# Numerical Simulation of Two-Phase Flow Using Locally Refined Grids in Three Space Dimensions

*Adaptive local grid refinement, operator splitting, multigrid, and nonlinear characteristic convection enable efficient and accurate numerical simulation for a class of two-phase flow problems.*

W. A. Mulder and R. H. J. Gmelig Meyling, Shell Research B.V.


## SUMMARY

A numerical method for incompressible, immiscible, two-phase flow is presented, that uses dynamic local grid refinement and operator splitting. Elliptic equations are solved by a multigrid method, hyperbolic equations by a nonlinear characteristic method. Large time steps are permitted and high accuracy and efficiency are obtained if the total flow field does not strongly depend on the saturation distribution.


## INTRODUCTION

The objective of reservoir simulation is the computation of fluid flow through porous media with an accuracy that is sufficient for the prediction of the hydrocarbon recovery. The mathematical model for porous media flow consists of a coupled system of nonlinear partial differential equations that represents the basic physical properties of the true solution. Numerical methods are often required to determine an approximate solution of the system.

Some difficulties associated with these numerical methods are: numerical diffusion, severe time step restrictions, and high computational complexity (see Ref. 2 for a survey). Certain fluid flow processes, such as viscous fingering, depend critically on the amount of physical diffusion. Computational techniques that introduce strong numerical diffusion and smear sharp fluid interfaces are therefore unacceptable. Stability conditions associated with some methods require very small time steps and lead to large computational costs. Many flow problems involve highly localized phenomena, such as sharp fluid interfaces, point or line sources and sinks, and geological heterogeneities. For large-scale simulations, it is impossible to accurately represent all the local phenomena by covering the entire domain with a uniform fine grid.

In this paper, we present a numerical method for solving two-phase, immiscible flow problems. The method avoids the above mentioned computational difficulties through the combination of adaptive local grid refinement and special numerical techniques. Grid refinement can be carried out an arbitrary number of times in any region of space (see Ref. 3 for a survey of refinement strategies). The refinement is adaptive and dynamic, allowing fine blocks to move with fluid fronts. Static refinement is carried out near wells and near large permeability contrasts.

Operator splitting turns the mathematical model into a sequence of elliptic, hyperbolic, and parabolic problems. The elliptic and parabolic problem are discretized in space by the lowest order mixed finite element method of Raviart and Thomas[8]. This method provides a better accuracy near wells and strong heterogeneities than standard finite difference or finite element methods. The large linear system arising from the mixed finite element discretization is solved efficiently by a multigrid method[11]. The equation for the saturation has a convection and diffusion part. The diffusion models the effect of capillary pressure and is generally small compared to the convection term. The convection part is treated separately by transforming to streamline coordinates and integrating the one-dimensional nonlinear hyperbolic problem with the use of Riemann solvers. This characteristic technique has a very low numerical diffusion, captures sharp fronts, and allows for arbitrary large time steps. Here we use a slightly more general variant of the technique described in Refs. 4 and 5, employing Roe's approximate Riemann solver[9,10] rather than exact Riemann solvers.

The plan of the paper is as follows. First, we state the mathematical model for two-phase, immiscible flow. Secondly, the various aspects of the numerical method, namely operator splitting, mixed finite elements, multigrid, grid adaptation, convection, and diffusion, are described. Finally, we present some numerical examples to demonstrate the capabilities of the method.

## GOVERNING EQUATIONS

The flow of oil and water in a porous medium can be modeled by the following equations[1]:

$$\nabla.\vec{u} = q(\vec{x},t), \tag{1}$$

$$\vec{u} = -k(\vec{x})\lambda_t(S)\left[\nabla p - \bar{\rho}(S)\vec{g}\right], \tag{2}$$

$$\phi(\vec{x})\frac{\partial S}{\partial t} + \nabla.\left[f(S)\vec{u} + f_g(S)\vec{v} - D(S,\vec{x})\nabla S\right] = C(S,\vec{x})q(\vec{x},t). \tag{3}$$

For reservoir simulation, we have nonzero $q(\vec{x},t)$ at the wells, with $\int_\Omega q(\vec{x},t)d\vec{x} = 0$. Boundary conditions will be specified later on. We chose a composition factor $C(S,\vec{x}) = f(S)$ at the wells.

The incompressibility condition (1) and Darcy's law (2) form an elliptic system of equations. The convection-diffusion equation (3) is of parabolic type. Without diffusion, it is a hyperbolic equation.

## NUMERICAL METHOD

### Operator splitting

The system of equations (1)–(3) is integrated in time by means of operator splitting. First, we solve a mixed finite element discretization of (1) and (2) by a multigrid method[11]. Secondly, the convection part of (3) is treated by a nonlinear characteristic method. Thirdly, the diffusion part of (3) is discretized in time by a fully implicit scheme and in space by the mixed finite element method. The resulting system of equations is solved by the multigrid method and successive substitution (also known as Picard iteration).

