

Calculation of the Permeability of Porous Media Using Hydrodynamic Cellular Automata

G. A. Kohring¹

Received October 28, 1991

The permeability of two-dimensional porous media is calculated numerically as a function of porosity using the hydrodynamic cellular automata (lattice gas) approach. Results are presented for systems with up to 22 million sites (8192×2688). For randomly distributed solid obstacles whose macroscopic dimensions are much longer than the mean free path of particles in the fluid, the permeability κ varies with porosity ε as $\kappa \propto (\varepsilon - 0.6)/(1 - \varepsilon)$ for $\varepsilon > 0.7$. When the solid obstacles are much smaller than the mean free path of particles in the fluid, i.e., when they form a dust of point objects, then such a relationship no longer holds and the permeability is more than an order of magnitude smaller than for the former case. The program used for the simulations is discussed and a listing is presented in the Appendix which achieved a sustained speed of 185 million sites updated per second on a single processor of the Cray-YMP. (On a Sun Sparc Workstation, the same program ran about 100 times slower.)

KEY WORDS: Lattice gas; hydrodynamics; cellular automata; porous media.

The last several years have seen convincing evidence that a certain class of cellular automata models originally introduced by Frisch, Hasslacher, and Pomeau (FHP)⁽¹⁾ can be used for modeling hydrodynamic fluid flow. By now the strengths and weaknesses of these models are well known and have been discussed in detail by many researchers.⁽²⁾ One area in which these models should be particularly useful is that of fluid flow involving complicated geometries at low Reynolds number.⁽³⁾ In fact, Brosa has recently shown that for flows involving porous membranes, the cellular automata approach actually outperforms the traditional finite-element methods.⁽⁴⁾ In this paper a related problem is studied for which there seems to be little progress via traditional methods, namely flow in porous media.

¹ Institut für Theoretische Physik, Universität zu Köln, D-5000 Köln 41, Germany.

Characterizing such flows is important for geologists seeking to understand the evolution of sandstone formations as the water drains off, as well as for engineers attempting to estimate the oil and gas recovery potential of a particular reservoir.^(5,6)

This problem has been studied qualitatively by previous researchers using small to modest size lattices.^(5,6) In this paper, an attempt is made at a more quantitative study with larger lattices. Additionally, a six-bit model,⁽²⁾ i.e., a model without rest particles, is used and shown to yield results in agreement with the seven-bit models,⁽²⁾ i.e., models using rest particles. The use of the six-bit model enables one to develop a more efficient computer algorithm which is six times faster and able to handle lattices eight times larger than the Brosa–Stauffer algorithm.⁽⁷⁾ The Fortran program used for these studies is given in the Appendix. On a single processor of the Cray-YMP/832 it achieved a sustained speed of 185 million site updates per second (185 Mups) with the porous media in place. When running in the multiprocessor batch mode (autotasking), the program reached a peak speed of about 800 Mups, depending upon the batch environment at the time the job is running. (We did not make any studies of performance on a dedicated machine.) This program does not, however, require a supercomputer. On a Sun Workstation it runs at about 1.3 Mups.

The collision rules used here are those of the original FHP model,⁽¹⁾ supplemented with a set of four-body collisions. Additionally, following Brosa and Stauffer,⁽⁷⁾ there is an angular momentum bit to ensure local conservation of angular momentum. These collision rules are designed to avoid spurious conservation laws.

For studying flow properties in porous media, the porous material is placed in the center half of a pipe. The length of the pipe L in these studies is always about three times longer than the width. The porous material is generated by randomly placing on the lattice a predetermined number of obstacles whose characteristic size is 2% of the lattice length. For convenience, the obstacles used in these studies were rhombi. (Duarte and Brosa⁽⁸⁾ have previously shown for flow around a single object that the precise shape of the object does not significantly affect the results.) In placing the rhombi on the lattice, they are allowed to overlap, creating highly irregular structures. After all the rhombi have been placed on the lattice, the porosity of the medium is calculated as the ratio of the unoccupied sites to the total number of sites in the center half of the pipe. For random structures this should give a good definition of porosity; however, a more sophisticated definition is needed in the case of highly structured material.

