

Computer simulations of fluid flow in sediment: From images to permeability

JONAS TÖLKE, CHUCK BALDWIN, YAOMING MU, NAUM DERZHI, QIAN FANG, and AVRAMI GRADER, Ingrain
JACK DVORKIN, Stanford University

Permeability, unlike porosity, saturation, and lithology, is, arguably, impossible to infer from well data. Yet, it is the key parameter in reservoir simulation and also in well completion and stimulation decisions. The traditional method of obtaining permeability is in the physical laboratory where a pressure difference (ΔP) is applied to the opposite sides of a regularly shaped rock sample and the volumetric flux (Q) of the fluid is measured.

The absolute permeability k_{Absolute} (or the single-phase permeability) is defined as

$$k_{\text{Absolute}} = -Q \frac{\mu}{A \Delta P} L, \quad (1)$$

where μ is the dynamic viscosity of the fluid, A is the cross-sectional area of the sample, and L is its length. k_{Absolute} has units m^2 . In the industry it is reported in millidarcy (mD) with $1 \text{ mD} \approx 10^{-15} \text{ m}^2$.

Typically, at least two fluid phases are simultaneously present in porous rock. To relate pressure drop to the movement of these phases through the reservoir, relative permeability is used (e.g., k_{ro} for oil and k_{rw} for water):

$$k_{ro} = -\frac{Q_o \mu_o}{k_{\text{absolute}} A \Delta P} L, \quad k_{rw} = -\frac{Q_w \mu_w}{k_{\text{absolute}} A \Delta P} L, \quad (2)$$

where the subscripts “o” and “w” refer to oil and water, respectively. The fluxes Q_o and Q_w are usually measured at fixed water saturation S_w .

An alternative to physical measurements of permeability, which are fairly laborious and time-consuming, especially for two-phase flow, is computational simulation of flow in the pore space. First, a small sample of rock is CT-scanned to obtain its gray-scale image at resolution required to accurately resolve the pore conduits where fluid flow occurs (Figure 1).

Next, the image is processed (segmented) to precisely identify the mineral/pore interfaces and, by so doing, resolve the pore space in all its natural complexity (see example at the top of Figure 6). Finally, viscous fluid flow is simulated in this pore space and permeability is determined from Equations 1 and 2.

Slow incompressible flow of a viscous fluid is governed by the Navier-Stokes equations which, for a simple case of stationary axial flow in a circular pipe, become

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} = \frac{1}{\mu} \frac{dP}{dx}, \quad (3)$$

where u is the velocity of the fluid in the axial direction x , and r is the radial coordinate directed outward from the central axis of the pipe.

