A novel numerical approach for the solution of the problem of two-phase, immiscible flow in porous media: Application to LNAPL and DNAPL

Amgad Salama, Shuyu Sun, and Mohamed F. El Amin

Citation: AIP Conference Proceedings **1453**, 135 (2012); doi: 10.1063/1.4711165

View online: http://dx.doi.org/10.1063/1.4711165

View Table of Contents: http://scitation.aip.org/content/aip/proceeding/aipcp/1453?ver=pdfcov

Published by the AIP Publishing

A Novel Numerical Approach for the Solution of the Problem of Two-Phase, Immiscible Flow in Porous Media: Application to LNAPL and DNAPL

Amgad Salama*[♦], Shuyu Sun* and Mohamed F. El Amin*

*Computational Transport Phenomena Laboratory (CTPL), Division of Physical Sciences and Engineering (PSE),
King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, KSA
[♦]Nuclear Research Center, AEA, Cairo, Egypt

Abstract. The flow of two immiscible fluids in porous media is ubiquitous particularly in petroleum exploration and extraction. The displacement of one fluid by another immiscible with it represents a very important aspect in what is called enhanced oil recovery. Another example is related to the long-term sequestration of carbon dioxide, CO_2, in deep geologic formations. In this technique, supercritical CO_2 is introduced into deep saline aquifer where it displaces the hosting fluid. Furthermore, very important classes of contaminants that are very slightly soluble in water and represent a huge concern if they get introduced to groundwater could basically be assumed immiscible. These are called light non-aqueous phase liquids (LNAPL) and dense non-aqueous phase liquids (DNAPL). All these applications necessitate that efficient algorithms be developed for the numerical solution of these problems. In this work we introduce the use of shifting matrices to numerically solving the problem of two-phase immiscible flows in the subsurface. We implement the cell-center finite difference method which discretizes the governing set of partial differential equations in conservative manner. Unlike traditional solution methodologies, which are based on performing the discretization on a generic cell and solve for all the cells within a loop, in this technique, the cell center information for all the cells are obtained all at once without loops using matrix oriented operations. This technique is significantly faster than the traditional looping algorithms, particularly for larger systems when coding using languages that require repeating interpretation each time a loop is called like MatLab, Python and the like. We apply this technique to the transport of LNAPL and DNAPL into a rectangular domain.

Keywords: Shifting matrix technique, DNAPL, LNAPL, Porous media, Cell-centered finite difference.
PACS: 02.60.Cb, 02.70.Bf

INTRODUCTION

Dense non-aqueous phase liquids (DNAPLs) represent a particular class of soil and groundwater contaminant that exist as a separate liquid phase in the presence of water. These contaminants, such as coal tar, chlorinated solvents and polychlorinated biphenyl oils, are only slightly soluble in water, and therefore can persist for many decades or even hundreds of years. Some DNAPLs are highly toxic and even very low concentrations in groundwater or the atmosphere can pose an unacceptable risk to human health or the environment. For example, EPA standard for tetrachloroethylene, called a maximum contaminant level (MCL), is set at 0.005 mg/L or 5 ppb. The fact that DNAPLs are denser than water allows them to migrate to substantial depths below the water table in both unconsolidated deposits and fractured bedrock. Unlike DNAPLs which is denser than water, light non-aqueous phase liquids (LNAPLs) such as petrol, heating oil and supercritical carbon dioxide are less dense than water. Of particular interest is supercritical carbon dioxide because of its potential use in enhanced oil recovery to displace more crude oil as well as its promising sequestration capacity in the subsurface. That is amongst the promising tools for reducing global warming, sequestration of carbon dioxide in deep geologic formations remains, by far, the most attractive one. Several reasons may be invoked to explain this conclusion including the larger repository capacity of typical geologic formations, the relatively safer sequestration for longer period of time, the possible geochemical processes that could result in CO_2 to partition between different phases, the availability of technology, etc. Three types of geologic systems have been proposed for CO_2 storage in the subsurface (Holloway, [1]). These include: depleted oil or gas reservoirs, unmineable coal beds and saline aquifers. Deep saline aquifers are geologic layers of permeable rock located 1 to 3 km below the surface and is saturated with salty groundwater. They offer the