For flow at low Reynolds number, the permeability can be defined as

the ratio of the fluid flux through the medium and the pressure difference across the medium. The fluid flux in these simulations was measured over the last one-eighth of the lattice, while the pressure difference across the media can be calculated very simply by measuring the pressure on the walls in front of and behind the medium and taking the difference.

In these simulations the fluid is kept flowing by replacing the first column of the lattice with a new Poiseuille configuration at each time step. The new configuration was chosen so that the maximum speed v_m in the center of the channel was approximately $v_m \approx 0.15$. For all the results presented here the density of particles per link, d , is approximately $d \approx 0.15$, which corresponds to a density per site of 0.9.

Figure 1 shows the results for lattices of various size: 2048×640 , 4096×1344 , and 8192×2344 . For each configuration of obstacles, the system was allowed to equilibrate and then the above described measurements were averaged over the next 20,000 time steps. For clarity, only an estimate of the largest statistical error is shown for the 2048×640 system at a porosity of $\epsilon = 0.93$. From this figure, it can be seen that the permeability varies with porosity as $\kappa \propto (\epsilon - \epsilon_0)/(1 - \epsilon)$ for $\epsilon \geq 0.7$. From these data ϵ_0 can be estimated as $\epsilon_0 \approx 0.6$. These results are in good agreement with the data of Brosa and Stauffer, even though they used circular objects and a seven-bit model.⁽⁷⁾

Now, it might be questioned whether it is possible to use point objects instead of solid objects since such a “dust” is much easier to generate on