An important property of this flow is that its velocity is

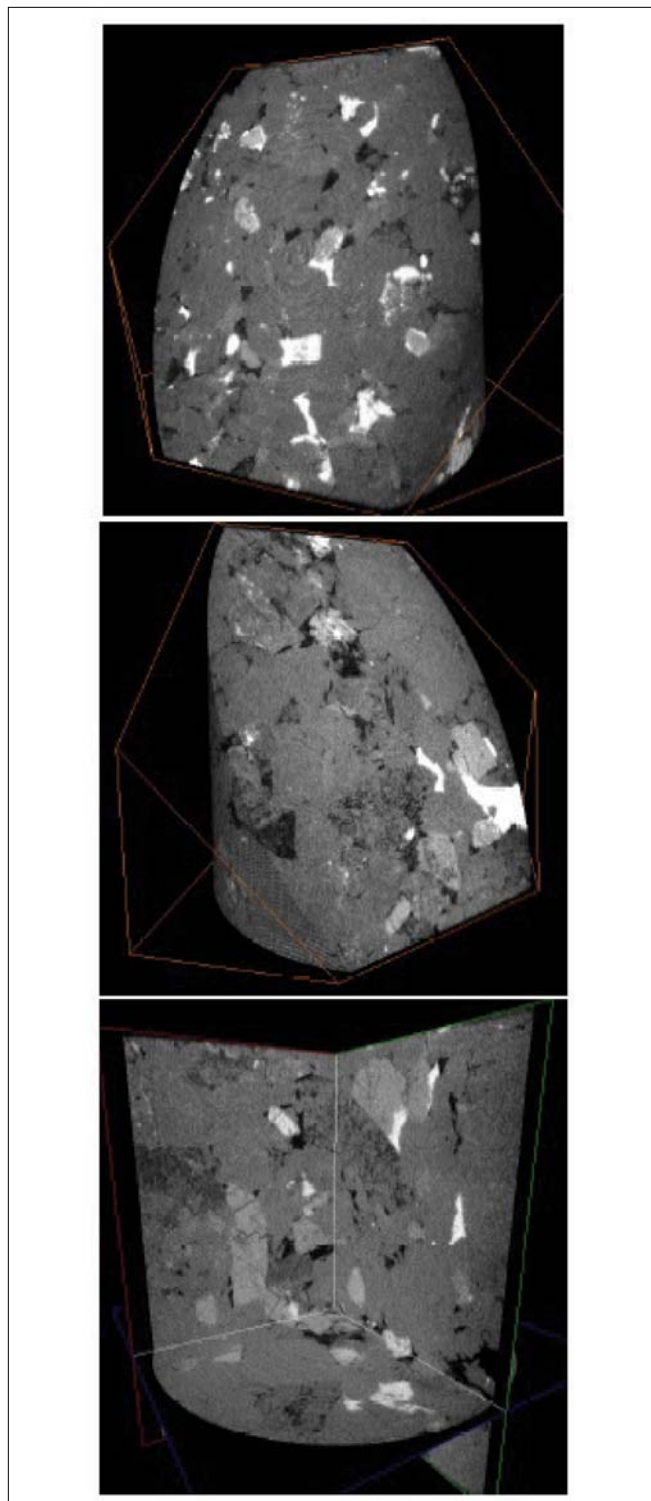


Figure 1. Gray-scale 3D images of a tight sandstone sample with three different sections of the cylinder shown. Pores are black, quartz is gray, and carbonate inclusions are white.

zero at the walls of the pipe—the so-called no-slip boundary condition $u = 0$ at $r = a$. With this condition, the solution of Equation 3 is

$$u(r) = -\frac{1}{4\mu} \frac{dP}{dx} a^2 \left(1 - \frac{r^2}{a^2}\right), \quad (4)$$

so that u is maximum in the center of the pipe and zero at the walls where $r = a$.

It is clear from Figure 1 that real pores are not round pipes. Nevertheless, the three-dimensional Navier-Stokes equations still govern the flow and can be discretized by a numerical scheme like finite differences. However, solving such finite-difference systems in a complex pore space is far from simple, if not impossible, especially where more than one fluid phase is present in the pore space. The main reason is that the no-slip boundary condition has to be satisfied on an extremely intricate boundary surface—the pore/solid interface—which results in a very large system of equations to be numerically solved. Moreover, in the case of multiple fluid phases, the moving fluid-fluid interface has to be incorporated and appropriate balances have to be satisfied on this boundary.

One way of handling this problem is by replacing a realistic pore space with a complicated set of round pipes (conduits) and capacitors (volumes at the intersections of the pipes). Although such pore-network models can qualitatively mimic fluid flow in rock, by no means do they directly address the extreme complexity of a real pore space. Moreover, because a large number of setup parameters is required to construct a 3D idealized network model, it is hardly predictive and, essentially, self-fulfilling, not necessarily providing a physically justifiable permeability if the latter is not known in advance. An alternative, the lattice Boltzmann method, discussed below appears to be more appropriate to handle the task at hand.

Lattice Boltzmann method

In the last two decades, the lattice Boltzmann method (LBM) has matured to become an efficient numerical scheme for simulations of fluid flow. Unlike conventional numerical schemes based on discretization of macroscopic continuum equations, LBM is based on microscopic models and mesoscopic kinetic equations. The fundamental idea of this method is to construct simplified kinetic models that incorporate the physics of microscopic and/or mesoscopic phenomena in a way that the macroscopic-averaged properties obey the desired continuum equations.

Unlike the traditional computational fluid-dynamics methods, which numerically solve the conservation equations of mass, momentum, and energy, LBM models the fluid as consisting of fictitious particles (mass fractions). These particles propagate and collide on a regular lattice. In the collision, the mass fractions are modified in such a way that mass and momentum are conserved and that the correct relation between strain rate tensor and stresses as used in the Navier-Stokes equations is obtained.