We will describe the methods for convection and diffusion in more detail below. For the total flow $\vec{u}$ and pressure $p$, the lowest order mixed finite element representation of Raviart and Thomas[8] is used. The system (1)–(2) is solved by the multigrid method described in Ref. 11. This method finds an approximate solution to the mixed finite element discretization on a locally refined grid of

$$\begin{aligned} c\psi + \nabla.\vec{u} &= q, \\ \nabla\psi + W\vec{u} &= \vec{g}, \end{aligned} \tag{4}$$

subject to the boundary conditions

$$\vec{n}.\vec{u} = 0, \quad \text{on } \partial\Omega^{(1)} \subseteq \partial\Omega, \tag{5a}$$

and

$$\psi - \alpha\,\vec{n}.\vec{u} = \beta, \quad \text{on } \partial\Omega - \partial\Omega^{(1)}. \tag{5b}$$

Here $\vec{n}$ is the outward normal on the boundary $\partial\Omega$ of the domain $\Omega$. The scalar quantities $c$ and $q$ are defined per block and $c \geq 0$. The unknown scalar $\psi$ is also defined per block, whereas the components of the unknown vector $\vec{u}$ are defined on block faces. Only the normal component of $\vec{u}$ is defined on each face. The symmetric tensor $W$ and vector $\vec{g}$ are specified per block. Equations (1) and (2) can be expressed in the form (4) and solved by the multigrid method.

Once the total flow $\vec{u}$ at a given time is computed for a saturation distribution $S$, the convection-diffusion equation (3) is solved with a frozen flow field. Operator splitting is applied. First, we solve two convection equations of the form

$$\phi\frac{\partial S}{\partial t} + \nabla.f(S)\vec{u} = 0, \tag{6a}$$

and

$$\phi\frac{\partial S}{\partial t} + \nabla.f_g(S)\vec{v} = 0. \tag{6b}$$

The order in which this is done is alternated at every time step. Next, the source term $C(S,\vec{x})q(\vec{x},t)$ is treated. Finally, diffusion is taken into account by considering an equation of the form

$$\phi\frac{\partial S}{\partial t} - \nabla.\left[D(S,\vec{x})\nabla S\right] = 0. \tag{7}$$

Each of these steps will be described in the following sections. The convection equation is solved by a nonlinear characteristic method. The diffusion equation is discretized in time by a fully implicit scheme, that can be rewritten in the form (4) and solved by the multigrid method and successive substitution of the diffusion coefficient.

### Grid adaptation

For problems with different length scales, adaptive local grid refinement is highly desirable. We employed the approach described in Ref. 12, which allows for dynamic regridding and is geared towards the mixed finite element method and the multigrid solver.

A grid consists of *blocks* on which scalar quantities are defined, separated by *faces*, on which fluxes are stored. Each face carries only the normal component of the flux. The subroutines in our code are organized in such a way that the dimension can be chosen as 1, 2, or 3. A hierarchical grid structure is employed. We start with a static, rectangular base grid created by orthogonal "planes" (constant value of a space coordinate). The rectangular boxes are called *blocks*, and the separating "planes" *faces*. The base grid can be refined and unrefined by the *basic refinement*, which divides a block into $2^{N_d}$ identical smaller parts. Our code uses a cartesian coordinate system. The extension to curvi-linear coordinates is straightforward, but has not been carried out so far.

The grid is statically refined near wells. Dynamic grid adaptation is used for the saturation $S$ and the flow field $\vec{u}$. The regridding is controlled by the requirements

$$tol_{\min}^{(m)} \leq \varepsilon^{(m)} \leq tol_{\max}^{(m)}, \tag{8}$$

where the error indicator $\varepsilon^{(m)}$ is determined by numerical approximations to the following quantities:

$$\begin{aligned}
&h_{il}\left|\tfrac{\partial}{\partial x_i}(\phi S)\right|_i, && \text{for } m = 1,\\
&h_{il}\left|\tfrac{\partial^2}{\partial x_i^2}(\phi S)\right|_i, && \text{for } m = 2,\\
&h_{il}\left|k^{-1}\tfrac{\partial}{\partial x_l}k\right|_i, && \text{for } m = 3,\\
&a_j|u_j|, && \text{for } m = 4,
\end{aligned} \tag{9}$$

and the maximum is taken over $l = 1,\ldots,N_d$. Here $h_{il}$ is the width of a block $i$ in the $l^{\text{th}}$ coordinate direction, $a_j$ the area of a face, and $u_j$ the normal component of the flow stored at face $j$. If the upper or lower bounds of (8) are violated, the block is refined or coarsened, respectively. Here refinement takes precedence over coarsening.

The refinement based on the first and second derivatives of the saturation, is carried out iteratively during a time step. New saturation values are computed, the grid is refined where necessary, new saturations are computed again, etc. The number of iterations does not exceed the maximum number of levels, as refinement can only be carried out to one level finer a time. At the end of the time step, refinement is suppressed and only coarsening is carried out to remove any redundant refinements caused by the previous location of the saturation front.

### Convection

We will concentrate on the numerical solution of (6a). First, we transform this equation to the streamline coordinate $\xi$, defined by

$$d\vec{x} = \frac{\vec{u}}{\phi}d\xi. \tag{10}$$

The transformed equation is

$$\frac{\partial S}{\partial t} + \frac{\partial f(S)}{\partial \xi} = 0. \tag{11}$$

Because the normal components of $\vec{u}$ are given on cell faces and vary linearly between opposing faces, we have $u_k = u_k(x_k)$ as a linear function in $x_k$. The porosity $\phi$ is constant per block. The system (10) decouples in the coordinates and each equation can be solved analytically per block.

