J. Fluid Mech. (1982), *vol.* 119, *pp.* 249–267
Printed in Great Britain

Capillary displacement and percolation in porous media

By RICHARD CHANDLER, JOEL KOPLIK, KENNETH LERMAN AND JORGE F. WILLEMSEN
Schlumberger-Doll Research, P.O. Box 307, Ridgefield, Connecticut 06877, U.S.A

(Received 10 August 1981)

We consider capillary displacement of immiscible fluids in porous media in the limit of vanishing flow rate. The motion is represented as a stepwise Monte Carlo process on a finite two-dimensional random lattice, where at each step the fluid interface moves through the lattice link where the displacing force is largest. The displacement process exhibits considerable fingering and trapping of displaced phase at all length scales, leading to high residual retention of the displaced phase. Many features of our results are well described by percolation-theory concepts. In particular, we find a residual volume fraction of displaced phase which depends strongly on the sample size, but weakly or not at all on the co-ordination number and microscopic-size distribution of the lattice elements.

1. Introduction

Fluid transport through porous structures is a widely studied topic with a number of applications. Within the petroleum industry, in particular, displacement processes involving immiscible fluids have received much recent attention at the microstructure scale, i.e. the scale of grains and pores. We have investigated such processes and report here on the effect of void-space topology and geometry on so-called capillary displacement processes – flow induced by capillary forces at extremely low, controlled flow rates.

Natural porous structures consist of a completely connected solid matrix and, with regard to fluid flow, a completely connected void space. This void space is often modelled as a hydraulic network consisting of interconnected channels of variable cross-section. We consider the simplest relevant model, a regular planar network of given co-ordination number, with pore-channel cross-sections assigned randomly from a given distribution, and finite volume associated with the channel intersections, or pore sites. Fluid displacement is modelled by computer simulation, using a Monte Carlo procedure, wherein we hope to obtain information of qualitative value without delving into the complicated details of interface motion in an irregular geometry. We find many aspects of such displacement processes to be governed by percolation theory, including an important scaling law that predicts the residual fractional volumes of the fluids.

Network modelling of the pore space is an active research area with a correspondingly large literature. The crudest model is that of a bundle of non-intersecting tubes, each of constant but different cross-section corresponding to some given pore-size distribution. This model is the basis of so-called 'hydraulic radius' theories used

to describe single-phase transport in porous media (Scheidegger 1974). More elaborate models in which each tube radius is permitted axial variation have been applied to capillary displacement processes (Dullien 1975). Such models fail to predict the salient features of capillary displacement – the known hysteresis between imbibition and drainage, and the retention of displaced fluid. These features require modelling by interconnected networks.

Studies of displacement processes in interconnected networks are often implemented through computer simulation. Chatzis & Dullien (1977) have investigated linear displacement in various two-dimensional networks where one fluid is assumed to be completely compressible. Mohanty, Davis & Scriven (1980) simulated low-velocity displacements of oil by water on square networks, including a pinch-off process that can encourage the disconnection of the non-wetting fluid. Androutsopoulos & Mann (1979) simulated a mercury porosimetry experiment on a square network. We find that the results of such simulations are strongly affected by the finite size of the network, and that a careful application of percolation theory accounts for this behaviour.

In §2, we review the physics of capillary displacement. We describe the details of our simple capillary displacement simulation in §3. In §4 we describe the steady-state displacement configuration and effects thereon of pore-space topology and geometry. Percolation theory is discussed in §5 and applied to the problem of capillary displacement. A summary of results appears in §6. After this paper was completed, we learned of related work by de Gennes & Guyon (1978) and Lenormand (1980). The latter paper simulates capillary displacement using essentially the same model considered here, but analyses the results somewhat differently.

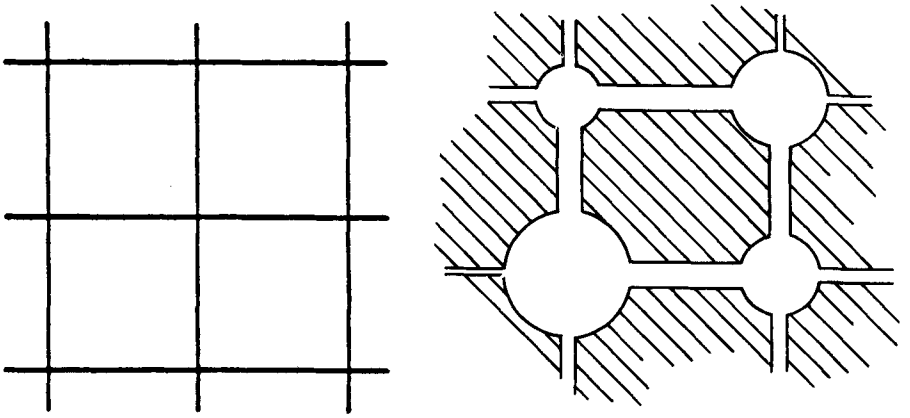
2. Physics of capillary displacement

Our intention is to gain insight into two-phase displacement processes in fluid-porous rock, under conditions where capillary forces dominate. The standard dimensionless group for this process is the capillary number

$$C = \mu v/\gamma, \tag{1}$$

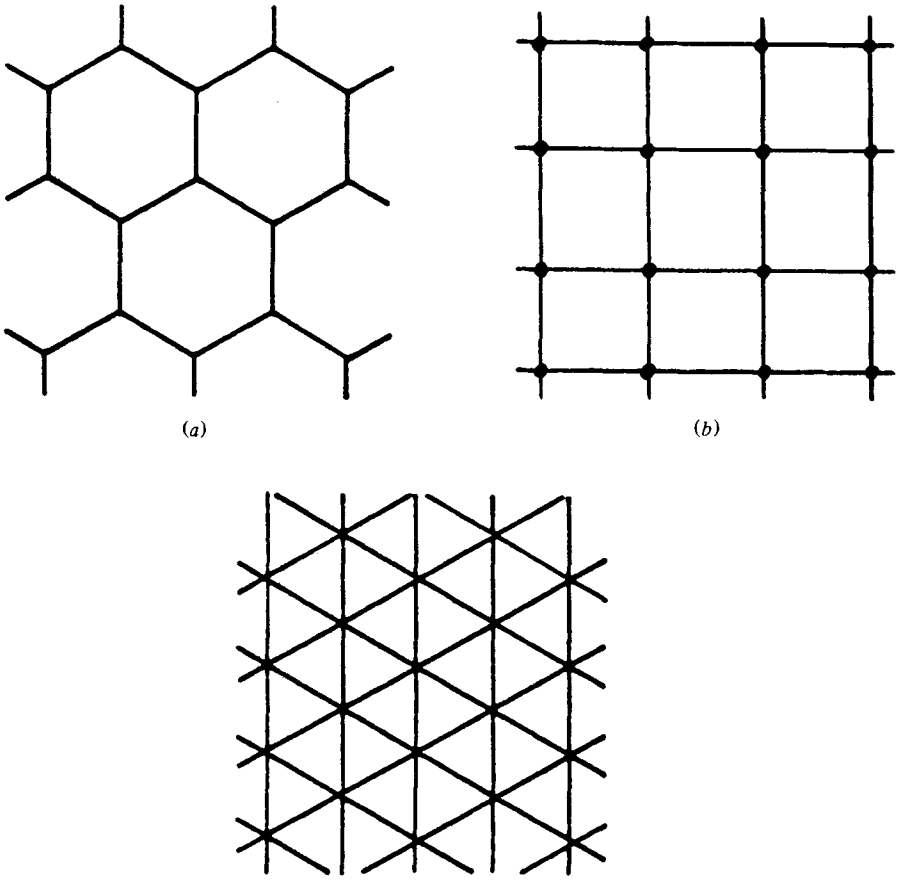
where μ, v and γ are respectively the viscosity, average velocity and surface-tension coefficient. Roughly speaking, C is the ratio of viscous to surface-tension forces, and capillary displacement is characterized by $C \ll 1$. In this regime the flow rate is so low that the motion is quasi-static, consisting of intermittent motion in only one element of the network at a time, which we may think of as the limiting case $C \to 0$. We will be more explicit about the restrictions imposed by this limit at the end of this section.