The lattice has to be chosen in such a way that a space-filling partition results and that the stress tensor can be rep-

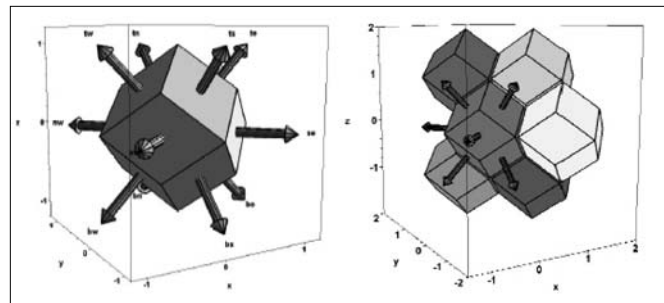


Figure 2. (left) Unit cell with 13 particle velocity directions. (right) Space filling by rhombic dodecahedrons.

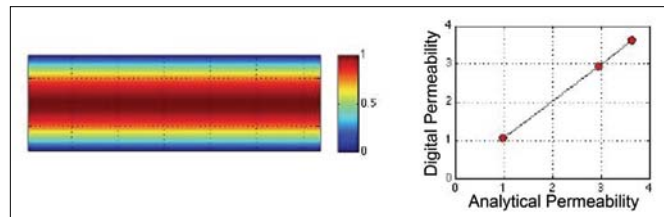


Figure 3. (left) LBM velocity field (normalized by maximum velocity in the center) computed in a circular pipe. Viscous fluid moves in the horizontal direction. (right) LBM permeability computed for rectangular conduits versus analytical solutions. The axis units are the decimal logarithm of permeability (mD).

resented correctly, meaning that we provide enough lattice directions and lattice symmetry to construct this tensor. An example of such a lattice with 13 particle velocity directions and the corresponding 3D unit cell is shown in Figure 2. This unit cell is a rhombic dodecahedron and generates a space-filling lattice. The directions are marked as, e.g., “top-west” (tw) and “bottom-east” (be). Other 3D schemes may use 15, 19, or 27 directions.

The lattice-Boltzmann equation is

$$f_i(t + \Delta t, \mathbf{x} + \mathbf{e}_i \Delta t) = f_i(t, \mathbf{x}) + \Omega_i(f_j), \quad (5)$$

where i is a number corresponding to each of the 13 directions shown in Figure 2; f_i is the mass fraction in a given direction; t and Δt are the time and time increment, respectively; \mathbf{x} is the location of the lattice nodes and \mathbf{e}_i is the particle velocity vector; and $i = 0, \dots, 12$; $j = 0, \dots, 12$.

The operator Ω_i models the collision of the particles in a lattice node. This operator is designed to relax the particles $f_i(t, \mathbf{x})$ toward an equilibrium state. The strength of the relaxation is controlled by relaxation rates, which are directly linked to the viscosity of the fluid.

Even though LBM is based on particulate motion, its principal focus is the averaged macroscopic behavior. Indeed, it is possible to recover the governing continuity and Navier-Stokes equations from the LBM algorithm (Frisch et al., 1987). This scheme is particularly convenient in problems where transport problems are described by advective and diffusive operators. An example is transport phenomenon described by mass and momentum conservation and Newton’s linear viscous friction law—the Navier-Stokes equations.

Due to its particulate nature and local dynamics, LBM

has several advantages over other conventional computational fluid-dynamics methods, specifically in dealing with *complex boundaries* (as inside real rock), incorporating of microscopic interactions (such as surface tension between two fluid phases and wettability between fluid and solid), and *parallelization of the algorithm*.

Single-phase fluid flow

We have tested a single-phase LBM to assess its accuracy for various simple conduit shapes with existing analytical solutions. These digital results provide correct velocity fields as well as match analytically derived permeability for flow in pipes and rectangles (Figure 3). Examples in, e.g., Dvorkin et al. (2008), show that our LBM calculations also match physical data in sandstones and carbonates.

In spite of abundant optimistic reports of LBM application to viscous flow, we have encountered, in our tests and during practical assembly line implementations, several serious problems which we have successfully addressed. One of them, very fundamental, is the dependence of the computed permeability on the fluid's viscosity as well as on pressure gradient. This effect is contrary to the definition given by Equation 1, where the measured flux is normalized by the viscosity and pressure gradient.

The second problem is the accuracy for flow in under-resolved pores. This problem is illustrated in Figure 3, where the LBM permeability slightly exceeds the analytical value as the number of lattice nodes resolving the conduit becomes small. A solution is by modifying collision operators at the pore walls or, simply, by obtaining digital images at higher resolution.

