outgoing state possible corresponds to no collisions, i.e., the particles pass through the node unaffected.
Figure 1. Schematic Illustration of Velocity Distribution Functions: Equilibrium, Non-Equilibrium, and Rarefied.

Figure 2. Cellular Automation Grid and Lattice.

Figure 3. All Possible Hexagonal Lattice Collision Rules.
Free-stream boundary conditions are implemented by forcing the distribution function to a fixed value at the outer boundaries. At solid surfaces the particles are assumed to reflect diffusely. That is, if a particle encounters a solid object it randomly reflects to one of the fluid nodes adjacent to the solid. This enforces the no-slip condition on the surface. In this study, specular or slip conditions were not investigated.

The simplicity of the lattice gas model comes from the fact that at the end of each numerical time step the particles are always at a node. All of the nodes in the lattice are exactly the same distance apart and the time step is chosen such that $\Delta x = v \Delta t$; where $\Delta x$ is the distance between the nodes, $v$ is the (constant) particle speed, and $\Delta t$ is the time step size. The method then becomes simply a set of Boolean operations that are applied after each time step to account for collisions. Therefore very few floating point operations are required. The state of the particles at a given node can be represented by seven logical bits (one additional bit can be used to denote whether or not the node is on a solid body). Therefore the amount of storage required at each node is very small when compared to typical finite difference codes which require many floating point numbers per cell. However, one must keep in mind that finite difference solutions to the continuum equations have cells that are very large compared to the mean free path. The nodes in a lattice gas are separated on the order of a mean free path. Therefore, while the computer memory requirements per node are small, the memory required for a given problem will depend on how many mean free paths are included in the flow field. In addition, the time step will be on the order of the mean molecular collision time.

The lattice gas also presents a number of new challenges. For example, most current supercomputer compilers vectorize floating point calculations, but not logical operations. Thus this method, as programmed here in Fortran, only runs about 3-11 times faster on a Cray-XMP than on a VAX-8600. The method is particularly well suited to massively parallel machines such as the Connection Machine produced by Thinking Machines, Inc. However, explicit numerical method for the continuum equations are also well suited to this type of computer. In addition, complex shapes would have to be modeled by "jagged" surfaces to keep all the cells the same size; and the surface roughness elements would be on the order of a mean free path. One particular limitation of the lattice gas model is that the nodes must all be equally spaced and cannot be concentrated in regions of high gradients as in standard finite difference calculations. Wolfram (Ref. 18) has suggested using node spacings that are not uniform, but in order to have all the particles at nodes after every time step some particles would essentially be traveling faster than others. The physical meaning and consequences of this still needs to be evaluated.

In a true 2-D gas such as we have here, the particles do not have all the degrees of freedom that 3-D gases have (i.e. they can only move in 2 directions). For a gas composed of point-mass particles and $N$ degrees of freedom, the ratio of specific heats is $\gamma = (N+2)/N$ (see Ref. 21, p. 42). Therefore, for a monatomic 2-D gas, such as the lattice gas used here, $\gamma = 2$. The flows produced by this method are for a 2-D gas, not simply 2-D flows. In the 2-D DSMC method the third velocity component is always carried along in order to simulate a real 3-D gas. For a monatomic, 3-D gas $\gamma = 5/3$. For air at sea level, $N = 5$ and therefore $\gamma = 1.4$.

As Bird points out (Ref. 7), an important parameter in kinetic theory is the ratio of the molecular spacing to the mean free path. For air this ratio is approximately 0.05 and 0.0000003, at 0.0 and 100.0 km, respectively. In using the cellular automaton method, one must also try to match this in order to compare to experiment.
The constant speed particle assumption is not easily justified, especially for flows with heat transfer effects. Temperature is defined here by adopting the standard practice in kinetic theory of assuming it is proportional to the kinetic energy of the molecules in a frame of reference fixed to the local flow velocity (Ref. 3):

$$3 \ R \ T = \iiint (\vec{v} - \vec{u})^2 f(\vec{x}, \ t, \ \vec{v}) \ dv\eta$$

$$\vec{v} = \ -\infty$$

For gases in equilibrium, this can be equated to the thermodynamic temperature. For a 2-D gas one must use '2 R T' in place of '3 R T' to account for the reduced number of degrees of freedom, assuming the equipartition of energy.