Porous Media and Its Applications in Science, Engineering, and Industry
AIP Conf. Proc. 1453, 135-140 (2012); doi: 10.1063/1.4711165
© 2012 American Institute of Physics 978-0-7354-1053-4/$30.00

advantages of having more volume capacity and being the more abundant in the subsurface compared to the other systems. They are usually bordered on the top by a region of much less permeable caprock that help maintain CO_2 for extended period of time. CO_2 is expected to be injected into the subsurface as a dense phase under supercritical conditions in order to occupy less volume. At reservoir conditions, CO_2 is less dense than the saturating groundwater and will migrate upward due to buoyancy. When CO_2 plume reaches the caprock it spreads laterally and is then said to be structurally trapped, Fig.1. While trapped, several processes may take place including dissolution of CO_2 in the brine, mineral trapping, representing CO_2 that has been incorporated into minerals due to chemical precipitation, etc. Residence times of CO_2 in the gas phase is planned to be several thousand years allowing for nearly complete dissolution (Lindeberg and Bergmo, [2]).

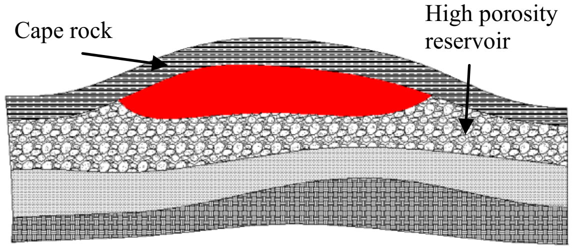

FIGURE 1. Schematic diagram of typical site configuration for CO_2 sequestration

The dissolved CO_2 is slightly denser than the original brine; which induces gravity-driven flows, flow instability, and convective circulation (Weir et al. [3]). Mineral trapping, on the other hand, offers a larger sequestration potential (Bachu et al., [4]) with residence times on the order of geologic times. Simulations of these systems, in the recent few years, have been diverse with respect to emphasis, setups, and numerical schemes. In terms of emphasis, some works have been concentrating on fluid flow neglecting both dissolution and mineral trapping, Audigane et al., [5]. Others have been focusing on modeling geochemical processes with respect to the characterization of mineral trapping only. Some other works have been conducted on the long-term behavior of the trapped plume with respect to dissolution mechanisms and the associated gravity-induced instabilities. Less work, however, have been done on reactive transport modeling, corresponding to simulations that couple structural, dissolution and mineral trapping processes. In terms of setups, several configurations have been examined including homogeneous and layered systems as well as real systems. In terms of numerical schemes, several methods have been applied including the finite difference method, mixed finite element method, discontinuous Galerkin method, etc. as well as several commercial codes. These include ECLIPSE (Lindeberg and Bergmo, [6], Torp and Gale, [7], and Zhou et al., [8]), TOUGH (Pruess, [9], [10], [11]), TOUGH2 (Weir et al., [3], Pruess and Garcia, [12], Oldenburg and Benson, [13]), etc. In all these simulation methodologies the governing set of equations are discretized on a generic cell and the solution is obtained for all other cells in a loop. Calling and updating these loops would require more time particularly for programming languages that require repeating interpretation each time a loop is called, like Matlab, Python and the like. In this work, we consider the hydrodynamic simulations of CO_2 injection into deep saline aquifer using shifting matrix technique. Shifting matrix technique has been introduced to simulating the problem of two-phase, immiscible flows in porous media with application to CO_2 sequestration by Salama et al. [14]. In this work we apply this technique to the problem of NAPL and DNAPL migration in the subsurface.

FORMULATIONS

Two phase immiscible flows in porous media is, generally, described in the framework of the continuum hypothesis in which field variables like pressure, velocity, saturation, etc. are continuous functions of space and time. This point of view may be adapted provided certain conditions and length scale constraints are satisfied, Salama and Van Geel [15] and [16]. The governing equations for this problem are basically the conservation of mass of both phases as well as the Darcy's equations. These may be written as

- Mass conservation equation for both phases

$$(\partial \emptyset S_\alpha)/\partial t+\nabla\cdot\mathbf{v}_\alpha=q_\alpha \tag{1}$$
$$(\partial \emptyset S_\beta)/\partial t+\nabla\cdot\mathbf{v}_\beta=q_\beta \tag{2}$$

- Darcy's law

$$\mathbf{v}_\alpha=-K\ k_{r\alpha}/\mu_\alpha\nabla(p_\alpha-\rho_\alpha gz) \tag{3}$$
$$\mathbf{v}_\beta=-K\ k_{r\beta}/\mu_\beta\nabla(p_\beta-\rho_\beta gz) \tag{4}$$