For each block, we start at the block center and track backward and forward along the streamline, integrating (10) per block. In this way, a one-dimensional list of saturation values $S_j$ is obtained, as found in each block intersected by the streamline. These values are defined on intervals $(\xi_{j-1/2}, \xi_{j+1/2})$, with each $\xi_{j+1/2}$ corresponding to a block face in the original domain. If the index $j = 0$ is assigned to the interval that is obtained from the block which was chosen as starting point, then the maximum distances that have to be tracked backward and forward are determined by the minimum and maximum values of $\xi$:

$$\begin{aligned}
\xi_{\min} &\leq \xi_{0-1/2} - (t^{n+1} - t^n)\max\left[0, \max_s f'(S)\right],\\
\xi_{\max} &\geq \xi_{0+1/2} - (t^{n+1} - t^n)\min\left[0, \min_s f'(S)\right].
\end{aligned} \tag{12}$$

Given this list of saturation values $S_j^n$, $(j = j_{\min},\ldots,j_{\max})$, at time $t^n$ that are piecewise constant in cells separated by faces $\xi_{j-1/2}$, $(j = j_{\min},\ldots,j_{\max}+1)$, we have to determine a new saturation value $S_0^{n+1}$ in cell $j = 0$ at time $t^{n+1}$. We chose the solution method in Ref. 7, which is based on co-moving coordinates and Roe's approximate Riemann solver[9,10]. (Methods based on co-moving coordinates with a projection step to the original grid are also known as Euler-Lagrange or Lagrangian-Eulerian methods.) In Ref. 7, a system of hyperbolic equations is considered. Our scalar case is much simpler. We extended the method by a different treatment of shocks and expansion fans, to allow for larger time steps and a better resolution of the fans.

Before going over to co-moving coordinates, we review Roe's scheme. Define $\Delta_j h = h_{j+1/2} - h_{j-1/2}$ for any quantity $h$, and let $\Delta t = t^{n+1} - t^n$. An explicit time discretization of (11) leads to

$$\frac{S_j^{n+1} - S_j^n}{\Delta t} = -\frac{\Delta_j f}{\Delta_j \xi}. \tag{13}$$

Here $f_{j+1/2}$ is a numerical flux depending on $S_j$ and $S_{j+1}$. The exact solution of the *Riemann problem*

$$S = \begin{cases} S_j, & \text{for } \xi < \xi_{j+1/2}, \\ S_{j+1}, & \text{for } \xi > \xi_{j+1/2}, \end{cases} \tag{14}$$

at $t = t^n$, yields a unique state $S_{j+1/2} = S(\xi_{j+1/2}, t)$ for $t > t^n$ that can be used to evaluate $f_{j+1/2}=f(S_{j+1/2})$. Roe's scheme uses an *approximate* solution of the Riemann problem by setting

$$f_{j+1/2} = \tfrac{1}{2}\left[f_j + f_{j+1} - |\hat{a}_{j+1/2}|(S_{j+1} - S_j)\right]. \tag{15}$$

Here $f_j = f(S_j)$ and the Roe speed

$$\hat{a}_{j+1/2} = \begin{cases} \frac{f_{j+1}-f_j}{S_{j+1}-S_j}, & \text{if } S_{j+1} \neq S_j, \\ f'(S_j), & \text{if } S_{j+1} = S_j. \end{cases} \tag{16}$$

Eq.(15) can be interpreted as a central difference scheme with an added viscosity term proportional to $|\hat{a}_{j+1/2}|$. Another interpretation is obtained by defining

$$a^+ = \max(0, a), \quad a^- = \min(0, a). \tag{17}$$

Then

$$\Delta_j f = \hat{a}^+_{j-1/2}(S_j - S_{j-1}) + \hat{a}^-_{j+1/2}(S_{j+1} - S_j), \tag{18}$$

which demonstrates the upstream character of the scheme. The two terms on the right-hand side can be viewed as *linear waves*, one moving to the right from $\xi_{j-1/2}$ with speed $\hat{a}^+_{j-1/2}$ and strength $|S_j - S_{j-1}|$, and one moving to the left from $\xi_{j+1/2}$ with speed $\hat{a}^-_{j+1/2}$ and strength $|S_{j+1} - S_j|$. The *nonlinearity* of the problem is accounted for in the computation of the speed $\hat{a}_{j+1/2}$.

The explicit scheme (13) can be rewritten as

$$\begin{aligned} S_j^{n+1} = &\left[1 - \frac{\Delta t}{\Delta_j \xi}(\hat{a}^+_{j-1/2} - \hat{a}^-_{j+1/2})\right] S_j^n \\ &+ \left[\frac{\Delta t}{\Delta_j \xi}\hat{a}^+_{j-1/2}\right] S_{j-1}^n - \left[\frac{\Delta t}{\Delta_j \xi}\hat{a}^-_{j+1/2}\right] S_{j+1}^n. \end{aligned} \tag{19}$$

It is *positive* for

$$\frac{\Delta t}{\Delta_j \xi}(\hat{a}^+_{j-1/2} - \hat{a}^-_{j+1/2}) \leq 1. \tag{20}$$

Positivity implies stability in the maximum norm.