We begin by representing the porous medium as a lattice of fixed co-ordination number (the number of links meeting at a node), whose elements have variable sizes following some probability distribution. For example, in figure 1, we show a schematic two-dimensional square lattice, together with a possible realization as a network of circular pores (located on the sites or nodes) and straight throats (on the bonds or links). In this paper, we study two-dimensional lattices whose co-ordination number σ is 3, 4 or 6, as indicated in figure 2. As discussed in §3, there is little variation



(*a*) (*b*)

FIGURE 1. A two-dimensional square network: (*a*) schematically and (*b*) realistically.



(*a*) (*b*)

(*c*)

FIGURE 2. Lattices studied in this work.

of the results with σ, and there seems to be no need to study more values in two dimensions.

The process of displacement of an incompressible fluid from the network by a second immiscible and incompressible fluid is hereinafter referred to as water displacing oil. Consider a lattice initially filled with oil, and allow water to enter along one edge in the presence of capillary pressure differences across the interface. If the rock is preferentially wetted by water, then water will spontaneously be imbibed while, if the rock is preferentially wet by oil, we imagine applying an external pressure head across the lattice and then capillary pressure will provide a variable resisting force. In either case, at very small flood rates, there will be little pressure variation in either fluid phase due to viscous effects. However, the capillary pressure difference across the oil–water interface

$$p_c = \frac{\gamma}{R}, \tag{2}$$

where γ is the coefficient of surface tension and R the mean radius of curvature, can show considerable variation because R is proportional to the local cross-sectional radius of the pore space, which may vary strongly from point to point. The individual menisci in any pore or throat are then subjected to forces whose magnitude may vary dramatically along the interface, and the interface will move in a very irregular fashion with strong local variation in velocity and in almost-discrete steps corresponding to motion through individual lattice elements. We are then led to *simulate* the interface motion as a discrete, stepwise process, where in each step the oil–water interface moves through exactly one element of the lattice, where the chosen element has a lowest *ranking* along the interface.

The connection between the rankings and the geometry of the pore space is as follows. We assign a ranking to the lattice throats (links) alone, reasoning that the pores (nodes) are much larger and have much smaller values of p_c than the throats, and the interesting variation in p_c then occurs in the throats. For imbibition, the ranking would increase with size, so that as in (2) the smallest rank would correspond to the largest force, while in drainage the ranking would decrease with size because here motion is more likely when there is less capillary (resisting) pressure. The *simulation* is indifferent to which fluid wets the matrix, and for convenience, we shall continue to refer to the displacing fluid as water and the displaced fluid as oil. Most of the pore-space volume is in the pores, and so we count the amount of oil present as those *nodes* occupied by oil. We also make the simplifying assumption that all pores have equal volume, although we could easily incorporate a variable pore volume. Following conventional petroleum terminology, we shall refer to the volume fraction of displaced fluid phase remaining after a displacement process as *residual saturation*.

The simulation process is carried out by computer as follows. Starting from an initial interface at one end of the lattice, ranks are assigned to all throats on the interface. The lowest-rank throat is selected, water displaces oil in the appropriate pore, and throats are added or removed as required from the interface list. Ranks are assigned to the new throats, the lowest is again selected, and the process repeated. A possible sequence of interface positions is shown in figure 3. Residual or trapped oil occurs when a region of oil-filled pores is surrounded by water (figure 4), because the fluids are incompressible and motion of an oil blob requires at least two simultaneous