- Constraints relations

$$S_\alpha+S_\beta=1 \tag{5}$$
$$p_c=p_\alpha-p_\beta \tag{6}$$

Adding Equations (1) and (2) and using Equation (5), one obtains

$$\nabla\cdot(\mathbf{v}_\alpha+\mathbf{v}_\beta)=q_\alpha+q_\beta \tag{7}$$

Introducing $\mathbf{v}_t=\mathbf{v}_\alpha+\mathbf{v}_\beta$ and $q_t=q_\alpha+q_\beta$, Equation (6) reduces to

$$\nabla\cdot\mathbf{v}_t=q_t \tag{8}$$

Here K is the permeability, p is the pressure, $\mathbf{v}$ is the phase velocity, k_r is the relative permeability, μ viscosity and ρ is the density. Let $\lambda_\alpha=k_{r\alpha}/\mu_\alpha$ and $\lambda_\beta=k_{r\beta}/\mu_\beta$ be the mobility of the α and the β phases, respectively. The total mobility is, thus, defined as:

$$\lambda_t=\lambda_\alpha+\lambda_\beta \tag{9}$$

We define the fractional flow of the two phases as

$$f_\alpha=\lambda_\alpha/\lambda_t \tag{10}$$
$$f_\beta=\lambda_\beta/\lambda_t \tag{11}$$

Therefore, Equations (3) and (4) take the form

$$\mathbf{v}_\alpha=-Kf_\alpha\lambda_t\nabla(p_\alpha-\rho_\alpha gz) \tag{12}$$

$$\mathbf{v}_\beta=-Kf_\beta\lambda_t\nabla(p_\beta-\rho_\beta gz) \tag{13}$$

The total velocity is, therefore given as

$$\mathbf{v}_t=-K\lambda_t\nabla p_\alpha+Kf_\beta\lambda_t\nabla p_c+K\lambda_t(f_\alpha\rho_\alpha+f_\beta\rho_\beta)\nabla z \tag{14}$$

Using Equations (6), (12) and (13) Equation (8) becomes

$$\nabla\cdot[-K\lambda_t\nabla p_\alpha+Kf_\beta\lambda_t\nabla p_c+K\lambda_t(f_\alpha\rho_\alpha+f_\beta\rho_\beta)\nabla z]=q_t \tag{15}$$

$$\nabla\cdot K\lambda_t\nabla p_\alpha=\nabla\cdot(Kf_\beta\,\lambda_t\,\nabla p_c)+\nabla\cdot[K\lambda_t\,(f_\alpha\,\rho_\alpha+f_\beta\rho_\beta)\nabla \mathrm{z}]-q_\mathrm{t} \tag{16}$$

If we choose our coordinate system such that the gravity vector is aligned with the z-direction, the above equation reduces to:

$$\nabla\cdot K\lambda_t\nabla p_\alpha=\nabla\cdot(Kf_\beta\lambda_t\nabla \mathrm{p}_\mathrm{c})+g_z\partial/\partial z[K\lambda_t(f_\alpha\,\rho_\alpha+f_\beta\rho_\beta)]-q_t \tag{17}$$

Furthermore, using Equation (14), Equation (1) may be reduced to

$$(\partial\varnothing S_\alpha)/\partial t+\nabla\cdot f_\alpha\mathbf{v}_t=\nabla\cdot[(Kf_\alpha f_\beta\lambda_t\nabla \mathrm{p}_c)+Kf_\alpha f_\beta\lambda_t\,(\rho_\beta-\rho_\alpha\,)\nabla z]+q_\alpha \tag{18}$$

This is simplified to:

$$(\partial\varnothing S_\alpha)/\partial t+\nabla\cdot f_\alpha\mathbf{v}_t=\nabla\cdot(Kf_\alpha f_\beta\lambda_t\nabla \mathrm{p}_c)$$
$$+g\partial/\partial \mathrm{z}[Kf_\alpha f_\beta\lambda_t(\rho_\beta-\rho_\alpha)]+q_\alpha \tag{19}$$

Equations (17) and (19) together with the equations that relates both capillary pressure and relative permeability with saturation and the appropriate boundary conditions represent a closed system which may be solved to obtain saturation of the α-phase with time.