Finally, we move towards drastically reducing the usually large computation times for transient 3D simulations. This is achieved by increasing the single-core performance as well as the parallelization of LBM codes. The scalability of our LBM codes is illustrated in Figure 4 for experiments on five different high-performance computing clusters (named here RE, GR, BR, OR, and GO), all with processors with multiple cores. Each experiment was run on a 3D digital image of a glass bead pack represented by a $250 \times 250 \times 250$ rectangular grid (about 15.5 million grid cells). The code scales well with varying configurations as well as across different processor/core architectures. The slowest computing time, using just one core, was in the range of a day while the fastest was done in a couple of minutes by using a graphical processing unit (GPU) with 240 cores (discussed below).

Two-phase fluid flow

Immiscible flow of two phases in pores is also described by the Navier-Stokes equations, separate for each fluid, wetting and nonwetting, and an interface condition at the moving fluid-fluid interface, including the capillary pressure. The pressure difference $p_{1,2}$ at the interface of the two fluid phases depends on the interfacial tension γ and the two principal radii, R_1 and R_2 , of curvature of the interface surface: $p_{1,2} = \gamma(R_1^{-1} + R_2^{-1})$. Another important parameter of a two-phase flow is the contact angle θ at the fluid/fluid/solid

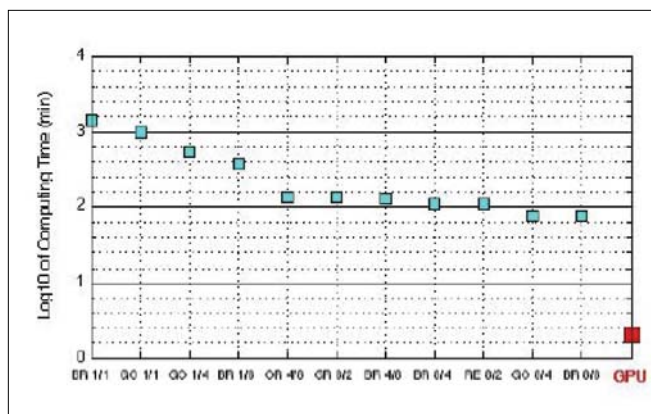


Figure 4. Decimal logarithm of computing time versus cluster, configuration, and hardware. For example, 8/4 is for 8 processors and 4 cores per processor. The fastest computing time was achieved on a graphical processing unit and took just a few minutes (red symbol).

boundary. For the wetting phase, this angle is less than 90° while for the nonwetting phase it is larger than 90° . At a triple junction of solid wall, wetting phase, and nonwetting phase, the contact angles simply add up to 180° , so that $\theta_1 = 180^\circ - \theta_2$, where the subscripts "1" and "2" stand for the wetting and nonwetting phase, respectively. At the triple junction, two other forces (in addition to γ) affect fluid dynamics: interfacial tension between the wetting phase and solid ($\sigma_{1,s}$) and between the nonwetting phase and solid ($\sigma_{2,s}$). The contact angle depends on these parameters as described by Young's equation: $\cos\theta_1 = (\sigma_{1,s} - \sigma_{2,s})/\gamma$.

In a two-phase implementation of LBM, a second field representing the distribution of the phases has to be added to the first field and the collision operator has to be extended to include these interfacial forces. Mass and momentum are conserved for both phases locally at each lattice node (Ahrenholz et al., 2008).

Such computations are especially instructive for simple configurations, such as shown in Figure 5. Here a square-cross-section cylinder split with a solid bar placed \hat{S} in its upper part is initially filled with a nonwetting phase. Then a wetting phase is injected under pressure from the left. It forces the nonwetting phase out of the bottom section quickly while the upper section is slower in evacuating. Once the bottom is filled, a small tube of non-wetting fluid is formed in the top section. This tube allows the nonwetting phase to completely exit the cylinder.

The observed behavior appears to be physically reasonable yet somewhat unexpected. The latter implies that LBM digital experiments allow us to make discoveries rather than blindly reproduce preconceived behavior.

As with any developing technique, two-phase LBM has its share of practical problems, among them: (a) long runtimes for time-resolved 3D multiphase simulations; (b) computational accuracy for large viscosity ratios between the phases as well as for high interfacial tensions and extremely small or extremely large contact angles; and (c) computational accuracy for under-resolved pores as well as at large content of one of the phases and, respectively, small content of the other.