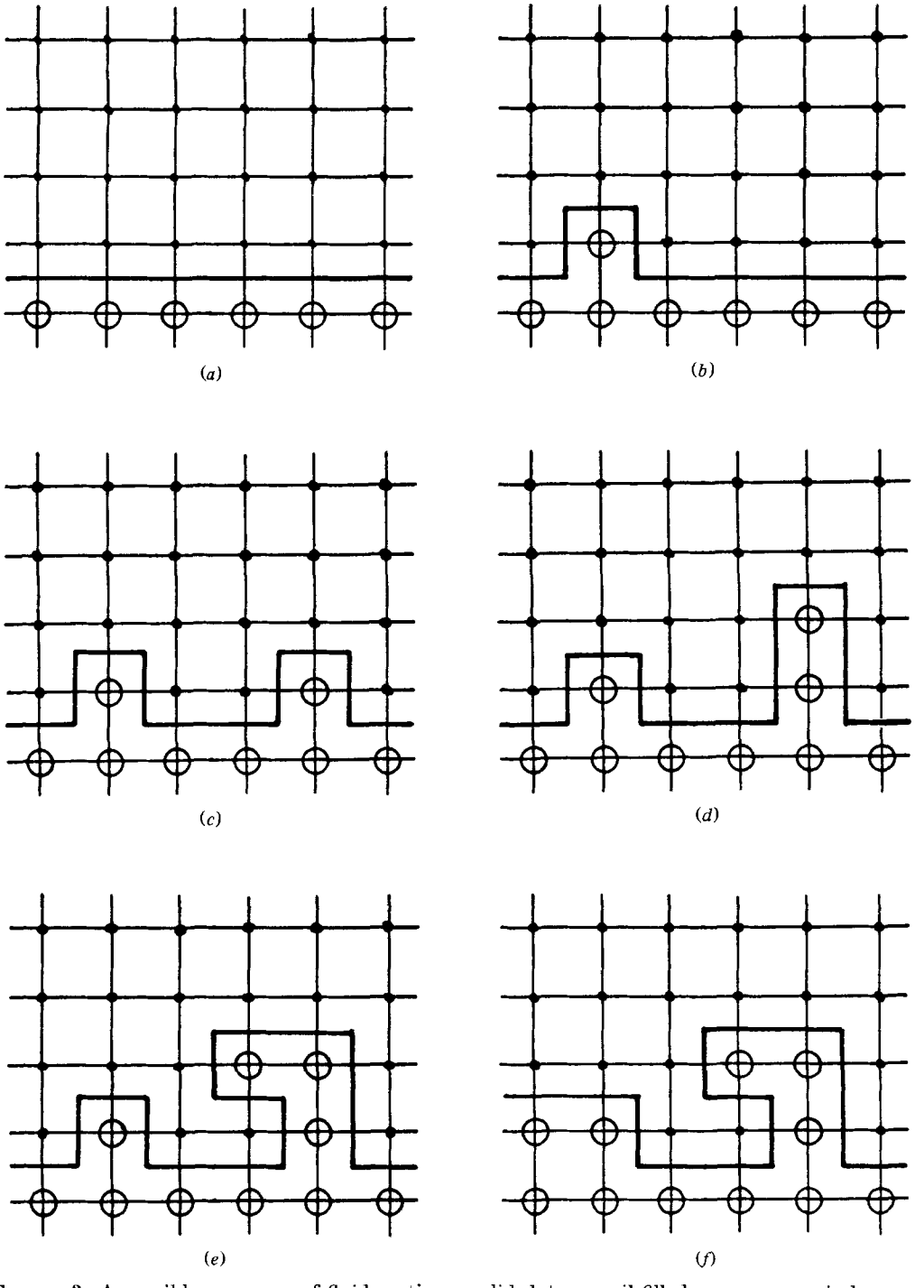


FIGURE 3. A possible sequence of fluid motions: solid dots are oil-filled pores, open circles are water-filled pores, light lines are throats, and the heavy line is the oil–water interface.

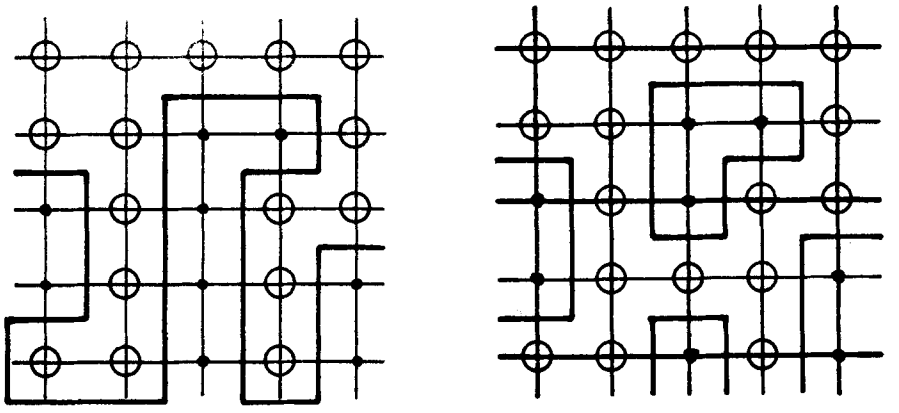


FIGURE 4. An example of oil trapping, before and after.

steps – oil filling one pore and draining from another.† After each step in the simulation, the interface is checked for trapped regions; these are stored and the relevant throats are removed from the interface list. Further explanation of the programming is given in §3.

In practice, one chooses a lattice of some σ, length and width (in number of pores) and lets water invade one end of the lattice while oil is driven out of the opposite end. On the remaining sides of the lattice, some boundary condition is imposed: permeable, impermeable or, usually, periodic. The interface works its way from end to end, trapping oil regions as it goes, and the process stops when only residual oil remains. Note that since the motion always occurs through the one throat of highest ranking, only the *sequence* of numbers is significant. The actual shape of the pore-space size distribution is irrelevant, and it suffices to generate randomly a uniform probability distribution of ranks.

The actual size distribution of the void space, while irrelevant for the motion, does enter into the computation of the *amount* of trapped oil. To the extent that the pores are much larger than the throats, the residual saturation of oil for any lattice will be simply proportional to the number of oil-filled pores, and independent of the size distribution. In this manner, we assume essentially a factorization of residual saturation into a topological term and a volume term, and focus our attention on the topological term.

The applicability of our results is limited by our strong initial assumptions. Since we allow fluid motion in only one throat at a time, we require that the lattice cannot be too large, and that the probability distribution of throat sizes is not too narrow. Ignorance of the details of interface motion preclude a precise estimate of a largest reasonable size, but it is clear that the latter increases with C because the number of pore displacements per time interval should increase with flow rate. A further reason to apply our results only to very small C is the presence of large regions of trapped oil (see §4). Following the work of Melrose & Brandner (1974) and others, we expect

† Realistically speaking, motions involving two pores at once are bound to occur since the total capillary force on the boundary of a trapped region is unlikely to be exactly zero. However, the resultant force will tend to be small and randomly oriented, and we neglect the consequent slow motion of oil blobs.

large oil blobs to be mobilized once the viscous pressure difference in the flow direction exceeds the net capillary pressure, and for consistency we must assume C so small that the largest trapped oil blobs remain immobile.

3. Programming of the simulation

We refer to the computer program that models capillary displacement under the approximations described above as the *simulation*, and the porous medium in question as the *bed*. The bed consists of a finite regular array of *pores*, each of which is connected to its *nearest neighbours* by *throats*, and the number of nearest neighbours of each pore is the *co-ordination number*. Certain of the pores at one edge of the bed are connected to a *source* that supplies water, while those at the opposite edge are connected to a *sink* that absorbs oil, and the pores at the sides are connected in such a way as to satisfy some boundary condition. In the usual case, the edges are connected to each other to form a cylindrical surface (periodic boundary conditions), but some results have been obtained with the sides connected to the sink (corresponding to a permeable boundary condition) or unconnected (corresponding to an impermeable boundary condition).

At any point in time during the simulation a set of pores containing oil will have one or more neighbours that contain water. If there is a path entirely in oil from such a pore to the sink, it is said to be *on the interface*. Throats that have at one side a pore on the interface and at the other side water are also said to be on the interface. Pores containing oil for which there is no such path to the interface are said to be *trapped*, or to contain *residual oil*.

The simulation process proceeds as follows.

(i) Create a regular bed of specified length, width, co-ordination number and boundary conditions. Assign to each throat a random ranking in the interval [0, 1]. Initialize the list of pores on the interface to consist of those pores connected to the source.

Although this step is conceptually simple, some effort has gone into writing the code so at to maximize the versatility of the simulation. Actually we have a family of simulations which use (mostly) common code and different control tables. The simulation can handle beds of two, three or higher dimensions. In this paper we present two-dimensional results only, deferring other cases to a future publication. The simulation can be run with any co-ordination number, with boundary conditions that are either 'open', 'closed' or 'periodic', with arbitrary connections between source and sink, and (within processor space and time limitations) any size of bed.

(ii) Choose one of the pores on the interface to be filled with water according to the criterion that water will flow through the lowest-ranking throat on the interface. This choice must be made at each step, hence on the order of N^2 times, where N is the number of pores on a side of a square bed. A brute-force approach involves searching the entire list of pores on the interface (which is of length at least order N) each time, and this approach therefore takes time proportional to at least N^3. More-sophisticated approaches involve sorting the pores by throat size as they are added to the interface, and these techniques can reduce the time per step to order log N.