SOLUTION ALGORITHM

The above system of equations is solved using the Implicit Pressure, Explicit Saturation (IMPES) scheme. In this technique the pressure field at the new time step is obtained using the saturation from the previous time step. That is solving Equation (17), given all the parameters based on the saturation of the previous time step, the pressure field of the α-phase may be obtained. The total velocity may then be calculated using Equation (14) and finally the saturation at the new time step is obtained explicitly solving Equation (19) based on the upwind scheme. In the framework of the cell-centered finite difference method the pressure field is defined at the center of the cells while the velocities are defined at the center of the faces. The traditional solution methodology is to discretize the above mentioned equations on a generic cell and obtain the solution for all the cells within loops. In programming languages that repeatedly interpret the instructions in loop operations (e.g., Matlab, Python), exhaustively longer time is required particularly for larger systems, which limits the usefulness of such algorithms. Recently, Sun et al. [17], introduced a new technique, which replaces spatial loops with matrix oriented operations. This technique enhances very much the performance of the code and significantly reduces the CPU time so that it is very much comparable with languages like FORTRAN, C, etc. In this work we use this technique to solve the problem of two-phase immiscible flows in porous media. We follow Salama et al. [14], who used this technique to investigate CO_2 sequestration in the subsurface. We consider the migration of LNAPL and DNAPL in the subsurface.

Traditional Algorithm

Using the cell-centered finite difference scheme the difference equations are usually defined for a generic cell, Fig. 2, and the coefficients are obtained for all the cells within loops. As an example to the traditional cell-center finite difference scheme, we apply it to the various terms of Equation (17) on a generic cell as follows:

The left hand side term: $\nabla\cdot K\lambda_t\nabla p_\alpha=$

$$\frac{K_{xx}\lambda_t(i+1,j+\frac{1}{2})\dfrac{p_\alpha(i+\frac{3}{2},j+\frac{1}{2})-p_\alpha(i+\frac{1}{2},j+\frac{1}{2})}{x(i+\frac{3}{2})-x(i+\frac{1}{2})}}{x(i+1)-x(i)}+$$

$$\frac{K_{xx}\lambda_t(i,j+\frac{1}{2})\dfrac{p_\alpha(i+\frac{1}{2},j+\frac{1}{2})-p_\alpha(i-\frac{1}{2},j+\frac{1}{2})}{x(i+\frac{1}{2})-x(i-\frac{1}{2})}}{x(i+1)-x(i)}+$$

$$\frac{K_{zz}\lambda_t(i+\frac{1}{2},j+1)\dfrac{p_\alpha(i+\frac{1}{2},j+\frac{3}{2})-p_\alpha(i+\frac{1}{2},j+\frac{1}{2})}{z(j+\frac{3}{2})-z(j+\frac{1}{2})}}{z(j+1)-z(j)}+$$

$$\frac{K_{zz}\lambda_t(i+\frac{1}{2},j)\dfrac{p_\alpha(i+\frac{1}{2},j+\frac{1}{2})-p_\alpha(i+\frac{1}{2},j-\frac{1}{2})}{z(j+\frac{1}{2})-z(j-\frac{1}{2})}}{z(j+1)-z(j)} \quad (20)$$

Similarly, the first term in the right hand side, $\nabla\cdot(Kf_\beta\lambda_t\nabla p_c)$, is discretized as given in the previous term. And the second term in the left hand side is discretized as:

$$g_z\partial/\partial z[K\lambda_t\,(f_\alpha\,\rho_\alpha+f_\beta\,\rho_\beta)]=$$

$$\frac{g_z k_{zz}\lambda_t(f_\alpha\rho_\alpha+f_\beta\rho_\beta)(i+1/2,j+1)}{z(j+1)-z(j)}-\frac{g_z k_{zz}\lambda_t(f_\alpha\rho_\alpha+f_\beta\rho_\beta)(i+1/2,j)}{z(j+1)-z(j)} \quad (21)$$

The right hand side terms of Equation (17) are known based on the saturation of the previous time step and using the relations which relate relative permeability and capillary pressure with saturation.