In the 2-D hexagonal lattice gas, only six directions for the molecular velocities are possible. These velocities can be written

$$\vec{v}_k = v \sin(\frac{2 \pi k}{6}) i + v \cos(\frac{2 \pi k}{6}) j \quad (k = 1, 2, \ldots, 6)$$

where \(i\) and \(j\) are unit vectors in the \(x\)- and \(y\)-directions. The zero-speed particles are given by \(v_7 = 0\). Therefore the distribution function, instead of being a function of a continuous spectrum of velocities, depends on only seven velocities:

$$f(\vec{x}, \ t, \ \vec{v}) = f(\vec{x}, \ t, \ \vec{v}_k) = f_k(\vec{x}, \ t) \quad (k = 1, 2, 3, \ldots, 7)$$

Also, since only one particle can occupy a given path into a node, the values for \(n \times f_k\) are either 0 or 1. Thus computer programs based on this method typically use logical bits to represent the states. For dense gases this is an efficient means of storing the flow field information. However, in an actual fluid there is a vast amount of space between molecules. Therefore, if one is interested in rarefied flows where the particles are not closely spaced, many nodes will contain logical zeros because there are no particles at them. Since the integral of the distribution function over all velocities must be unity (the probability that the velocity lies between plus and minus infinity), the sum of the \(f_k\)'s must equal 1. Therefore the temperature is

$$2 \ R \ T = \sum_{k=1}^{7} f_k |\vec{v}_k - \vec{u}|^2 = \sum_{k=1}^{7} f_k |\vec{v}_k|^2 - u^2$$

and since \(v_1, \ldots, v_6 \equiv v\) and \(v_7 \equiv 0\), and the sum of the \(f_k\)'s is unity, one obtains

$$2 \ R \ T = (1 - f_7) v^2 - u^2$$

Adding additional molecular speeds to the model would allow better modeling of temperature. The fraction of particles with speed zero \(f_7\) will change throughout the flow, even though these particles remain in their original locations. Using the integral formula definition of macroscopic velocity

$$\bar{u}(\vec{x}, \ t) = \iiint \vec{v} \ f(\vec{x}, \ t, \ \vec{v}) \ dv\eta$$

the macroscopic velocity in the lattice gas is

$$\bar{u} = v \ i \ \sum_{k=1}^{6} f_k \ \sin(\frac{2 \pi k}{6}) + v \ j \ \sum_{k=1}^{6} f_k \ \cos(\frac{2 \pi k}{6})$$

where throughout this paper \(\bar{u}\) refers to macroscopic velocities and \(\vec{v}\) refers to
particle velocities. Note that the $k=7$ state need not be considered because it has zero velocity. The above reduces to

$$
\tilde{u} = \frac{\sqrt{3}}{2} \nabla \cdot \left( f_1 + f_2 - f_4 - f_5 \right) i + \frac{\sqrt{3}}{2} \left( f_1 - f_2 - 2f_3 - f_4 + f_5 + 2f_6 \right) j
$$

Notice that if all the $f_k$'s are equal, the flow velocity is zero, as it should be. If $f_6=1$, then $u = v = 0$. This is essentially an infinite Mach number, since the temperature in this case is zero. It is interesting to note that the maximum Mach number attainable in the $i$-direction is only 2.0, which corresponds to $f_1=f_2=0.5$.

Reynolds Number

In order to determine the Reynolds number, one must calculate the viscosity. Wolfram (Ref. 18) and Broadwell (Ref. 19) both estimate the viscosity of lattice gases for hexagonal and rectangular lattices, respectively. In this paper we will simply use the approximate Chapman-Enskog relationship (Ref. 3):

$$
\mu = \frac{(5\pi/32) \rho \bar{v} \lambda_0}{p_0}
$$

where $\mu$, $\rho$, $\bar{v}$, and $\lambda_0$ are the viscosity, density, mean thermal speed, and mean free path, respectively. Broadwell (Ref. 19) shows that this is a fair assumption for the rectangular lattice gas. The mean thermal speed and mean free path are determined in the classical manner (Ref. 3) from the freestream lattice gas properties.

Another limitation of this method is the large number of nodes required to achieve high Reynolds numbers. The Reynolds number of a particular flow is proportional to the number of nodes per characteristic length. For the flat plate examples to be shown later on, a grid of 1200 x 800 (960,000) nodes were used with 200 nodes on the flat plate. The Reynolds number for these flows was estimated to be 137. This means that in order to model flow at a Reynolds number of 1 million (at the same density and Mach number), one would need approximately 1.5 million nodes per characteristic length. Clearly the number of mathematical operations performed per node must be small to calculate this on today's computers. This also illustrates why massively parallel computers (especially those with a very large number of processors) are extremely effective for this approach. These methods will not be used to design automobiles or airplanes, as suggested in Reference 22, without major advances in both the method and computer hardware.