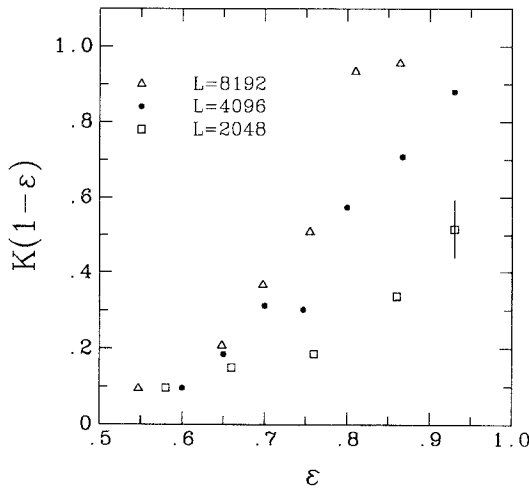


Fig. 1. Plot of the reduced permeability $\kappa(1 - \epsilon)$ versus the porosity ϵ at different lattice dimensions $L \times (L/3)$; the size of the obstacles is taken to be $L/50$. In all cases, the particle density is 0.15 particles per link of the lattice, i.e., 0.9 particles per site.

```

SUBROUTINE UPDATE(X1,X2,X3,X4,X5,X6,Y1,Y2,Y3,Y4,Y5,Y6,
&                ANG,AANG,OB,NOTOB)
  IMPLICIT NONE
  INTEGER B,L,V,H,LOB,VM1OB,BM1,HM1
  PARAMETER (B=64,L=8192,H=2344,V=L*H,LOB=L/B,
&           VM1OB=(V-1)/B,BM1=B-1)
  INTEGER X1(0:VM1OB),X2(0:VM1OB),X3(0:VM1OB),
&        X4(0:VM1OB),X5(0:VM1OB),X6(0:VM1OB),
&        Y1(0:VM1OB),Y2(0:VM1OB),Y3(0:VM1OB),
&        Y4(0:VM1OB),Y5(0:VM1OB),Y6(0:VM1OB),
&        ANG(0:VM1OB),AANG(0:VM1OB),COL,NCOL,
&        OB(0:VM1OB),NOTOB(0:VM1OB),RULE,A,C,D,E,W
  RULE(A,C,D,E)=( E.AND.OB(W) )+ ( ( A.AND.COL.AND.AANG(W) )+
&   ( C.AND.COL.AND.ANG(W) )+(D.AND.NOCL) )
C
C***** This following loop looks for the
C collisions and sets the bits for the outgoing particles.
C
  DO 10 W=0,VM1OB
    NCOL=( ( X1(W).XOR.X4(W) ).OR.(X2(W).XOR.X5(W) ).OR.
&          ( X3(W).XOR.X6(W) )
&          .AND.( X1(W).XOR.X3(W) ).OR.(X3(W).XOR.X5(W) ).OR.
&          ( X2(W).XOR.X4(W) ).OR.(X4(W).XOR.X6(W) )
&          ).AND.NOTOB(W)
    COL=(.NOT.NCOL).AND.NOTOB(W)
    Y1(W)=RULE(X2(W),X6(W),X1(W),X4(W))
    Y2(W)=RULE(X3(W),X1(W),X2(W),X5(W))
    Y3(W)=RULE(X4(W),X2(W),X3(W),X6(W))
    Y4(W)=RULE(X5(W),X3(W),X4(W),X1(W))
    Y5(W)=RULE(X6(W),X4(W),X5(W),X2(W))
    Y6(W)=RULE(X1(W),X5(W),X6(W),X3(W))
    ANG(W)=(COL.AND.AANG(W)).OR.(NCOL.AND.ANG(W))
  10  AANG(W)=.NOT.ANG(W)
C
C***** The next loops propagates the particles
C on the interior sites (the pipe is assumed to have walls at
C the top and bottom of the lattice)
C
  DO 20 W=LOB,VM1OB-LOB
    X1(W)=Y1(W+LOB-1)
    X2(W)=Y2(W-1)
    X3(W)=Y3(W-LOB)
    X4(W)=Y4(W-LOB+1)
    X5(W)=Y5(W+1)
  20  X6(W)=Y6(W+LOB)
  DO 30 W=0,LOB-1
    X1(W)=Y1(W+LOB-1)
    X6(W)=Y6(W+LOB)
  30
  DO 40 W=VM1OB-LOB+1,VM1OB
    X3(W)=Y3(W-LOB)
  40  X4(W)=Y4(W-LOB+1)
C
C***** Then corrections are made for the left
C and right edges of the lattice (periodic B.C. are assumed).
C
  DO 50 W=0,VM1OB-LOB,LOB
    X1(W)=SHIFT(Y1(W+2*LOB-1),1)
    X2(W)=SHIFT(Y2(W+LOB-1),1)
  50
  DO 60 W=2*LOB-1,VM1OB,LOB
    X4(W)=SHIFT(Y4(W-2*LOB+1),BM1)
    X5(W)=SHIFT(Y5(W-LOB+1),BM1)
  60
  RETURN
  END

```

Fig. 2. Fortran subroutine for the updating of the lattice.

a computer.⁽⁹⁾ Simulations with such dust models produced permeabilities more than one order of magnitude smaller than those in Fig. 1, even when the porosity was as large as $\varepsilon \approx 0.98$. Hence, such dust models are to be avoided and obstacle sizes much larger than the mean free path of a particle in the fluid should be used.

In conclusion, hydrodynamic cellular automata on a relatively large scale, 22 million sites, have been easily and successfully applied to the problem of flow in porous media. The results presented here look reasonable and it would be interesting if they could be verified in the real-world laboratory.⁽¹⁴⁾ For further study, a detailed calculation of the object size dependence would be of interest as well as a study of how these results depend upon the object distribution, i.e., whether the distribution is fractal or not.