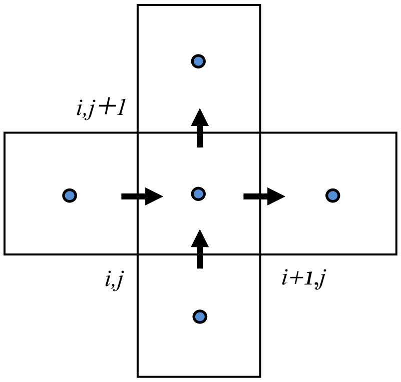

FIGURE 2. Schematic of a generic cell

The coefficients that define the grid and the parameters characteristics to the porous medium domain and the properties of the fluids (e.g., relative permeability, mobility, density, viscosity, etc.) define what is called the matrix of coefficients. This matrix is sparse and diagonal. Figure 3 below shows an illustration of the generated matrix. In traditional programming algorithms, this matrix is populated within a loop.

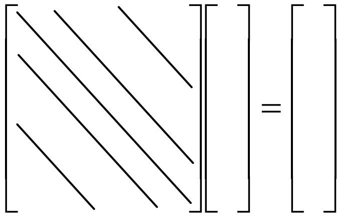

FIGURE 3. Illustration of the reduced matrix system

As indicated earlier, for programming languages which require interpretation of instruction, calling these loops apparently would require longer CPU time, particularly for time-dependent problems in which the elements of the matrix of coefficients require frequent update. It is important, therefore, when coding using these languages, to minimize looping operations and replace them with matrix oriented operations. This is the essence of the shifting matrix technique as will be explained in the next section.

Shifting Matrix Technique

The idea of this technique is to shape the unknown variables (e.g., pressure, velocity, etc.) as column vectors and shift these vectors between cell centers and face centers such that subtracting the newly constructed vectors produces the difference scheme. That is, as illustrated in Fig. 4 below, the cell center information are shifted to the West and East edges. In other words, the cell center vector (*m* elements) produces another vector of size (*m*+*1*).

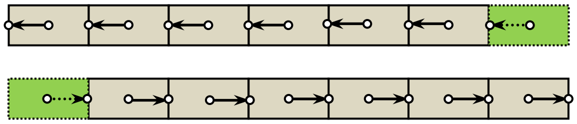

FIGURE 4. Illustration of the shifting process.

As an example, consider a 2D rectangular domain, the mesh constructed for this domain is based on *m* segments in the x-direction and *n* segments in the y-direction. The pressure field is saved in a column vector of size *mn*. Now, shifting this vector to the West and to East edges produces the vectors shown in Fig. 5 below of size (*m*+*1*)*n* by inserting *n* zeros. These zeros will be filled with information at the boundary. Now subtracting these vectors generates the difference in pressure between adjacent cells which is related to the velocity at the edge between two adjacent cells. This is done by introducing a set of shifting matrices as given below.

$$\begin{bmatrix} p(1,1) \\ p(2,1) \\ \vdots \\ p(m,1) \\ p(1,2) \\ \vdots \\ p(m,2) \\ \vdots \\ \vdots \\ p(1,n) \\ p(2,n) \\ \vdots \\ p(m,n) \end{bmatrix} \quad \begin{bmatrix} p(1,1) \\ p(2,1) \\ p(3,1) \\ \vdots \\ p(m,1) \\ 0 \\ p(1,2) \\ \vdots \\ p(m,2) \\ 0 \\ \vdots \\ \vdots \\ p(1,n) \\ p(2,n) \\ \vdots \\ p(m,n) \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ p(1,1) \\ p(2,1) \\ p(3,1) \\ \vdots \\ p(m,1) \\ 0 \\ p(1,2) \\ \vdots \\ p(m,2) \\ \vdots \\ \vdots \\ 0 \\ p(1,n) \\ p(2,n) \\ \vdots \\ p(m,n) \end{bmatrix}$$

FIGURE 5. Illustration of shifting

$A_E \in \{0,1\}_{(m+1)n\times mn}$

$$A_E = \left[\begin{matrix} \begin{pmatrix} 0_{1\times m} \\ I_{m\times m} \end{pmatrix} & & & \\ & \begin{pmatrix} 0_{1\times m} \\ I_{m\times m} \end{pmatrix} & & \\ & & \ddots & \\ & & & \begin{pmatrix} 0_{1\times m} \\ I_{m\times m} \end{pmatrix} \end{matrix}\right]_{(m+1)n\times mn} \tag{22}$$