(iii) Add all of the nearest neighbours of the chosen pore which contain oil and are not already on the interface to the list of pores on the interface. Assign a throat

ranking at random to each new throat on the interface. If the pore to be added is already on the interface list, and the new throat is of lower rank than the previous lowest-ranking throat connecting this pore to the interface, remove the previous throat from the (sorted) interface list. Place the pore on the interface list according to the size of its lowest-ranking throat on the interface.

The choice of the precise list structure is similar to the problems faced in the choice of structures for event lists in discrete time simulations (for a discussion of this problem see Knuth 1973; Vaucher & Duval 1975). In this case, ease of programming and relative timing led to the choice of an array of linear lists for this structure.

(iv) Now determine if any of the pores on the list of pores on the interface are trapped. Remove each such pore (if any) from the list of pores on the interface.

Determining which pores on the interface have become trapped is both conceptually and computationally more complex. In the brute-force approach, where a path is traced from each pore on the interface to the sink, the time required is of order N^2 at each step. Since there are N^2 steps, this leads to a time proportional to N^4 for the complete simulation. More-sophisticated techniques make use of the fact that each step in a simulation with a co-ordination number σ divides the bed into at most $\sigma - 1$ disconnected regions containing oil. Each of these regions may be explored (by tracing the perimeter or spanning the volume for instance) in time proportional to N. This yields a simulation with time proportional to N^3.

In two dimensions, tracing the perimeter is fast and simple since there is a natural ordering associated with throats on the interface (i.e. one can trace them in a clockwise direction). In three dimensions, there is no such simple ordering, although an algorithm due to Artzy, Frieder & Herman (1978) associates a digraph (directed graph) with the surface bounding a volume consisting of cubic volume elements in such a way that the surface can be easily traced. Expansion of this idea to volumes with connectivity other than simple-cubic appears non-trivial, however, so in order to maintain a consistent technique for simulations involving various co-ordination numbers in two or three dimensions led to the choice of an area-spanning (in two dimensions) or volume-spanning (in three dimensions) method.

Further time savings are made by using the fact that most steps of the simulation involve no trapping of oil. Organizing the exploration of the $\sigma - 1$ potentially disconnected regions in a manner that quickly discovers that they are in fact connected limits the time spent in this step to be proportional to the trapped oil discovered at each step. A classical-breadth first search (where the search for connected neighbours close to the starting pore is done before search of further neighbours) meets this requirement. Thus the total time is proportional to the amount of trapped oil. Running the simulation has shown that this is roughly proportional to N^2.

(v) If there are any pores left on the interface, return to step (ii) and repeat the procedure with the next-lowest-ranking throat. If there are no pores on the interface, the simulation is complete.

4. Results and discussion

Qualitative features

Figure 5 shows a typical fluid-displacement process on a periodic 100×150 rectangular lattice, showing the pores occupied by oil at intervals of 2500 steps. The black dots

are oil-filled pores, the throats have not been drawn, and the white region has been invaded by water. The key qualitative features to note are the irregularity of the interface, and the relatively large regions of trapped oil. If one examines the process step by step there is an alternation of two kinds of motion: an almost-smooth advance of the interface through a pore or two in different locations, and a 'fingering' process in which the interface comes to a sequence of high-ranking throats and runs through a long tortuous path of random orientation. Large trapped oil blobs result when a region is pinched off by long fingers.

In figure 6, we illustrate the effects of the boundary conditions on the sides of the lattice. In the first case the side boundaries are permeable, as is the oil end of the lattice, and the unsurprising result is that the sides are depleted of trapped oil. In the second case the side boundaries are impermeable to flow, and there results an excess of trapped oil blobs pinned against the sides. The boundary regions extend surprisingly far into the interior of the lattice. These boundary regions have important experimental implications, since the width of a typical core of Berea sandstone used in laboratory studies, where usually the sides of the core are rendered impermeable by jacketing, is only about 100 pores. Our results indicate that laboratory case studies may seriously overestimate residual oil saturation at low C. We are not claiming that periodic boundary conditions are completely realistic, but rather that results from a jacketed core may not reflect the properties of a rock *in situ* in a reservoir.

Lattice dependence of S

For the remainder of this discussion we restrict ourselves to periodic boundary conditions on the sides, and ask how the volume fraction S of residual oil depends on the length L, width W and co-ordination number σ of the lattice. In figure 7, we plot the amount of trapped oil *vs.* row number (i.e. distance downstream) for a square lattice of width 50 and length 375, averaged over many runs. If the regions at the ends are disregarded as unrepresentative of the bulk of the lattice, then we find a 'central plateau' whose height shows no significant length dependence. The origin of this behaviour may be seen in figure 8, which shows the result of one flow on this 50×375 lattice: there is no evident systematic variation in the trapped oil blobs along the length. We infer that, once the length of the lattice approximately exceeds the width, there is no longer any variation of S with length. This has been verified by repeating the simulations at fixed W but varying L. There is a clear dependence of S on the width of the lattice. In figure 9 we plot the oil saturation in the central plateau region of a rectangular lattice *vs.* the width of the lattice in number of pores. All of our data, for all lengths exceeding W, are nicely fitted by the equation

$$S(W) = 1 - 0{\cdot}95 W^{-0{\cdot}17}. \tag{3}$$

The data has the *a priori* surprising feature that $S \to 1$ as $W \to \infty$, originating from the fact that water forms fingers through oil and surrounds regions that increase in size with the width of the lattice, while the fingers themselves (which contribute to *water* saturation) can remain at a narrow finite width. In view of our remarks at the end of § 2 we do not regard this last result as directly relevant to very wide lattices, or, for that matter, to macroscopic oil reservoirs (which have the additional complicating feature of statistical inhomogeneity). Equally surprising is the observation that (3) holds for every value of σ we have examined.