APPENDIX

The complete Fortran subroutine which handles the updating of the lattice is shown in Fig. 2. The triangular lattice is oriented such that the bases of the equilateral triangles lie parallel to the x axis. The lattice directions are numbered 1–6 clockwise starting from the upper right-hand corner,⁽⁷⁾ while the lattice sites are numbered sequentially starting with 0 in the upper left-hand corner of the lattice and running from left to right. Each new row of the lattice is shifted one lattice site to the right of the previous row, thus forming a large parallelogram. The lattice has L sites in the x direction and H sites in the y direction.

Particles incoming to a site from one of the six directions are stored in the one-dimensional arrays $X1$ – $X6$, and particles outgoing from a site are stored in the arrays $Y1$ – $Y6$. The angular momentum at each site is stored in ANG and in its negation $AANG$. Whether a given site is or is not part of an obstacle is stored in OB and $NOTOB$, respectively. If the computer has B bits, then there are B lattice sites stored in each word of the arrays. For example, if, say, the first bit of the first word of $X1$ is set to one, then there is a particle at site number 0, traveling *into* lattice direction number 1. Similarly, if the first bit of the first word of $Y1$ is set to one, then there is a particle at site 0, traveling outward in direction number 1.

The DO-LOOP ending on statement number 10 examines the X vectors for possible collisions at a given site. Collisions occur whenever the incoming state consists of two, three, four, or six particles arranged symmetrically about the site. If a collision is detected, then the appropriate bits designating outgoing particles in the Y vectors are set. Otherwise, the particles are simply allowed to propagate onward. The rule for setting the

outgoing bits in the presence of a collision is very simple: if the angular momentum bit ANG at a site is set to zero, i.e., the corresponding bit in $AANG$ is set to one, then the outgoing state is obtained from the incoming state by $X1 \rightarrow Y6$, $X2 \rightarrow Y1$, $X3 \rightarrow Y2$, $X4 \rightarrow Y3$, $X5 \rightarrow Y4$, $X6 \rightarrow Y5$. If the angular momentum bit is set to one, then in the presence of a collision, the outgoing state is obtained from the incoming state by $X1 \rightarrow X2$, $X2 \rightarrow Y3$, $X3 \rightarrow Y4$, $X4 \rightarrow Y5$, $X5 \rightarrow Y6$, $X6 \rightarrow Y1$. After a collision has taken place, the angular momentum bit is flipped to its opposing value. Finally, should it happen that the site is an obstacle of some sort, e.g., the top or bottom of the pipe, then all of the incoming particles are simply reflected back along the direction they came from. This corresponds to no-slip boundary conditions.

Note that the collision rule, defined by the statement function $RULE$, has a combination of logical and arithmetic operations. This is possible, since $I+J$ and $I.OR.J$ are interchangeable whenever I and J do not simultaneously have corresponding bits set to one. Now, by construction this is true for the present collision rule. On a computer like the Cray-YMP with separate logical and arithmetic units, this enables several operations to be performed simultaneously and yields nearly a 20% speedup over the standard method of not mixing logical and arithmetic operations in a single statement.

After handling the collisions, the program then propagates the outgoing particles to the neighboring sites. Now, the sites, although numbered consecutively throughout the lattice, are not stored consecutively within each word; rather, they are stored *across* the words as is commonly done in statistical mechanics. For example, site zero is stored in the first bit of word one, site one is stored in the first bit of word two, site three in the first bit of word three, etc., until the first bit of each of the first $LOB = L/B$ words are used up. The next site then goes into the second bit of the first word, etc. This is repeated until the first LOB words are full. These LOB words then contain all the sites of the first row of the lattice. The next LOB words will contain the sites of the second row and so on.

With this method of storing the sites, loops 20, 30, and 40, which propagate particles on the interior sites of the lattice, are simple assignment statements. If one were to have stored the sites consecutively within each word, then the propagation statements would have contained many shift and logical operations. The present storage scheme requires shift operations only at the left and right boundaries, loops 50 and 60.