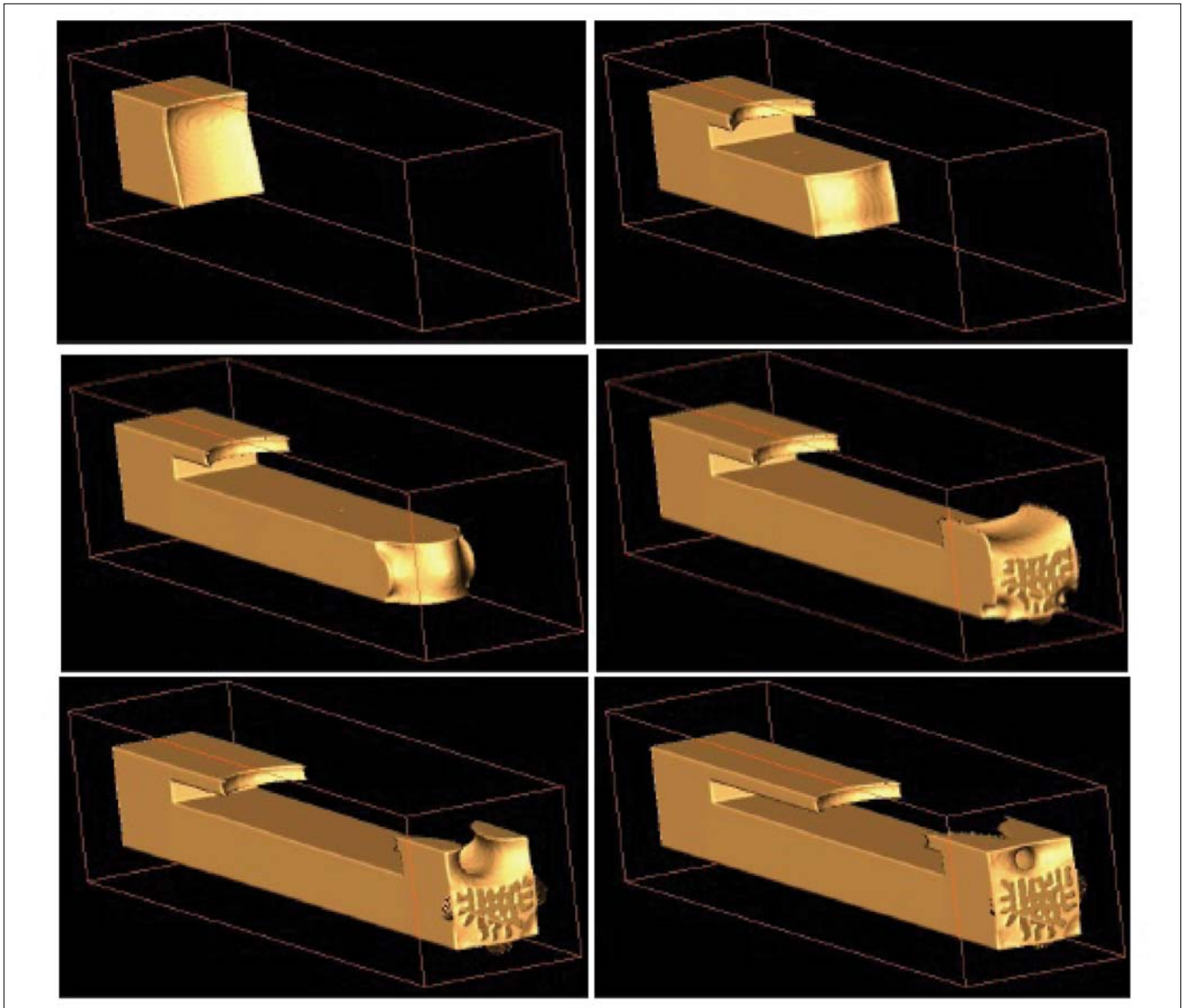


Figure 5. Split cylinder with square cross section gradually filled with wetting fluid, which replaces the nonwetting phase (left to right and top to bottom). The solid walls and nonwetting fluid are transparent.