Equilibrium

Translational equilibrium exists when the collision integral in the Boltzmann equation is zero. That is, when the number of particles scattered into every region of phase space is equal to those scattered out of it. This state is relatively easy to define for the lattice gas, for binary collisions. Using an approach similar to Broadwell (Ref. 20) one can show that equilibrium will exist for a single-speed gas if:

$$
f_1 f_4 = f_2 f_5 = f_3 f_6
$$

where these $f$'s are averaged over some region of space. It is important to initialize the flow field to an equilibrium state if there are free-stream boundary conditions applied. Otherwise the interior flow will equilibrate, while the boundary conditions will not, and a secondary flow develops.

It would be very beneficial to incorporate an additional class of particles with a speed equal to twice the current particle speeds. This would allow a much richer interaction amongst the particles and would allow a better modeling of heat transfer and temperature effects. The current single-speed particles and stationary particles cannot really interact. If the particles had non-dimensional
speeds of \( v = 0, 1, \) and \( 2; \) then a collision between two \( v = 1 \) particles could result in a \( v = 0 \) and a \( v = 2 \) particle. In addition, a particle could then gain or lose energy at a solid boundary. With only \( v = 0 \) and \( v = 1 \), there is no way to change the total number of particles in each class and still conserve energy during a collision. Thus one cannot reach equilibrium amongst the different classes of particles; and the energy will not be equally distributed between the classes. In a real gas the energy becomes equally distributed between the translational, rotational, and vibrational states at equilibrium.

The code used here can store the logical bits representing the nodal states in either integer or character variables. The character data representation allows 4 and 8 times more nodes per unit of computer memory on the VAX and Cray, respectively. Using 32 million bytes of Cray memory allows us to use 1.9 or 15.2 million nodes using integer and character packing, respectively. The central processing unit (CPU) time requirements are proportional to the number of nodes used and the number of times steps. The execution times for two different computers and for the two different storage techniques are shown in Table 1.

<table>
<thead>
<tr>
<th>Computer</th>
<th>Integer Storage</th>
<th>Character Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAX 8600</td>
<td>( 52.0 \times 10^{-6} )</td>
<td>( 37.0 \times 10^{-6} )</td>
</tr>
<tr>
<td>Cray XMP-24</td>
<td>( 4.6 \times 10^{-6} )</td>
<td>( 11.6 \times 10^{-6} )</td>
</tr>
</tbody>
</table>

Table 1. COMPUTER TIME REQUIREMENTS

where the numbers shown are CPU seconds per node per time step. The Cray using this code is able to calculate roughly \( 3 \times 10^5 \) collisions per CPU second, but this will be problem dependent. This is significantly faster than the DSMC method, but the DSMC method is modeling much more of the physics.

RESULTS AND COMPARISONS

This section presents numerical results from applying the cellular automaton computer program described above to several problems: an impulsively started flat plate (normal to flow), an impulsively started supersonic piston, and Stokes First Problem. The numerical results are compared with experiments and other theoretical results. The predictions for the flow field about a normal flat plate will be compared with flow visualizations from a towing tank experiment. The impulsive piston example is compared to classical 1-D gas dynamic theory. The predictions for Stokes First Problem are compared with the similarity solution from the Navier-Stokes equations.

Impulsively Started Flat Plate Normal to Flow

The first example is for a 2-D flat plate at 90 degrees angle of attack. The plate is impulsively accelerated from rest to a constant speed. Flow separation occurs at the plate edges and the resulting free shear layers form a symmetrical pair of standing vortices behind the plate. As time progresses, the flow pattern associated with these two standing vortices becomes asymmetric. This process eventually leads to the direct interaction of the two free shear layers and the periodic formation of vortices -- a phenomenon commonly known as Karman vortex shedding.

Predicted velocity vectors are shown in figure 4a for a time corresponding to a plate movement of 1/2 its width (\( Ut/L = 0.5 \)), where \( U \) is the plate velocity, \( t \) is the time, and \( L \) is the plate width. The computational grid used for this example is composed of 1536 cells with each cell containing 625 nodes, which is quite a small statistical sampling. There were 960,000 nodes in the computational domain. The Reynolds number was estimated to be 137.
Also shown (Figure 4b) is a photograph from an experiment conducted in the NASA-Ames Fluid Mechanics Laboratory towing tank for the same normalized time. The plate was moved through water by a traversing mechanism with the camera fixed to the traverse. The flow field was visualized by a suspension of neutrally-buoyant particles illuminated by a sheet of laser light. Short time exposure photographs were taken of the induced particle motion. The particle streaks recorded on the photograph correspond to instantaneous streamline patterns. The Reynolds number was 120.