$A_W \in \{0,1\}_{(m+1)n\times mn}$

$$A_W = \left[\begin{matrix} \begin{pmatrix} I_{m\times m} \\ 0_{1\times m} \end{pmatrix} & & & \\ & \begin{pmatrix} I_{m\times m} \\ 0_{1\times m} \end{pmatrix} & & \\ & & \ddots & \\ & & & \begin{pmatrix} I_{m\times m} \\ 0_{1\times m} \end{pmatrix} \end{matrix}\right]_{(m+1)n\times mn} \tag{23}$$

$A_N \in \{0,1\}_{(m(n+1)\times mn}$

$$A_N = \left[\begin{matrix} \left(0_{m\times mn}\right) \\ I_{mn\times mn} \end{matrix}\right]_{m(n+1)\times mn} \tag{24}$$

$A_S \in \{0,1\}_{(m(n+1)\times mn}$

$$A_S = \left[\begin{matrix} \left(I_{mn\times mn}\right) \\ 0_{m\times mn} \end{matrix}\right]_{m(n+1)\times mn} \tag{25}$$

$B_E \in \{0,1\}_{(m+1)n\times n}$

$$B_E = \left[\begin{matrix} \begin{pmatrix} 0_{m\times 1} \\ I_{1\times 1} \end{pmatrix} & & & \\ & \begin{pmatrix} 0_{m\times 1} \\ I_{1\times 1} \end{pmatrix} & & \\ & & \ddots & \\ & & & \begin{pmatrix} 0_{m\times 1} \\ I_{1\times 1} \end{pmatrix} \end{matrix}\right]_{(m+1)n\times n} \tag{26}$$

$$B_W = \left[\begin{matrix} \begin{pmatrix} I_{1\times 1} \\ 0_{m\times 1} \end{pmatrix} & & & \\ & \begin{pmatrix} I_{1\times 1} \\ 0_{m\times 1} \end{pmatrix} & & \\ & & \ddots & \\ & & & \begin{pmatrix} I_{1\times 1} \\ 0_{m\times 1} \end{pmatrix} \end{matrix}\right]_{(m+1)n\times n} \tag{27}$$

$B_N \in \{0,1\}_{m(n+1)\times m}$

$$B_N = \left[\begin{matrix} \left(0_{mn\times m}\right) \\ I_{m\times m} \end{matrix}\right]_{m(n+1)\times m} \tag{28}$$

$B_S \in \{0,1\}_{(m(n+1)\times m}$

$$B_S = \left[\begin{matrix} \left(I_{m\times m}\right) \\ 0_{mn\times m} \end{matrix}\right]_{m(n+1)\times m} \tag{29}$$

Note that the transpose of these matrices shifts back face centered information to the cell centers. We also define pressure-boundary and velocity-boundary flag vector. These vectors define the kind of boundary conditions assigned to the faces. For example, for pressure boundary conditions, the difference between pressures of adjacent cells in both the x and the y directions is done as follows

$$(A_W - A_E)^T[(A_W - A_E)p + B_E p^E - B_W p^W] \tag{30}$$
$$(A_N - A_S)^T[(A_N - A_S)p + B_S p^S - B_N p^N] \tag{31}$$

where p is the unknown pressure at the center of the cells, p^E, p^W, p^S, and p^N are the pressure at the East, the West, the South and the North boundaries, respectively.

NUMERICAL EXAMPLES

Following Salama et al. 2012, a two-dimensional, square domain (100m×50m) is considered for simulation. Two cases of injection of both NAPL and DNAPL are considered. For the NAPL case, CO_2 at the supercritical state is injected at the bottom West corner of the domain. Because of it is lighter than the hosting saline aquifer it buoyantly rises upwards, Fig. 6. In the DNAPL case the denser fluid is injected at the top left corner and is moving downward because of gravity as shown in Fig. 7.