High-performance computing using modern many-core systems

One of the most impressive recent developments in the area of high-performance computing is the transformation of graphical processing units (GPU) to general purpose compute devices. A GPU has hundreds of cores on a single chip and can be easily installed in standard hardware. Because LBM operates on a finite-difference grid, is explicit in nature, and requires only next-neighbor interactions, it is suitable for implementation on GPU.

A schematic description of the algorithm is as follows: The initial setup (importing the geometry and setting the initial conditions) is done on the host CPU. Then all data are transferred to the GPU memory and the simulation is running solely on the GPU. Only the data needed for postprocessing purposes are transferred from GPU to CPU.

Since GPU hardware allows for very fine grain parallelism, every lattice site can be handled by a different core. Data

have to be loaded from and stored to the device memory in such a way that dense access to the memory is ensured. This can be achieved by accessing the lattice nodes with respect to their contiguous memory locations (Tölke and Krafczyk, 2008).

A simulation of an imbibition process using a GPU-LBM implementation running in this case on 240 cores is illustrated in Figure 6. The size of this digital object that represents micrite carbonate was $250 \times 300 \times 300$ (22.5 million) voxels. The computation required 100,000 time steps and was completed in just 90 minutes.

Fractional flow and relative permeability

This computational efficiency and accuracy allows us to simulate two-phase flow in realistic pore systems in a short time, as opposed to physical laboratory experiments that can take days, weeks, and even months.

To verify such results, we have performed simulations on

Figure 6. (top) Segmented 3D image of micritic carbonate. Three following images (from top to bottom): gradual drainage and replacement of water (wetting phase) by oil (nonwetting phase) in this digital sample. Oil = cyan. Water is transparent.

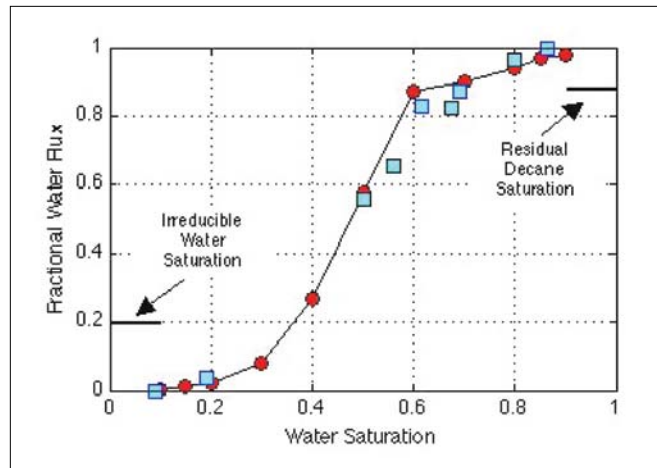
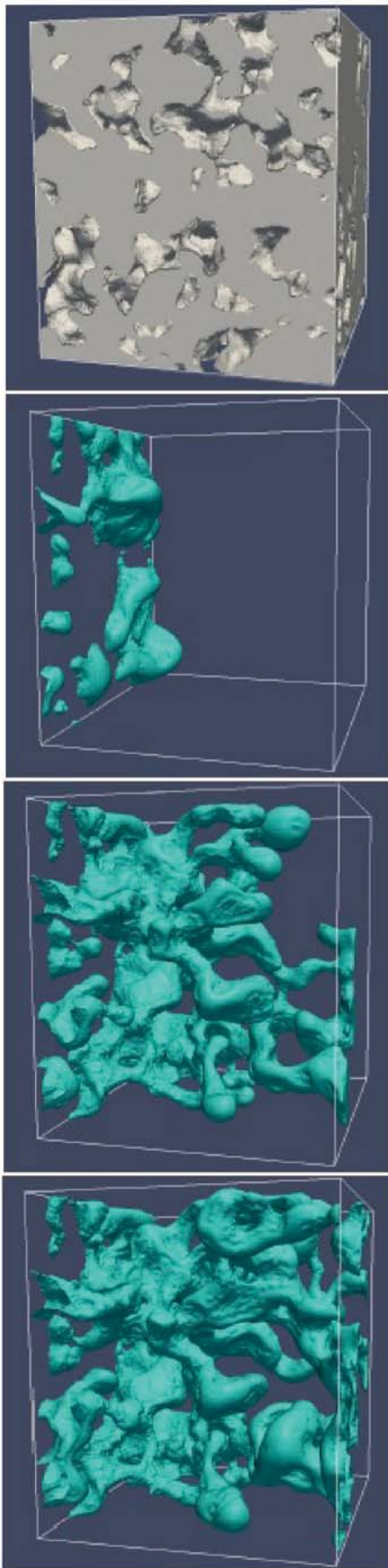


Figure 7. Fractional water flow versus water saturation in a glass-bead pack. Red = digital data. Blue = physical experiment.