One parameter that can be used to compare the predictions to the experiments is the location of the saddle point in the wake. The experimental saddle point location is shown as an 'x' on Figure 4a. Figures 5 and 6 show similar data, but for later times (Ut/L = 1.0 and 3.0, respectively). The saddle point location for the numerical results is not plotted explicitly, since it is difficult to determine with the large amount of statistical scatter in the results. However, it is clear that the predicted and experimental saddle points do not agree very well. This is most evident in comparing Figure 6a to Figure 6b.

There could be many reasons for this discrepancy, many of these have been discussed above. Another possible explanation may be due to the lack of compressibility effects in the experimental data, since it was performed in water. The numerical results correspond to a free-stream Mach number of 0.23. Steady-state flow fields usually have negligible compressibility effects at Mach numbers this low, but the effects on an impulsively-started separated flow such as this are not obvious. In order to illustrate the compressibility effect, a contour plot of pressure coefficient for Ut/L = 0.5 is shown in Figure 7. This shows an expansion and compression wave moving away from the plate. The discrepancy between the cellular automaton results and experiment may be due to this effect.

**Impulsively Started Piston**

The second numerical example is for supersonic flow in a duct. The flow is given an initial Mach number of 4.4 (using an isentropic speed of sound, $\gamma RT$) and then the ends of the duct are suddenly closed. This produces a shock wave at one end of the tube and an expansion region at the other. In a reference frame fixed with respect to the freestream, this is the classical impulsively started piston problem. However in this case there are two pistons each traveling in the same direction, which produces both an expansion wave and a compression wave (shock) as shown schematically in Figure 8. The lattice in this case had 1800 nodes in the flow direction and 900 nodes across the duct. The cells had 36 x 36 nodes.

Figure 9 shows density contours ($\rho/\rho_0$) in the duct for three different times (50, 300, and 600 time steps). Figure 9a shows the flow just beginning to compress and expand near the duct ends. The later times (Figures 9b and 9c) show the shock and expansion regions moving towards each other, with the region of free-stream flow becoming smaller. Figures 10, 11, and 12 show contours of the other flow quantities: pressure, Mach number, and temperature, respectively, after 600 time steps.

In this case the piston is moving at greater than the escape velocity. The expansion region should therefore be a vacuum with the temperature, pressure, and density equal to zero. The density and pressure, normalized with respect to the free-stream, do fall to very low values (.05 and .08, respectively), but cannot go to zero due to the rest particles. However, the temperature appears to increase. This effect can be attributed to at least two possible causes. First of all, the sample size in the expansion region is very low; in fact it should be zero. Secondly, the single-speed gas simply does not model temperature very well.

The jump conditions across the shock wave can be compared to 1-D gas dynamic results (for a $\gamma=2$ gas). From the above predictions, one can calculate that the
Figures 4 thru 6. Predicted Velocity Vectors (4a, 5a, and 6a) and Experimental Particle Streaks (4b, 5b, and 6b) Around Flat Plate at 90 Degrees Angle-of-Attack
Figure 7. Predicted Pressure Distribution Around Flat Plate at 90 Degrees Angle of Attack and Re = 137. (U/L = 0.5)

Figure 8. Sketch Illustrating Piston Problem.
Figure 9. Density Contours in Tube for $M_\infty = 4.4$ for Three Different Times (50, 300, and 600 time steps).
Figure 10. Pressure Contours in Tube for $M_\infty = 4.4$ After 600 Time Steps.

Figure 11. Mach Number Contours in Tube for $M_\infty = 4.4$ After 600 Time Steps.

Figure 12. Temperature Contours in Tube for $M_\infty = 4.4$ After 600 Time Steps.
shock is moving at Mach 6.6. The ratios of the various flow quantities across the shock are shown in Table 2.