We will not use the method in this form, because of the severe time step restriction (20). Instead, we transform to co-moving coordinates by letting the position $\xi_{j+1/2}$ of each cell face move with a speed $\hat{a}_{j+1/2}$. The transformation to co-moving coordinates $(\zeta, \tau)$ is given by

$$\xi_{j+1/2} = \zeta_{j+1/2} + \int_{t^n}^{\tau} \hat{a}_{j+1/2} d\tau', \quad t = \tau, \tag{21}$$

where $\hat{a}_{j+1/2}$ is computed from $S_j$ and $S_{j+1}$ according to (16). Note that the subscripts now refer to cells in the co-moving frame. The coordinate $\zeta$ refers to the initial position of a face and does not move.

For small $\Delta\tau = \tau - t^n$, the Roe speed in the transformed coordinates approximates $f(S) - sf'(S)$ and is zero. This implies that $S_j$ is constant, as long as the cell size does not vanish. This discrete result is consistent with the well-known fact that *S is constant on characteristics*, but is more general: it holds even in the case of shocks, until the moment that the cell size $\Delta_j\xi = \Delta_j\zeta + \Delta\tau\,\Delta_j\hat{a}$ vanishes. If that happens earlier than time $t^{n+1}$, we integrate up to the instance where the size of the particular cell $j$ vanishes, and then remove this cell from our list of saturations. After this has been done, the Riemann problem between the states $S_{j-1}$ and $S_{j+1}$ of the original list has to be solved. This is done approximately by computing a new speed $\hat{a}$ at the position $\xi_{j-1/2} = \xi_{j+1/2}$ from ($S_{j-1}$ and $S_{j+1}$) by Eq. (16). In this way, the computation can be continued up to an arbitrary large time $t^{n+1}$ in quite an efficient manner.

However, there is one problem which is caused by a fundamental flaw in Roe's approximate Riemann solver. Roe's scheme allows for non-physical expansion shocks. An initial discontinuity that should break up into an expansion fan, remains as it is. A usual remedy[6] is to add a small amount of extra dissipation into (15) by replacing $|\hat{a}_{j+1/2}|$ with $h(\hat{a}_{j+1/2})$, where

$$h(a) = \begin{cases} |a|, & \text{if } |a| \geq \varepsilon_a, \\ (a^2 + \varepsilon_a^2)/2\varepsilon_a, & \text{if } |a| < \varepsilon_a. \end{cases} \tag{22}$$

Here $\varepsilon_a$ is a user-specified constant. In our case, this would imply that $S_j$ is no longer constant, so we chose another remedy.

If an expansion fan occurs between two neighboring states $S_j$ and $S_{j+1}$, extra saturation values are inserted at the face $\xi_{j+1/2}$ using a fixed spacing $\delta S$, which can be user-specified and is typically around 0.01. A new value $S_i$ is inserted *only* if the resulting $\Delta_i\hat{a}$ is *strictly* positive. This excludes saturation values that lead to compression or linear convection with constant speed.

The actual subroutine that performs this insertion is somewhat complicated. To accelerate the process, a precomputed table with saturation values between 0 and 1 at increments of $\delta S$, and their corresponding fluxes and speeds, is used. The method works for any continuously differentiable flux function $f(S)$.

A shortcoming of the approach outlined so far, is its tendency to produce "staircases" after a number of time steps. The non-smoothness is a direct result of the piecewise constant representation of saturation values and is also noted in Ref. 7. The virtual absence of numerical dissipation causes the initial piecewise constant representation to persist, which does not "look nice", but actually is consistent with the first-order accuracy of the scheme. The effect of "staircases" can be reduced by switching to a second-order scheme, using a linear distribution of the data injected into expansion fans. This linear distribution is forced to lie between the cell centers of the original two cells, and preserves mass. Details are given in the Appendix.

Once the above techniques have been used to integrate the one-dimensional data from $t^n$ to $t^{n+1}$, the resulting saturations $S_j^{n+1}$ are projected back to the original cell $(\zeta_{0-1/2}, \zeta_{0+1/2})$ according to

$$\overline{S}_0^{n+1} = \frac{1}{\Delta_0\zeta}\int_{\zeta_{0-1/2}}^{\zeta_{0+1/2}} S(\xi, t^{n+1})\, d\xi, \tag{23}$$

where the subscript 0 now refers to the original coordinate system and $S(\xi, t^{n+1})$ denotes the piecewise constant values $S_j^{n+1}$ in the cells of the co-moving frame. The result $\overline{S}_0^{n+1}$ becomes the new saturation in the original block of the multi-dimensional problem. The method is conservative in one dimension, but not in more than one.