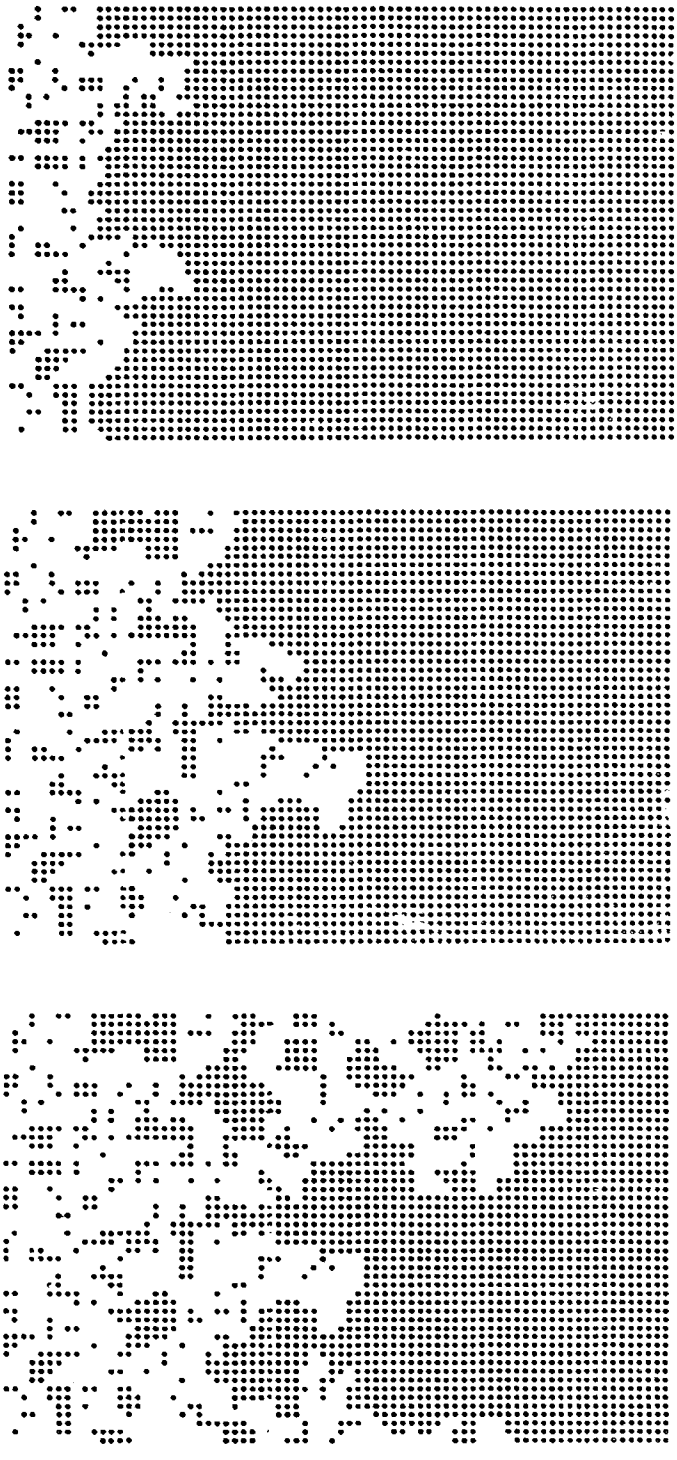


FIGURE 5. Sequence of fluid displacements on a 100×150 lattice at intervals of 2500 steps (water enters at left).

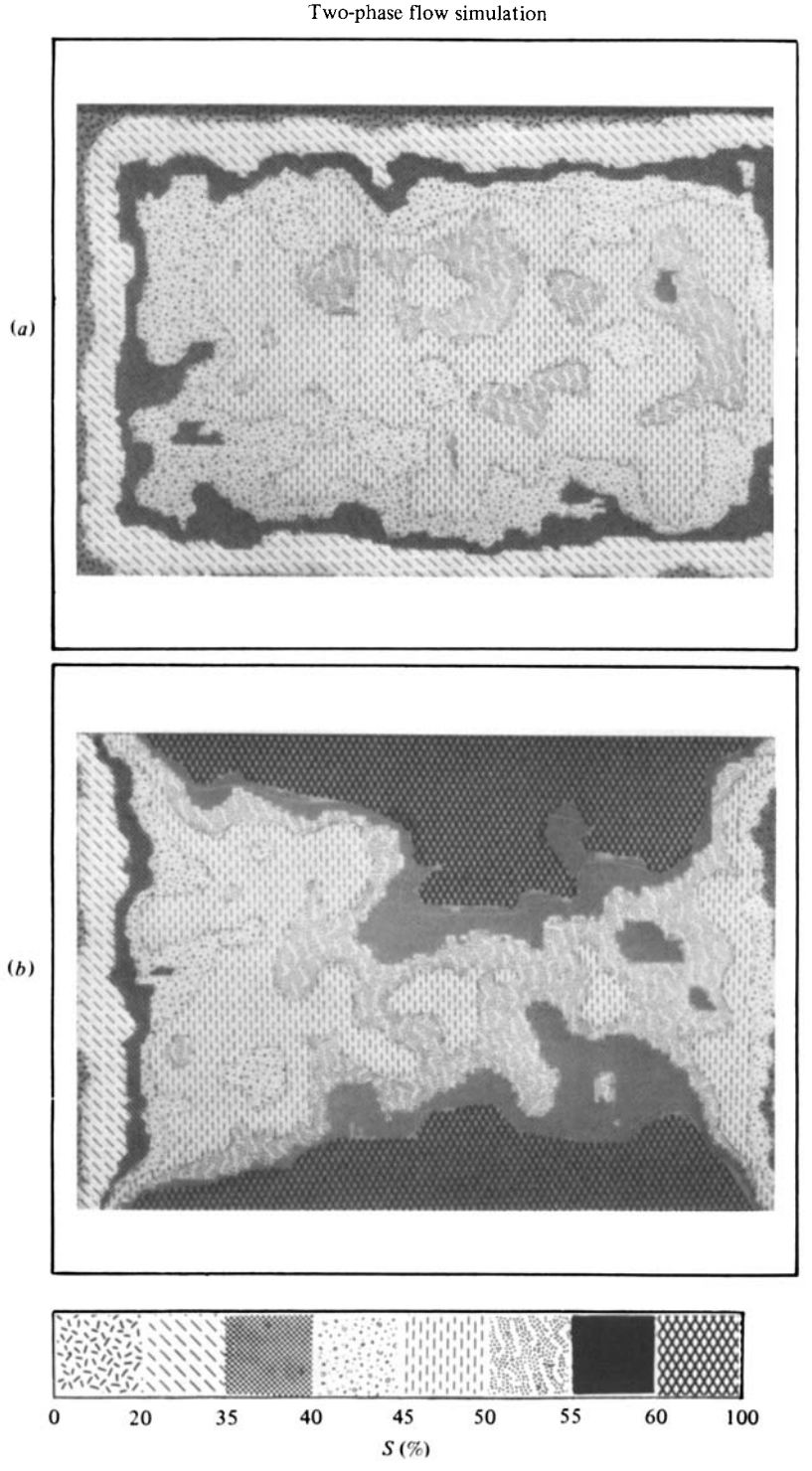


FIGURE 6. Residual saturation on a rectangular lattice with various boundary conditions and water entering at left: (*a*) permeable and (*b*) impermeable top and bottom boundaries.

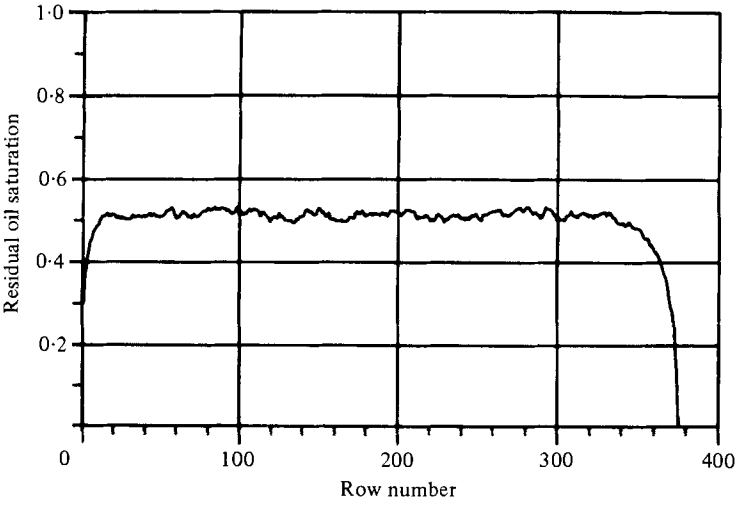


FIGURE 7. Residual saturation *vs.* distance downstream for a 50×375 lattice.