<table>
<thead>
<tr>
<th>QUANTITY</th>
<th>NUMERICAL</th>
<th>THEORETICAL</th>
<th>NUMERICAL</th>
<th>THEORETICAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_2/P_1$</td>
<td>60.0</td>
<td>57.8</td>
<td>4.38</td>
<td>4.58</td>
</tr>
<tr>
<td>$T_2/T_1$</td>
<td>20.6</td>
<td>20.1</td>
<td>2.28</td>
<td>2.36</td>
</tr>
<tr>
<td>$\rho_2/\rho_1$</td>
<td>2.91</td>
<td>2.87</td>
<td>2.10</td>
<td>1.94</td>
</tr>
</tbody>
</table>

Table 2. RATIOS OF FLOW QUANTITIES ACROSS MACH = 6.6 AND 1.9 SHOCK WAVES

The Mach = 1.9 results are very surprising since they were obtained using only 20 cells (100 nodes) in the axial direction.

As mentioned above, the cells are quite arbitrary and are only used to calculate flow quantities. The above fine grid solution was restarted (for zero time steps) in order to average over different size cells. In this case the cells had 2 x 900 nodes, i.e. the cells were two nodes wide and span the entire duct. This allows a more detailed examination of the shock. From the previous figures one can see that the boundary layers are quite thin, so this should not contaminate the averaging too much. Figure 13 shows temperature as a function of distance along duct using this averaging scheme. The temperature gradient through the shock wave is quite large. The large statistical scatter in the expansion region (due to the small sample size) is also apparent in this figure. Figure 14 shows temperature and velocity versus "distance along duct/mean free path" for the shock wave, in order to show the shock structure. The shock thickness, calculated by dividing the velocity jump by the maximum velocity slope is approximately 5 mean free paths.

Stokes First Problem

The third and final example is the classical Stokes First Problem (sometimes called the Rayleigh problem) where an infinite flat plate (or plane wall) is impulsively accelerated in its own plane to a constant speed. The flow (boundary layer) develops in time. In the computation, only 10 cells normal to the plate were used (100 nodes) with 500 nodes along the plate. The velocity profile for two different times is shown in figure 15 in terms of a similarity parameter $\eta$. The distance from the wall is normalized with respect to time and viscosity in the following form:

$$\eta = \frac{y}{2 \sqrt{v t}}$$

where $y$ is the normal distance from the wall, $v$ is the kinematic viscosity, and $t$ is the time from the start of motion. The theoretical velocity profile depends only on $\eta$, $u = u(\eta)$. The square and round symbols in Figure 15 are for 200 and 400 time steps, respectively. There are two symbols shown for each normalized distance from the wall because the flow was simulated on both sides of the plate. This gives an indication of the statistical scatter in the results. Also shown are theoretical results for the Navier-Stokes equations (Ref. 22). The two agree quite well, however there is some scatter in the numerical results. The scatter could probably be reduced by taking ensemble averages over several runs as is typically done in Monte Carlo methods or by using more nodes per cell. These results give us some confidence that the simple formula used for viscosity is a reasonable approximation.
This paper applied the gas dynamic model called cellular automata to the prediction of three different flow fields. In the past the method has been compared to very few experiments or analytical solutions. Although many of the cellular automaton results shown here compare well to other theories and experiments, the method does have some rather restrictive limitations. It may be misleading to compare the relative efficiency of the method to numerical techniques based upon the more complete Navier-Stokes or Boltzmann equations since it does not represent the physics as well as they do. It also cannot simulate many flow phenomena such as heat transfer effects or the effect of temperature on the transport properties. It should also be remembered that a 2-D Direct Simulation Monte Carlo (DSMC) method with hard sphere molecules, no chemical reactions, and a uniform grid could also be programmed to run very efficiently on a parallel processor, as could an explicit numerical method based upon the continuum equations. In order to include the physics in the cellular automaton approach, substantial reductions in its efficiency may be required. In addition, the method is not well suited to the computers used here or, for that matter, to Fortran because it is based upon bit operations instead of floating point numbers.

And finally, this method may be particularly useful in teaching some aspects of the kinetic theory of gases. As Broadwell notes (Ref. 20): "A method which leaves the physics of the collisions in such plain view should be helpful in developing intuitive ideas about flows for which the Navier-Stokes approximation may be inadequate." Indeed this may also be true for flows in which the Navier-Stokes equations are valid.

REFERENCES

Figure 13. Temperature Distribution Along Centerline of Tube for $M_{\infty} = 4.4$.

Figure 14. Temperature Profile Through Shock Wave $M_{\text{shock}} = 6.6$ After 600 Time Steps.

Figure 15. Predicted Velocity Profile for Stokes First Problem Compared to Theoretical Values.