**Diffusion**

The effect of capillary pressure is modeled by diffusion. We discretize the equation

$$\phi\frac{\partial S}{\partial t} = \nabla \cdot \left[D(S, \vec{x})\nabla S\right], \tag{24}$$

in time by

$$\phi\frac{S^{n+1} - \tilde{S}^{n+1}}{\Delta t} = \nabla \cdot \left[D(S^{n+1}, \vec{x})\nabla S^{n+1}\right], \tag{25}$$

where $\tilde{S}^{n+1}$ is the result obtained after convection. In the absence of convection, we simply have $\tilde{S}^{n+1} = S^n$. The spatial part of (24) is discretized by the lowest order mixed finite element method[8]. We obtain a system of the form (4):

$$\begin{aligned} \tfrac{\phi}{\Delta t} S^{n+1} + \quad \nabla \cdot \mathbf{F} \quad &= \tfrac{\phi}{\Delta t}\tilde{S}^{n+1}, \\ \nabla S^{n+1} + \left[D^{n+1}\right]^{-1} \mathbf{F} &= 0. \end{aligned} \tag{26}$$

The nonlinear diffusion term $D^{n+1}$ is treated by using successive substitution (or Picard iteration). The linear system (26) is solved with multigrid. We use $\tilde{S}^{n+1}$ as initial guess for $D^{n+1}$, solve the linear system with multigrid, substitute the resulting $S^{n+1}$ in $D^{n+1}$, and continue to repeat this for a number of times.

### Time step control

The total flow $\vec{u}$ is kept constant during a time step with the convection scheme. Accurate results can be obtained with large time steps if the total flow remains fairly constant in time. This is true if the total mobility varies only mildly with the saturation. For unstable displacement, $\vec{u}$ varies strongly with the saturation and large time steps cannot be taken. The same is true in the presence of gravity, if the gravity forces dominate the viscous forces and act in a different direction. We therefore relate the time step $\Delta t$ to the relative change of the total flow $\vec{u}$ over a time step, in order to keep this change within reasonable bounds.

In the presence of gravity, very small time steps are required if gravity acts in a direction perpendicular to the front. In that case, the total flow changes direction abruptly across the front, and any change in the position of the front results in a large change of the total flow field. Under these circumstances, it is better to use an intermediate value of the total flow field. A predictor-corrector scheme is employed. Given $\vec{u}^{\,n}$ and $S^n$ at time $t^n$, we perform convection and diffusion steps to determine $S^{n+1}$ at time $t^{n+1} = t^n + \Delta t$. Next, the new total flow $\vec{u}^{\,n+1}$ is computed, and the convection and diffusion steps are repeated with $\frac{1}{2}(\vec{u}^{\,n} + \vec{u}^{\,n+1})$. This approach allows us to take substantially larger time steps.

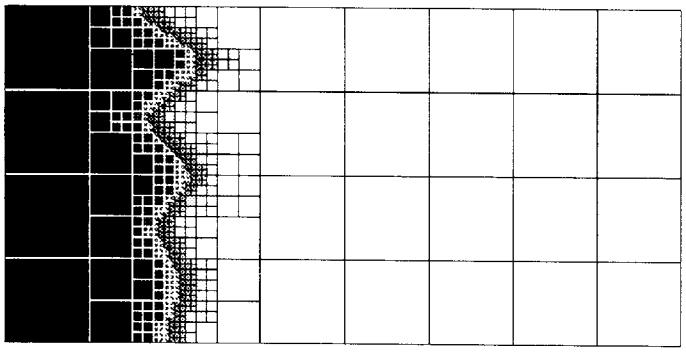

**Figure 1a.** Initial conditions for the unstable displacement problem. Darker gray shades correspond to higher water saturations.

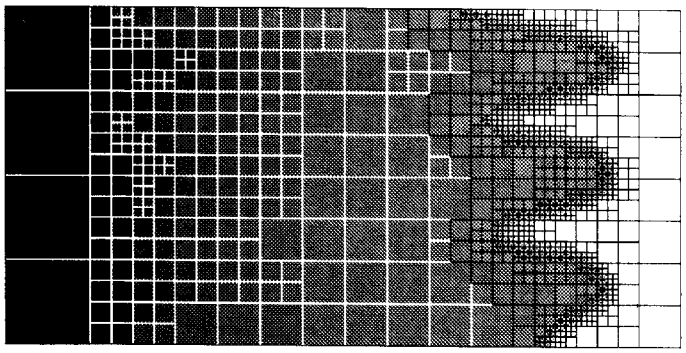

**Figure 1b.** Solution at later time, showing the growth of fingers.

## NUMERICAL EXAMPLES

We present a number of numerical examples to illustrate the capabilities of the method. The fluid, consisting of oil and water, is assumed to be incompressible, as is the rock formation.

### Unstable displacement

A two-dimensional rectangular domain with constant absolute permeability and porosity has inflow of water at the left-hand side and outflow of oil at the right-hand side. Initial conditions are shown in Fig. 1a. We chose an S-shaped fractional flow curve that leads to unstable displacement. The endpoint mobility ratio is 7.262 and the mobility ratio across the shock is 2.051 at a shock saturation $S = 0.565$. Capillary pressure is included. Gravity is absent.

The result after 41 time steps is shown in Fig. 1b, using variable time steps. The amplification of the initial perturbation, caused by the unstable displacement process, is obvious. The capillary pressure suppresses small-scale instabilities (cf. Ref. 4).

### Cusping

A three-dimensional rectangular domain $[0, 4]\times[0, 1]\times[0, \frac{1}{2}]$ has a constant permeability and porosity. Water is injected at the origin of the 15° updip reservoir. Oil is produced at the point $(4, 0, \frac{1}{2})$. The computation includes gravity and capillary pressure. Figures 2a and 2b show the result at 0.25 PVI after 4 time steps for unstable displacement with the same fractional flow curve as in the first example. The result for stable displacement is shown in Fig. 3. The fractional flow curve used here has an endpoint mobility ratio of 2, whereas this ratio across the shock is 0.845 for a shock saturation $S = 0.578$.