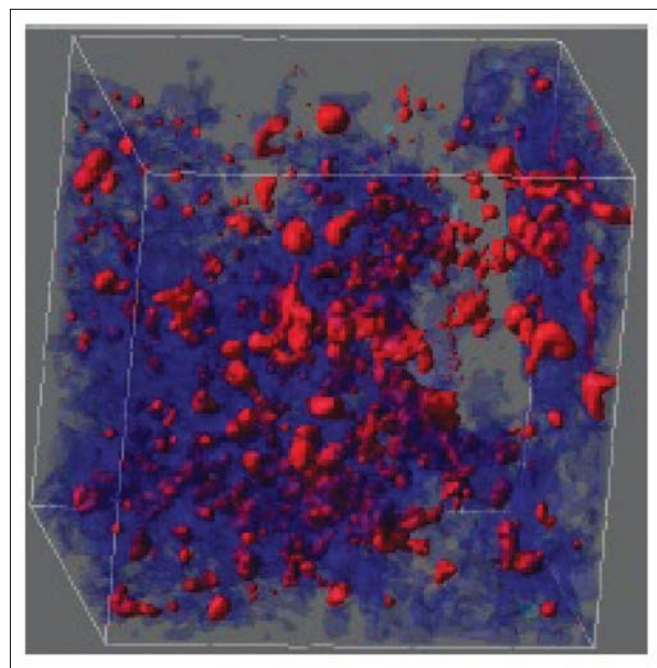


Figure 8. Digital LBM experiment. Nonwetting fluid phase (red) at its residual saturation after this phase was partially displaced by water (transparent) in a sandstone sample.

simple porous systems for which experimental data are available. An example of calculating fractional flow of water (the ratio of water flux to the combined flux of both fluids at fixed partial water saturation) in a water-wet bead pack is given in Figure 7. The other, nonwetting, phase was decane. The physical data have been obtained on a glass-bead pack with the bead size between 0.15 and 0.21 mm.

Part of this pack was later CT-scanned and two-phase LBM simulations performed on the resulting digital object. We observe a close match between the physical and digital results. What is especially important is that we were able to accurately match the two end points in this process: the irreducible water saturation (about 10%) and residual decane saturation (also about 10%).

Figure 8 helps visualize the geometry of nonwetting phase

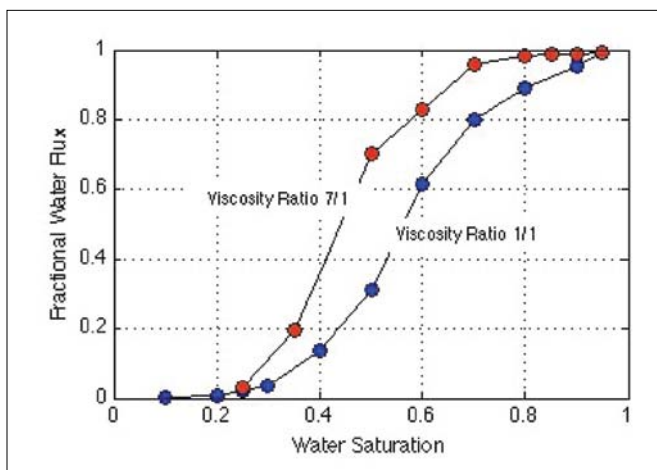


Figure 9. Digital experiment. Fractional water flow versus water saturation in a glass-bead pack. Blue = the viscosity of the nonwetting phase is the same as of water. Red = the viscosity of the nonwetting phase is 1/7 of that of water.