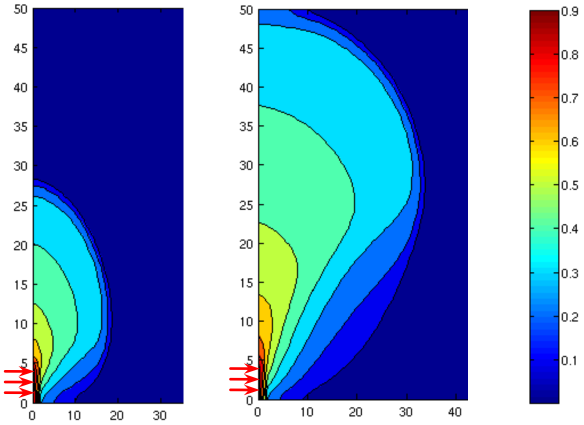

FIGURE 6. Snapshots of LNAPL migration

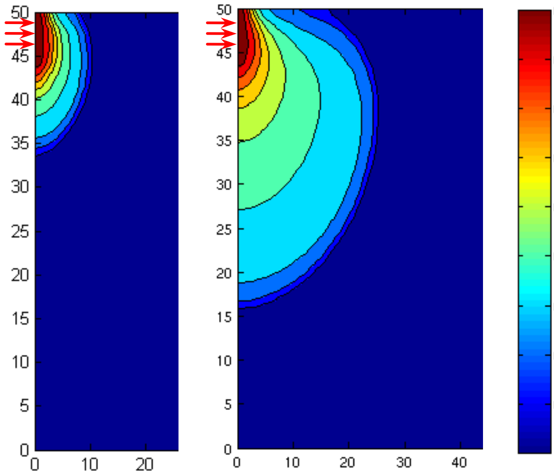

FIGURE 7. Snapshots of DNAPL migration

CONCLUSIONS

Using only matrix operations, the problem of two-phase, immiscible flow in porous media is solved numerically. The shifting matrix technique provides the framework to minimize loop operations which significantly reduce the CPU time when programming using languages like Matlab, Python etc.

REFERENCES

1. S. Holloway, S., Energy Conversion and Management, **38**, 193-198 (1997).
2. E. Lindeberg, and P. Bergmo, The long term fate of CO_2 injected into an aquifer: Proceedings of the 6th International Conference on Greenhouse Gas Control Technology: Kyoto, Japan, 1-4 October 2002.
3. G. Weir, S. P. White, and W. Kissling, Transport in porous media, 23, 37-60 (1996).
4. S. Bachu, W. D. Gunter, and E. H. Perkins, Energy Conversion and Management, **35**, **4**, 269-279, (1994).
5. P. Audigane, I. Gaus, I. Czernichowski-Lauriol, K. Pruess, and T. Xu, American Journal of Science 307, 974-1008, (2007).
6. E. Lindeberg and P. Bergmo, The long term fate of CO_2 injected into an aquifer: Proceedings of the 6th International Conference on Greenhouse Gas Control Technology: Kyoto, Japan, 1-4 October 2002.
7. T.A. Torp, and J. Gale, 2002, Demonstrating storage of CO_2 in geological reservoirs: the Sleipner and SACS projects: Proceedings of the 6th International Conference on Greenhouse Gas Control Technology: Kyoto, Japan, 1-4 October 2002.
8. W. Zhou, M. Stenhouse, R. Arhtur, S. Whittaker, D. Law, R. Chalaturnyk, and W. Jazrawi, The IEA Weyburn CO2 monitoring project modeling of the long term migration of CO2 from Weyburn, in Rubin, E.S., Keith, D.W. and Gilboy, C.F., editors, Proceedings of the 7th International Conference on Greenhouse Gas Control Technologies, Volume 1, 2004.
9. K. Pruess, Tough user's guide: Nuclear Regulatory Commission , report NUREG/CR-4645, 1987.
10. K. Pruess, TOUGH2, a general-purpose numerical simulator for multiphase fluid and heat flow: Lawrence Berkeley Laboratory Report LBL-29400, Berkeley, California, 1991.
11. K. Pruess, Vadose Zone Journal, Special section, Research advances in vadoze zone hydrology through simulations with the TOUGH codes, **3**, 738-746, (2004).
12. K. Pruess and J. García, Environmental Geology, **42**, 282-295, (2002).
13. C. Oldenburg, S. Benson, CO_2 injection for enhanced gas production and carbon sequestration: Proceedings SPE 74367 (2002).
14. A. Salama, S. Sun and M.F. El-Amin, An Efficient IMPES-Based, Shifting Matrix Algorithm To Simulate Two-Phase, Immiscible Flow in Porous Media with Application to CO_2 Sequestration in the Subsurface, Carbon Management Technology Conference, Orlando, 7-9 Feb. 2012, CMTC-150291-PP.
15. A. Salama and P.J. Van-Geel, Journal of Porous Media, **11**, **4**, 403-413, (2008).
16. A. Salama and P.J. Van-Geel, Journal of Porous Media, **11**, **5**, 421-441, (2008).
17. S. Sun, A. Salama, and M.F. El-Amin, J. Computers & Fluids, under review.