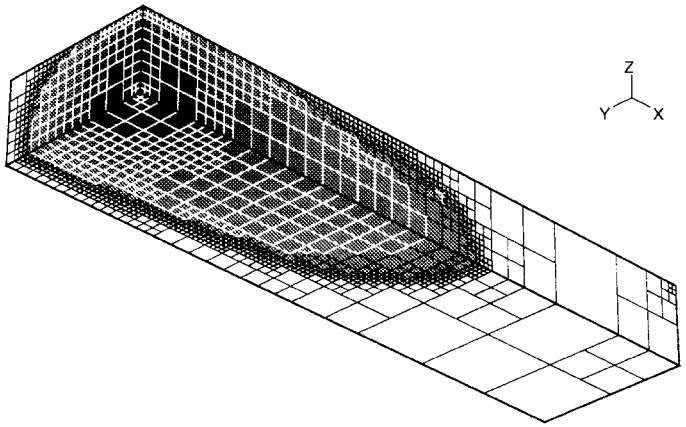


**Figure 2a.** Unstable displacement of oil by water in a 15° updip reservoir, viewed from below.

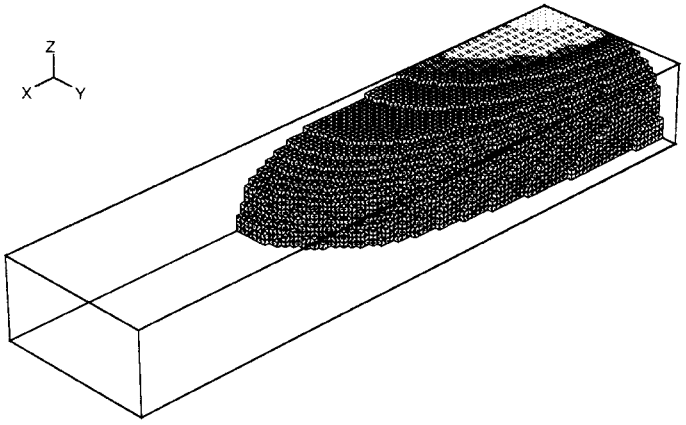


**Figure 2b.** As Fig. 2a, but viewed from the opposite direction. Only saturation values larger than 0.2 are shown.

## DISCUSSION AND CONCLUSIONS

A numerical method has been presented for accurately solving the problem of two-phase, immiscible flow through porous media. The method is based on truly adaptive local grid refinement and operator splitting. A mixed finite element discretization and a multigrid method are used to solve elliptic problems. Nonlinear hyperbolic problems are treated by the method of characteristics and approximate Riemann solvers. The method can be used for a substantial class of flow problems and has been

successfully applied to stable and unstable displacement in two and three dimensions.

The nonlinear characteristic method introduces almost no numerical diffusion and can be used with large time steps. However, the method is not conservative. The conservation error is of the same order of magnitude as the discretization error, which is typically proportional to the cell width divided by the square of the time step. Therefore, it becomes negligibly small if large time steps can be taken. If, on the other hand, the time step needs to be small, of the same order of magnitude as the time step dictated by the stability condition for an explicit scheme, the use of an explicit upstream scheme for the hyperbolic part of the equations becomes more attractive. This situation will occur, for instance, if gravity forces are dominant over viscous forces. In that case, the total flow field is strongly coupled to the position of the front and small time steps are required to limit the size of the numerical errors due to the operator splitting. Note that the global conservation error can be monitored as the computation proceeds. In the numerical experiments presented here, its value never exceeded one percent of the injection rate.

If gravity forces are small compared to viscous forces and the mobility ratio is not extremely large, the numerical method described in this paper easily out-performs traditional methods, both in terms of accuracy and efficiency.