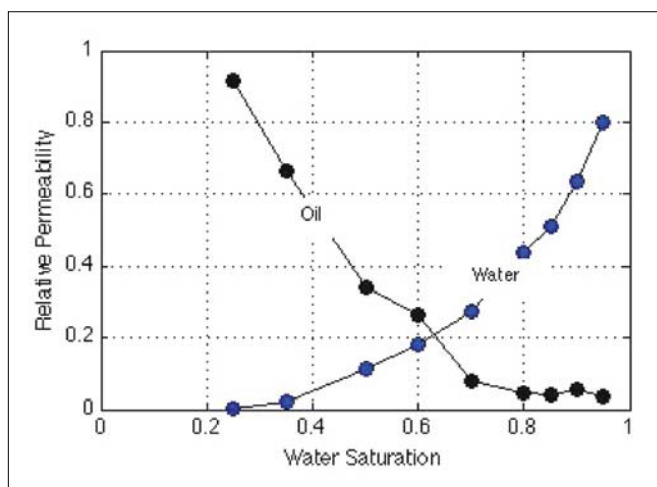


Figure 10. Digital experiment. Relative permeability for water (blue) and oil (black) for oil/water ratio = 1/7 and surface tension = 7 dyne/cm.

distribution in a real rock sample at such residual saturation.

Both the irreducible saturation and residual saturation are largely defined by the geometry of the pore space and capillary forces between the fluid phases and fluid and solid. However, the transition between these two endpoints is strongly influenced by the viscosities of the fluids (Figure 9). This is where the computational LBM technique becomes especially attractive as it is quite difficult to vary viscosity in the physical laboratory and obtain consistent results.

The inputs for LBM simulations are intrinsic fluid properties (viscosity and density) as well as interfacial properties (surface tension and wettability). All these parameters change during the lifespan of a reservoir due to pressure and temperature variations as well as chemical reactions. Moreover, even the pore-space geometry may change as hydrocarbons are produced, especially so in reactive carbonates. These conditions have to be provided by the reservoir engineer and geochemist. Once they are known or, at least constrained within a plausible range, LBM experiments can provide the final

practical deliverable—the relative permeability curves (Figure 10) at in-situ conditions to be used in reservoir simulation and forecasting.

Conclusions

LBM simulations of single-phase and multiphase fluid flow in a digitally imaged pore space offer a powerful means of obtaining permeability in different rock types and at varying in-situ conditions. The speedup of such computations implemented on graphical processing units make such experiments practical and massive. The result is a massive database that can be used to (a) quantify reservoir behavior throughout its lifespan and (b) derive and quantify relations between porosity, rock texture, and fluid properties and the absolute and relative permeability.

The main challenges that are being currently addressed are: (a) further speedup of computations using the newest hardware architectures and improved algorithms and (b) fluid-flow simulation in low-porosity and low-permeability formations, such as gas shale, and for large viscosity contrast, such as between water and gas.

The entire digital experimentation process on a single rock sample, starting with its 3D imaging and ending with flow simulation and permeability calculation, can be now conducted in a matter of several hours. Yet it is not intended to entirely replace physical results as they are required for occasional calibration and verification of digital data. Rather, digital rock physics is intended to drastically enrich and extend laboratory results into various rock types and fluid properties.

Finally, in dealing with real sediment that makes up real reservoirs, there is always a question of scale: How are digital data obtained on mm-sized samples or physical data obtained on cm-sized samples relevant to processes occurring at a scale of tens and hundreds of feet? One way of addressing this question is by conducting massive digital experiments and analyzing and synthesizing the results. **TLE**

References

- Ahrenholz, B., J. Tölke, P. Lehmann, A. Peters, A. Kaestner, M. Krafczyk, and W. Durner, 2008, Prediction of capillary hysteresis in porous material using lattice Boltzmann methods and comparison to experimental data and a morphological pore network model. *Advances in Water Resources*, 31, 9, 1151–1173.
- Dvorkin, J., M. Armbruster, C. Baldwin, Q. Fang, N. Derzhi, C. Gomez, B. Nur, A. Nur, and Y. Mu. 2008, The future of rock physics: computational methods versus lab testing. *First Break*, 26, 9, 63–68.
- Frisch, U., D. d'Humieres, B. Hasslacher, P. Lallemand, Y. Pomeau, and J.-P. Rivet, 1987, Lattice gas hydrodynamics in two and three dimensions. *Complex Systems*, 1, 75–136.
- Gunstensen, A. K. and D. Rothman, 1991, Lattice Boltzmann model of immiscible fluids. *Physical Review A*, 43, 4320–4327.
- Tölke, J. and M. Krafczyk, 2008, TeraFLOP computing on a desktop PC with GPUs for 3D CFD. *International Journal of Computational Fluid Dynamics*, 22, 7, 443–456.

Corresponding author: toelke@ingrainrocks.com