5. Percolation theory

Our aim in this section is to elucidate the manner in which constructs of percolation theory can be applied to the diphasic flow simulation described in earlier sections. The percolation problem was first stated by Broadbent & Hammersley (1957), a good general review is provided by Shante & Kirkpatrick (1973), and two more recent articles that emphasize aspects of percolation relevant to us are Kirkpatrick (1979) and Lubensky (1979).

Classical percolation

We begin by describing the classical 'bond' percolation problem. Consider an infinite regular lattice in which links are independently deleted at random. Suppose that p is the probability that any given bond is present and $1-p$ the corresponding probability that the bond is absent. At small values of p only a few bonds will be present and one expects these bonds to form small isolated clusters with no connected path of bonds traversing the lattice in any direction. In contrast, at values of p near 1 most bonds will be present and there should be a 'percolating cluster' of bonds running across the lattice in any direction. The nature of the transition between these two regimes is given by the fundamental theorem of percolation: there exists a critical probability p_c such that for $p < p_c$ there is no percolating cluster, while for $p > p_c$ there is exactly one. The value of p_c depends on the co-ordination number σ of the lattice as well as the dimension d of the space, and Monte Carlo studies indicate that

$$p_c \simeq d/(d-1)\,\sigma. \tag{4}$$

The details of the transition at p_c are typical of second-order phase transitions in statistical physics, as occur for example at the Curie temperature in a ferromagnet. One thinks of the percolating regime above p_c as analogous to the spontaneously ordered magnetic phase that occurs in a ferromagnet below its critical temperature, and defines a 'percolation probability' $P(p)$ that is analogous to the magnetization.

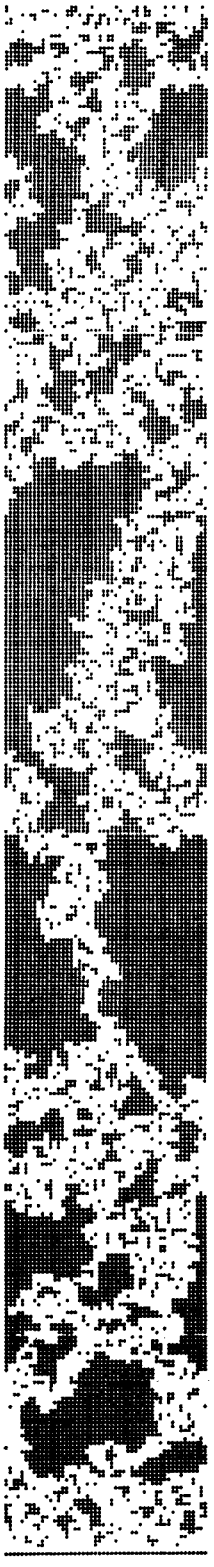


FIGURE 8. Example of a final state on a 50×375 lattice (water enters at top).

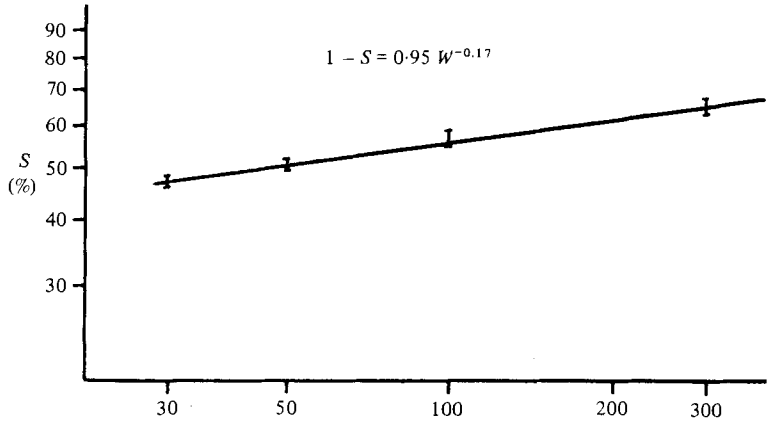


FIGURE 9. Residual oil saturation in the 'central plateau' *vs.* lattice width (once the length exceeds the width).

More precisely, $P(p)$ is the probability that a given bond is contained in the percolating cluster. The fundamental theorem of percolation implies that $P(p)$ is strictly zero below p_c. Just above p_c, numerical and theoretical arguments prescribe a power-law behaviour for $P(p)$:

$$P(p) \sim P_0(p-p_c)^{\beta}. \tag{5}$$

While p_c and P_0 depend on the particular lattice in question, the so-called critical exponent β exhibits the remarkable property of *universality*: it depends only on the spatial dimension and is independent of the co-ordination number and other properties of the system. A similar behaviour occurs for a second quantity of interest, the correlation length. If we define $g(r-r')$ to be the probability that points r and r' are connected by bonds, then, when the distance $|r-r'|$ is large and $p<p_c$, g behaves as

$$g(r-r') \sim e^{-|r-r'|/\xi}, \tag{6}$$

which defines the correlation length $\xi(p)$. Near the critical point, as $p \to p_c$, ξ diverges with another universal critical exponent ν:

$$\xi(p) \sim \xi_0 |p-p_c|^{-\nu}. \tag{7}$$

In two and three dimensions, numerical studies give $\nu \simeq 1{\cdot}365$ and $0{\cdot}845$ respectively.

The intuitive reason for the existence of such universal critical-exponent scaling laws as (5) and (7) is that near a critical point or phase transition the gross behaviour of the system is controlled by large-scale co-operative interactions. Suitably chosen bulk properties are insensitive to fine-scale details of the system. The divergence of the correlation length at the critical point is a direct signal of this behaviour.

Instead of having bonds present in the lattice with some probability p, one can instead consider 'site percolation', where any *node* has a given probability of being present, and connected clusters consist of nearest-neighbour nodes. Site and bond percolation are found to behave rather similarly: in both cases one finds scaling laws of the form (5) and (7) near the critical point at which a percolating cluster

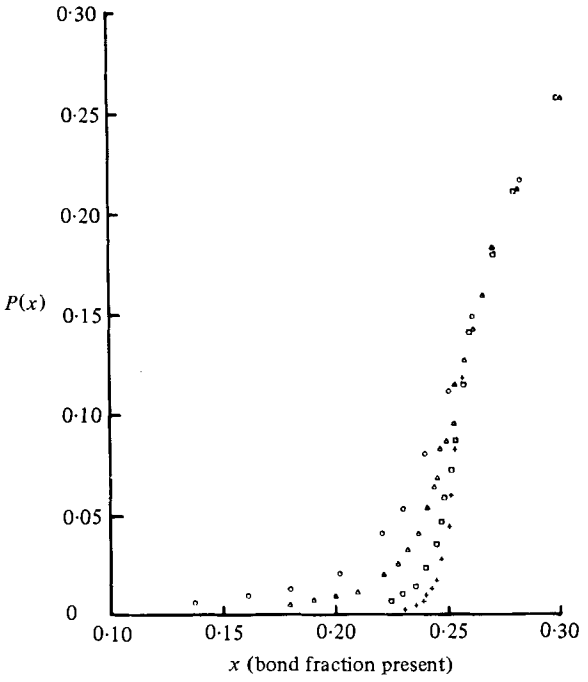


FIGURE 10. Finite-size dependence of $P(x)$ for a cubic lattice. Probability that a site belongs to a percolating cluster as function of bond fraction present, for different lattice sizes: circles represent $10 \times 10 \times 10$ samples, triangles represent $20 \times 20 \times 20$, squares represent $50 \times 50 \times 50$, and diamonds represent $80 \times 80 \times 80$. (Courtesy of Kirkpatrick 1979.)

appears. The value of p_c is different for site and bond percolation on the same lattice, but the universal critical exponents β and ν are the same.