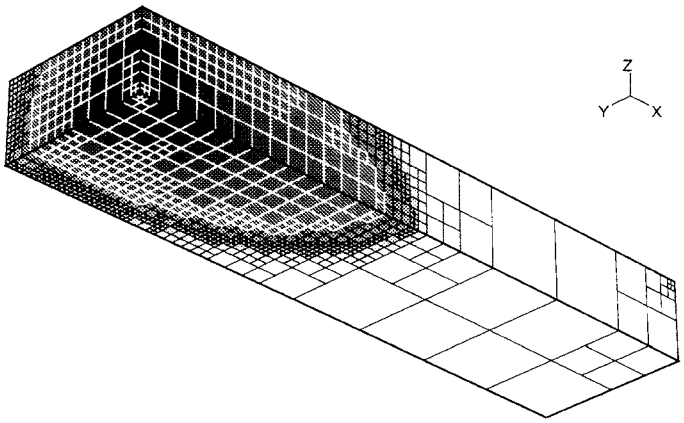


**Figure 3a.** Stable displacement of oil by water in a 15° updip reservoir, viewed from below.

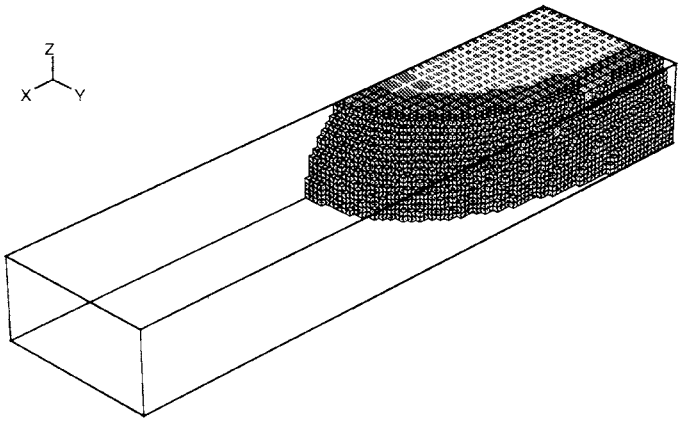


**Figure 3b.** As Fig. 3a, but viewed from the opposite direction. Only saturation values larger than 0.2 are shown.

## NOMENCLATURE

| | |
|---|---|
| $C(S,\vec{x})$ | fluid composition at the wells |
| $D(S,\vec{x}) = -f_g(S)\frac{dP_c}{dS}k$ | diffusion coefficient |
| $f(S) = \lambda_w/\lambda_t$ | fractional flow curve |
| $f_g(S) = \lambda_o\lambda_w/\lambda_t$ | |
| $\vec{g}$ | gravity acceleration vector |
| $k(\vec{x})$ | absolute permeability tensor |
| $k_{ro}(S), k_{rw}(S)$ | relative permeability of oil, water |
| $P_c(S)$ | capillary pressure |
| $p$ | total pressure |
| $q(\vec{x},t)$ | source term, nonzero at wells |
| $S_w = S$ | reduced water saturation |
| $S_o = 1 - S$ | reduced oil saturation |
| $t$ | time, $t \in [0,T]$ |
| $\vec{u}$ | total flow |
| $\vec{v} = (\rho_w - \rho_o)k\vec{g}$ | |
| $\vec{x}$ | position vector, $\vec{x} \in \Omega$ |
| $\lambda_o(S) = k_{ro}/\mu_o$ | relative mobility of oil |
| $\lambda_w(S) = k_{rw}/\mu_w$ | relative mobility of water |
| $\lambda_t(S) = \lambda_o + \lambda_w$ | total mobility |
| $\mu_o, \mu_w$ | viscosity of oil, water |
| $\rho_o, \rho_w$ | density of oil, water |
| $\bar{\rho}(S) = (\rho_o\lambda_o + \rho_w\lambda_w)/\lambda_t$ | average fluid density |
| $\phi(\vec{x})$ | porosity |
| $\Omega$, $\partial\Omega$ | reservoir domain and boundary |


## ACKNOWLEDGEMENTS

The authors are indebted to the management of Shell Internationale Research Maatschappij BV for permission to publish this paper. They thank Geert Schmidt for providing the multigrid and adaptive gridding routines, and for many stimulating discussions.


## REFERENCES


1. Chavent, G., Cohen, G., and Jaffre, J.: "Discontinuous Upwinding and Mixed Finite Elements for Two-Phase Flows in Reservoir Simulation," *Computer Methods in Appl. Mechanics and Eng.* (1984) **47**, 93–118.
2. Ewing, R. E.: "Problems Arising in the Modeling of Processes for Hydrocarbon Recovery," in *The Mathematics of Reservoir Simulation*, R. E. Ewing (ed.), Frontiers in Applied Mathematics, SIAM, Philadelphia (1983) 3–34.
3. Ewing, R. E.: "Adaptive Grid-Refinement Techniques For Treating Singularities, Heterogeneities, and Dispersion," in *Numerical Simulation in Oil Recovery*, M. F. Wheeler (ed.), Springer-Verlag, New York/Berlin (1988) 133–148.
4. Gmelig Meyling, R. H. J., Mulder, W. A., and Schmidt, G. H.: "Porous Media Flow on Locally Refined Grids," in *Proceedings of the Workshop on Numerical Methods for the Simulation of Multi-Phase and Complex Flow*, Amsterdam, May–June 1990, T. Verheggen (ed.), Springer-Verlag, New York/Berlin, in press.
5. Gmelig Meyling, R. H. J.: "Numerical Methods for Solving the Nonlinear Hyperbolic Equations of Porous Media Flow," in *Proceedings of the Third International Conference on Hyperbolic Problems*, Uppsala, June 1990, B. Engquist and B. Gustafsson (eds.), Studentlitteratur, Lund/Chartwell-Bratt Ltd, Bromley (1991) Vol. 1, 503–517.
6. Harten, A.: "High Resolution Schemes for Hyperbolic Conservation Laws," *J. Comput. Phys.* (1983) **49**, 357–393.
7. Harten, A., and Hyman, J. M.: " Self Adjusting Grid Methods for One-Dimensional Hyperbolic Conservation Laws," *J. Comput. Phys.* (1983) **50**, 235–269.
8. Raviart, P. A., and Thomas, J. M.: "A Mixed Finite Element Method for Second Order Elliptic Problems," in *Mathematical Aspects of Finite Element Methods*, Lecture Notes in Mathematics Vol. 606, Springer-Verlag, New York/Berlin (1977) 292–302.
9. Roe, P. L.: "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *J. Comput. Phys.* (1981) **43**, 357–372.
10. Roe, P. L.: "Characteristic-Based Schemes for the Euler Equations," *Ann. Rev. Fluid Mech.* (1986) **18**, 337–365.
11. Schmidt, G. H., and Jacobs, F. J.: "Adaptive Local Grid Refinement and Multi-Grid in Numerical Reservoir Simulation," *J. Comput. Phys.* (1988) **77**, 140–165.