The program presented here bears some resemblances to previous program presented by Hayot *et al.*⁽¹⁰⁾ Riccardi *et al.*,⁽¹¹⁾ and Gunstensen,⁽¹²⁾ but there are important differences, such as the local conservation of angular momentum,⁽⁷⁾ using one-dimensional arrays, storing the lattice

more efficiently, and taking advantage of occasional equivalences between arithmetic and logical operations which all add up to a large gain in computer speed. As mentioned previously, this program runs at 185 Mups on a single processor of the Cray-YMP, while the program of Riccardi *et al.* runs at 38 Mups on a single processor of the ETA10. In the multi-processor batch mode, i.e., autotasking, the program runs at about 800 Mups, depending upon the batch usage at the time the program is running. This compares quite favorably with a speed of 1000 Mups obtained on a full, dedicated connection machine. To transport the above program to a Sun Sparc Workstation, the number of bits B must be set to $B=32$ and the function *SHIFT* must be changed to *LSHIFT*. The program then runs at 1.3 Mups, which compares well with the 32-processor INTEL hypercube, 2.3 Mups, and the special-purpose machines CAM-6, 4 Mups, and ENS, 6.5 Mups (see ref. 10 and references therein). Even the new RAP machine reaching completion at ENS is still somewhat slower (although much cheaper) than one Cray-YMP processor.⁽¹³⁾

ACKNOWLEDGMENTS

It is a pleasure to thank D. Stauffer and E. Flekkoy for many useful discussions. A grant from the BMFT (#0326657D) for partial support of this project is gratefully acknowledged as well as a grant of computer time on HLRZ's Cray-YMP at the KFA Jülich.

REFERENCES

1. U. Frisch, B. Hasslacher, and Y. Pomeau, *Phys. Rev. Lett.* **56**:1505 (1986).
2. M. Hénon, *Complex Systems* 1:763 (1987); J.-P. Renet, M. Hénon, U. Frisch, and D. d'Humières, *Europhys. Lett.* **7**:231 (1988); H. A. Lim, *Phys. Rev. A* **40**:968 (1989); P. Kadanoff, G. R. McNamara, and G. Zanetti, *Phys. Rev. A* **40**: 4527 (1989).
3. G. D. Doolen, First Topical Conference "Computational Physics" of the American Physical Society, Boston, June 1989.
4. U. Brosa, *J. Phys. France* **51**:1051 (1990); U. Brosa, C. Küttner, and U. Werner, *J. Stat. Phys.* **60**:875 (1990).
5. D. H. Rothman, *Geophysics* **53**:509 (1988); S. Chen, K. Diemer, G. D. Doolen, K. Eggert, C. Fu, S. Gutman, and B. Travis, in Proceedings of the NATO Advanced Workshop on Lattice Gas Methods for PDE's, *Physica D*, to be published; S. Suci, A. Cancelliere, C. Chang, E. Foti, M. Gramignani, and D. Rothman, in *Computational Methods in Subsurface Hydrology*, G. Gambolati, A. Rinaldo, C. A. Brebbia, W. G. Gray, and G. F. Pinder, eds. (Springer-Verlag, Berlin, 1990).
6. U. Bosa and D. Stauffer, *J. Stat. Phys.* **63**:405 (1991).
7. U. Brosa and D. Stauffer, *J. Stat. Phys.* **57**:399 (1989).
8. J. A. M. S. Duarte and U. Brosa, *J. Stat. Phys.* **59**:501 (1990).

9. J. P. Boon, in *Correlations and Connectivity Geometry Aspects of Physics, Chemistry and Biology*, H. E. Stanley and N. Ostrowsky, eds. (Academic Press, Dordrecht, 1990).
10. F. Hayot, M. Mandal, and P. Sadayappan, *J. Comp. Phys.* **80**:277 (1989).
11. G. Riccardi, C. Bauer, and H. Lim, in Proceedings of the NATO Advanced Workshop on Lattice Gas Methods for PDE's, *Physica D*, to be published.
12. A. Gunstensen, MIT unpublished report.
13. D. d'Humières, Private communication.
14. J. Feder and T. Jossang, Private communication.