Finite-size scaling

In practice one never deals with infinite systems, and it is necessary to consider the modifications to the above picture when the lattice is finite. The fundamental theorem of percolation is no longer exact, and percolating clusters may exist even when $p < p_c$. However, the likelihood of finding a percolating cluster below p_c decreases rapidly with the size of the sample, and (see figure 10) for moderate sizes the main effect is to append a tail to $P(p)$ below p_c. More generally, it has long been known in the study of phase transitions that sharp (non-analytic) behaviour of physical quantities in the neighbourhood of a critical point is smoothed in finite samples.

The effects of finite size in this context have been investigated by Fisher (1971), and we can reproduce the crux of his results by the following heuristic argument. If we use (7) to eliminate $|p-p_c|$ in favour of the correlation length ξ, then from (5) the behaviour of the percolation probability P near the critical point is

$$P \sim \xi^{-\beta/\nu}. \tag{8}$$

Now suppose we have a sample of linear dimension N (measured in number of nodes). While in an infinite sample ξ diverges near the critical point, from its definition (6) the correlation length cannot exceed the overall size of a finite sample. This suggests

the identification ξ/N = constant near the 'smoothed' critical point $p_c(N)$, which leads to

$$P \sim N^{-\beta/\nu}. \tag{9}$$

Fisher's more detailed result is, for p near p_c,

$$P(p,N) \sim N^{-\beta/\nu} F\left(\frac{p-p_c}{p_c} N^{\nu}\right), \tag{10}$$

where F is an unknown function, together with a finite-size shift in the apparent percolation threshold:

$$p_c(\infty) - p_c(N) \sim N^{-1/\nu}. \tag{11}$$

If we evaluate (10) at $p = p_c(N)$, we obtain (9).

Percolation and the simulation

Although the study of flow in random media was one of the original motivations for the invention of percolation theory (Broadbent & Hammersley 1957), the simulation described in earlier sections is evidently not the same as the classical percolation problem just reviewed. The former is a process consisting of a sequence of motions in which each step is determined by the results of the previous one, while the latter provides a static configuration which either does or does not contain a percolating cluster. There are also further subtle differences to be noted below. Nevertheless, we shall argue that the final state of water produced in the simulation is qualitatively similar to a percolation cluster, and use this identification to analyse some of our results.

We have found that, provided $L > W$, S is independent of L. (As has been noted earlier, end effects are excluded from this consideration.) In spite of the presence of significant water fingering in the intermediate stages of the simulation, on average the water front advances through the lattice in what is best described as a uniform mean flow. That is, the amount of residual oil is, on average over many realizations, simply proportional to the length, so the residual oil *fraction* is independent of L. The above remark implies that any block of finite length larger than W (excluding end regions) is representative of a 'steady-state' distribution of residual oil. We note in passing that the observation that a steady state is achieved only when L is larger than order W also implies than in the limit of infinite W no true steady state will appear.

In the simulation, the probability that a given bond has a rank in the interval $[x, x+dx]$ is just dx, since the ranks are assigned randomly in $[0, 1]$. Suppose we consider just those bonds with ranks in the interval $[0, p_c]$; these comprise a fraction

$$\int_0^{p_c} dx = p_c$$

of the bonds in the lattice, and the fundamental theorem of percolation ensures that this subset of bonds will form a percolating cluster. At any step in the simulation, the interface between oil and water regions *must* cross the percolating cluster (because the latter provides a path of bonds traversing the lattice in any direction). Therefore, at the next step of the simulation there is guaranteed to be a bond with rank $x \leqslant p_c$,