12. Schmidt, G. H.: "A Dynamic Grid Generator and Multi-Grid for Numerical Fluid Dynamics," in *Proceedings of the $8^{th}$ GAMM Conference on Numerical Methods in Fluid Mechanics*, P. Wesseling (ed.), Notes in Numerical Fluid Mechanics Vol. 29, Vieweg, Braunschweig (1990) 493–502.

## APPENDIX. INSERTION IN EXPANSION FANS

To avoid "staircases" in the solution of the one-dimensional scalar hyperbolic equation, extra points inserted at the face between two cells $j$ and $j+1$ are spread over a finite interval between the two cell centers. Here we describe this procedure.

Suppose that $N$ new points have been injected at $\xi_{j+1/2}$, the face between cells $j$ and $j+1$. New index values $i = 0, \ldots, N+1$ are introduced, with $S_{i=0} = S_j$ and $S_{i=N+1} = S_{j+1}$. Let $\xi_L = \frac{1}{2}(\xi_{j-1/2} + \xi_{j+1/2})$ be the center of the original cell $j$, and $\xi_R = \frac{1}{2}(\xi_{j+1/2} + \xi_{j+3/2})$ the center of $j+1$. The new cells $i = 1, \ldots, N$ must have sizes $\Delta_i\xi$ such that

$$\begin{aligned} &\Delta_i\xi = \alpha(S_{i+1} - S_{i-1}), \quad i = 1, \ldots, N, \\ &\sum_{i=0}^{N+1} \Delta_i\xi = \xi_R - \xi_L \equiv L, \\ &\sum_{i=0}^{N+1} S_i\Delta_i\xi = S_0(\xi_{j+1/2} - \xi_L) + S_{N+1}(\xi_R - \xi_{j+1/2}) \equiv M. \end{aligned} \tag{A-1}$$

The first equation describes the linear distribution of saturations in terms of $\xi$, the second places this distribution between $\xi_L$ and $\xi_R$, the original cell centers, and the third imposes conservation. Two remaining quantities in the above are

$$\begin{aligned} \Delta_0\xi &\equiv \xi_{1/2} - \xi_L \equiv h_0, \\ \Delta_{N+1}\xi &\equiv \xi_R - \xi_{N+1/2} \equiv h_{N+1}. \end{aligned} \tag{A-2}$$

The system (A-1) has no unique solution, and $\alpha = 0$ is always a solution. Using the definitions

$$\begin{aligned} &F \equiv \frac{S_N S_{N+1} - S_0 S_1}{\sum_{i=1}^{N}(S_{i+1} - S_{i-1})}, \quad G \equiv M - FL, \\ &F_0 \equiv S_0 - F, \quad F_{N+1} = S_{N+1} - F, \end{aligned} \tag{A-3}$$

the system (A-1) reduces to

$$F_0 h_0 + F_{N+1} h_{N+1} = G. \tag{A-4}$$

We pick the solution that minimizes $h_0$ and $h_{N+1}$ subject to $h_0 \geq 0$ and $h_{N+1} \geq 0$. Without these constraints, the solution is

$$h_0 = \frac{GF_0}{F_0^2 + F_{N+1}^2}, \quad h_{N+1} = \frac{GF_{N+1}}{F_0^2 + F_{N+1}^2}. \tag{A-5}$$

If this violates the constraints, either $h_0 = 0$ or $h_{N+1} = 0$ is substituted into (A-4) to find the corresponding $h_{N+1}$ or $h_0$, respectively. If the constraints are still violated, we set $\alpha = 0$. Otherwise, we let

$$\alpha = \frac{L - h_0 - h_{N+1}}{\sum_{i=1}^{N}(S_{i+1} - S_{i-1})}, \tag{A-6}$$

and compute the new $\xi_{i+1/2}$. The entire procedure is suppressed (implying $\alpha = 0$) if the fractional flow curve $f(S)$ gives rise to a shock between $S_j$ and $S_{j+1}$, or if either state is close to 0 or 1 (within $\frac{1}{2}\delta S$).

**Authors**

The authors are research mathematicians in the General Research Department of the Royal Dutch/Shell Exploration and Production Laboratory at Rijswijk, The Netherlands. **W. A. Mulder** holds a PhD degree in astronomy from Leiden University, the Netherlands. Before joining Shell in 1989, he worked for two years as a post-doctoral fellow in the Numerical Analysis group of the Computer Science Department at Stanford University, followed by two years as assistant professor of computational and applied mathematics at UCLA. **R. H. J. Gmelig Meyling** holds a PhD degree in mathematics from the University of Amsterdam. He is involved with numerical simulation of hydrocarbon recovery processes and electromagnetic exploration. Before joining Shell, he was an assistant professor at Twente University and worked for the Physics and Electronics Laboratory FEL-TNO in the Netherlands.