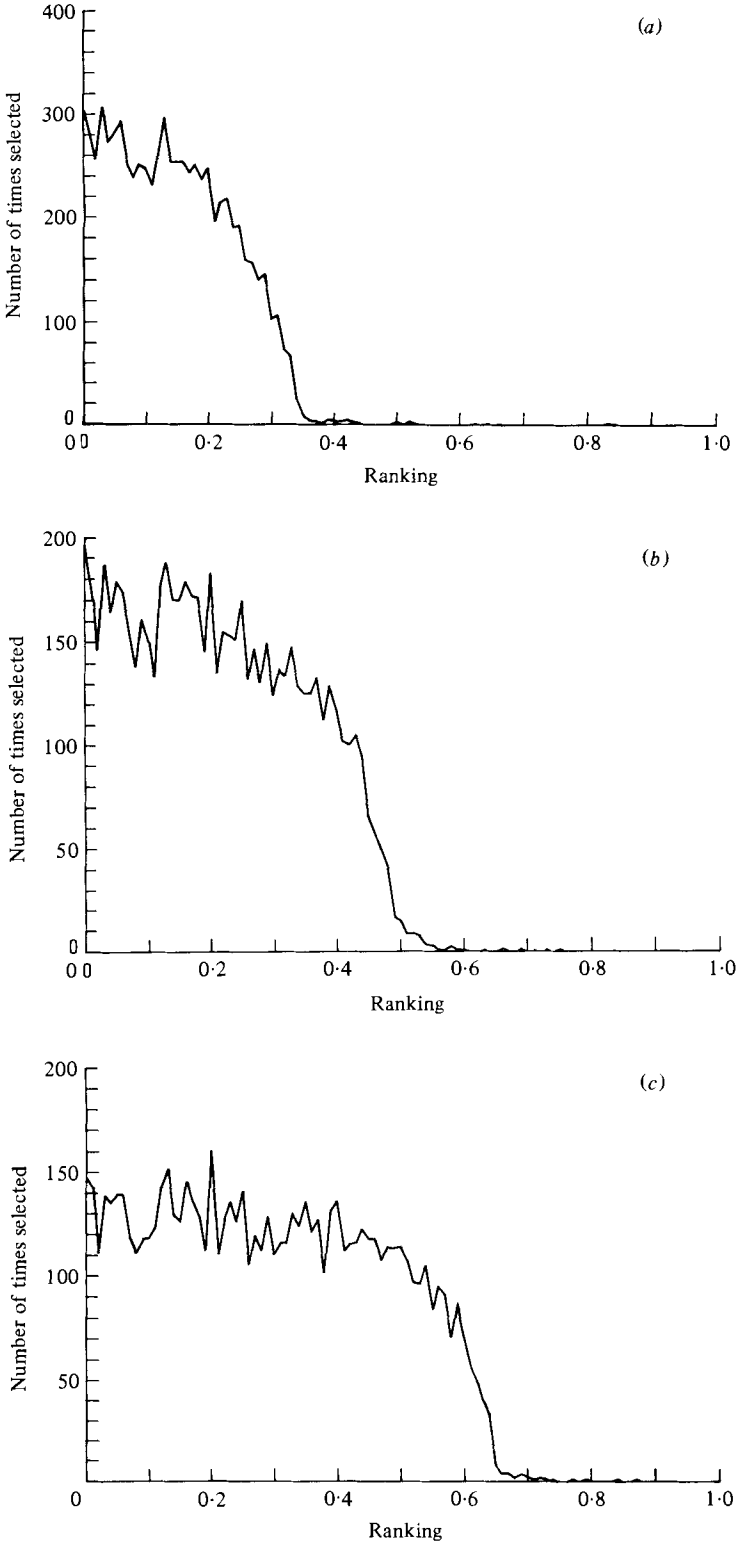


FIGURE 11. Histogram of bonds selected *vs.* ranking for various co-ordination numbers: (*a*) hexagonal, $\sigma = 6$; (*b*) rectangulor, $\sigma = 4$; (*c*) triangular, $\sigma = 3$.

and so the bond chosen for the advance of the interface must lie in the percolating cluster. The motion of the interface thus selects only bonds whose rank lies in the interval $[0, p_c]$.

The evidence for the latter assertion is in figure 11, where we show histograms of the ranks selected for $\sigma = 3$, 4 and 6 in one realization of the simulation. The percolation thresholds in the infinite-size limit for this case are at $\frac{1}{3}$, $\frac{1}{2}$ and $\frac{2}{3}$ respectively, and the curves approximately cut off at these points. The tails to the right are a mixture of two effects: a finite-size correction of the fundamental theorem given by (11), and the inappropriate inclusion of the regions at the ends of the sample.

The fact that in figure 11 not all of the bonds with $x < p_c$ have been filled with water reflects the difference between the simulation and pure percolation, namely that not all of the percolating cluster need fill with water. One reason is that in the simulation, when a region of oil is surrounded by water, no further activity takes place there, despite the fact that some of the bonds in the interior may have rank $x < p_c$. A second discrepancy occurs when water fills two nearest-neighbour pores before the throat between them has filled. Motion will never occur in this throat thereafter, whatever its rank.

Residual oil and finite-size scaling

We are now in a position to make theoretical contact with our numerical result (3). Recall that S is the fraction of sites that contain oil at the end of the simulation, so it is the water fraction $1-S$ that corresponds to the percolating phase. We must consider finite-size percolation, where, since the sample length is irrelevant for $L \gtrsim W$, the relevant linear size parameter is the width W. Classical percolation then predicts

$$1-S = AW^{-\beta/\nu}, \tag{12}$$

where the universal critical exponent $\beta/\nu = 0{\cdot}11$ in two dimensions, and the not-necessarily-universal coefficient A is unknown. We obtain qualitative agreement, in that our result for $1-S$ is power-behaved with an exponent of 0·17. The sign of the discrepancy in the exponent is in accord with our remarks in the previous paragraph, in that the simulation should have less water than would be expected from the size of the classical percolating cluster. The observed universality of the constant A is a slight surprise, but consistent with approximate calculations which show that it is a weak function of σ.

The fact that the agreement between theory and (numerical) experiment is inexact is not too surprising, given that the assumptions of the simulation correspond only roughly to those of percolation and we should expect only rough agreement. However, the threshold behaviour exhibited in figure 11 and the universal scaling form of (3) are characteristic of a wide class of phase-transition phenomena, and suggest that the simulation is indeed at a critical point.

6. Summary of results

We have studied two-dimensional capillary displacement by computer simulation, abstracting the essential features as a stepwise displacement process on a regular network of random-sized links. The simulation results in a moving interface charac-

terized by fingering at every scale, trapping clusters of residual oil. By counting the equally weighted nodes as the sole contributors towards fluid volume, we find that the steady-state residual oil saturation is independent of topology (as expressed by the co-ordination number) and depends only on the lattice size. Many features of the simulation, including this finite-size effect, are predictable from percolation theory and suggest that such displacement processes are indeed critical phenomena. Percolation theory will not describe all of the essential features of the displacement process; we have focused here on the volume fraction of residual oil. Nevertheless, the theory offers considerable guidance in extending our understanding of capillary displacement to large scale and three dimensions, where direct observation of the details of these processes is experimentally difficult.

We thank R. Dashen for discussions which led us to think about percolation, M. Gouilloud for inspiring this project, and M. E. Fisher.

REFERENCES

ANDROUTSOPOULOS, G. P. & MANN, R. 1979 *Chem. Engng Sci.* **34**, 1203.

ARTZY, E., FRIEDER, G. & HERMAN, G. T. 1978 *Computer Graphics*, **14**, 2.

BROADBENT, S. R. & HAMMERSLEY, J. M. 1957 *Proc. Camb. Phil. Soc.* **53**, 629.

CHATZIS, G. & DULLIEN, F. A. L. 1977 *J. Can. Petrol. Tech.* Jan.–Mar. 1977, 97.

DE GENNES, P. G. & GUYON, E. 1978 *J. Méc.* **17**, 403.

DULLIEN, F. A. L. 1975 *A.I.C.E. J.* **21**, 299.

FISHER, M. E. 1971 In *Critical Phenomena, Proc. Int. School of Physics E. Fermi, Course LI* (ed. M. S. Green). Academic.

KIRKPATRICK, S. 1979 In *Ill-Condensed Matter: 1978 Les Houches Lectures* (ed. R. Balian, R. Maynard & G. Toulouse), p. 321. North-Holland.

KNUTH, D. 1973 *Sorting and Searching: The Art of Computer Programming*, vol. 3, pp. 150–152. Addison-Wesley.

LENORMAND, R. 1980 *C. R. Acad. Sci. Paris* B, **291**, 279.

LUBENSKY, T. C. 1979 In *Ill-Condensed Matter: 1978 Les Houches Lectures* (ed. R. Balian, R. Maynard & G. Toulouse), p. 405. North-Holland.

MELROSE, J. C. & BRANDNER, C. F. 1974 *J. Can. Petrol. Tech.* Oct.–Dec. 1974, 54.

MOHANTY, K. K., DAVIS, H. T. & SCRIVEN, L. E. 1980 In *Proc. 55th Fall Technical Conf. of the SPE*. Preprint SPE 9406.

SCHEIDEGGER, A. E. 1974 *The Physics of Flow in Porous Media*. Toronto University Press.

SHANTE, V. K. S. & KIRKPATRICK, S. 1971 *Adv. Phys.* **20**, 325.

SUR, A., LEBOWITZ, J. L., MARRO, J., KALOS, M. H. & KIRKPATRICK, S. 1976 *J. Stat. Phys.* **15**, 345.

VAUCHER, J. G. & DUVAL, P. 1975 A Comparison of simulation event list algorithms. *Commun. ACM* **18**, 223.