Modeling of Multi Phase Flow in Porous Media: Operator Splitting, Front Tracking, Interfacial Area and Network Models

PhD Thesis in Applied Mathematics

Hans Fredrik Nordhaug

Department of Mathematics
University of Bergen



October 2001

Preface

An introductory note together with a collection of papers constitute my thesis presented in partial fulfillment of the requirements for the degree of Doctor Scientarum in applied mathematics at the Department of Mathematics, University of Bergen. The study started formally September 1998 and lasted over three years.

In this work several new numerical methods and a new mathematical model related to multi phase flow in oil reservoirs have been developed, implemented and tested. The numerical difficulties in multi phase flow stem from the highly nonlinear nature of the equations. The solution can develop sharp fronts that travel through the reservoir. We have used front tracking to capture these fronts in our numerical methods. In addition *corrected* operator splitting for systems has been used and developed because of the advection dominated nature of the equations. The new mathematical model was developed to improve short comings of existing models and to include interfacial area as a new independent variable.

This thesis consists of two parts. Part I, the introduction, is a summary of the papers presented in Part II. Some additional results that did not appear in the papers are also included in Part I. Part II consists of the following 5 papers:

Operator splitting methods for systems of convection-diffusion equations: Nonlinear error mechanisms and correction strategies

> This paper presents a modification of corrected operator splitting for scalar equations to systems. The idea is to construct a residual flux after the advection step and use this flux in the diffusion step.

A local streamline Eulerian-Lagrangian method for two-phase flow

> Two dimensional two-phase flow is solved by operator splitting. The advection part is solved by front tracking along streamlines, and the diffusion part by a standard finite difference scheme.

A streamline front tracking method for two- and three-phase flow including capillary forces

> Two dimensional multiphase flow is solved by operator splitting. The advection part is solved by front tracking along streamlines, and the diffusion part by a standard finite difference scheme. The method is compared to a fast marching method and a modified method of characteristics.

Two phase flow including interfacial area as a variable

> In this paper a new model for two phase flow in porous media is presented. Starting out from a model that covers all the physics we make simplifying assumptions. The resulting model includes interfacial area as one of the dependent variables.

A pore network model for calculation of interfacial velocities

> A simple dynamic network model (presented by Blunt & King) is used to calculate saturations, interfacial area, pressures and velocities. Since the results are obtained at a micro scale averaging is done to obtain macroscale variables like pressure and saturations. The averaged results are used to test the equations proposed in the paper above.

In more detail Part I consists of 5 chapters. Chapter 1 gives a short general introduction to multi phase flow in porous media. Some of the main physical and numerical problems investigated in this work are presented here. The subsequent three chapters in Part I give summaries of the papers: Chapter 2 gives a brief description of corrected operator splitting for systems. In Chapter 3 front tracking along stream lines is explained. Finally in Chapter 4 inclusions of interfacial area in two phase flow models and dynamic network models are discussed. Chapter 5 gives a general summary of conclusions and remarks.

The numerical models used in this thesis are written in either C, C++ or Fortran 77 and used under all or some of the following environments: Linux, Unix and Windows. Matlab has been extensively used, but mostly as a visualization tool. The thesis is written in LaTeX and the editor of my choice is Emacs.

Acknowledgments

This work was supported by the Norwegian Research Council (NFR) under grant 116153/431.

At the end of this project there are a lot of people I wish to thank. First of all, I wish to express my sincere gratitude to my supervisor Helge K. Dahle. Without his help this would never have come to an end. I also would like to express my gratitude to my co-supervisors Magne S. Espedal and Kenneth H. Karlsen. They have all contributed with their insights and interesting discussions.

There have been several others involved in this project, but in particular I would like to thank William G. Gray and Michael A. Celia. Without their deep insights some of the main ideas would never have emerged.

The fall 2000 I stayed as a guest researcher at the department of Civil and Environmental Engineering at Princeton University, New Jersey, USA. My advisor there was Michael A. Celia. I am very grateful to him that he made it possible for me to come and that he made it such an interesting stay. I would also like to thank Rudolph Held and Twan Gielen for interesting discussions during this period.

The three years at the department has been a nice time thanks to the colleagues in my group, LIM – Laboratory of Industrial Mathematics, and the students at the sixth floor. No one mentioned, no one forgotten.

And last, but not least, I would like to thank my girlfriend Celia M. Berg.

Hans Fredrik Nordhaug
Bergen, October 2001

Contents

I Introduction **1**

1 Multi phase flow in porous media **3**
1.1 From micro to macro scale 3
1.2 Non-linear equations and systems 4

2 Corrected operator splitting for systems **7**
2.1 Introduction 7
2.2 Operator splitting 8
2.3 Nonlinear error mechanisms 9
2.3.1 The COS strategy 10
2.4 A COS method 11
2.4.1 Convection solver 11
2.4.2 Diffusion solver 12
2.4.3 Construction of the residual flux 12
2.5 Concluding remarks 13

3 Front tracking along streamlines **15**
3.1 Governing equations 15
3.2 Solution strategy 16
3.2.1 A streamline front tracking method 16
3.3 Triangular systems 17
3.3.1 A Riemann solver for triangular systems 18
3.3.2 Approximation of a full three-phase flow system 19
3.4 Concluding remarks 20

4 Interfacial area and dynamic network models **23**
4.1 A generalized two phase flow model 23
4.1.1 Assumptions and governing equations 24
4.1.2 Fractional flow formulation 26
4.2 Dynamic network models and equation testing 26
4.2.1 The dynamic network model 28

4.2.2 Calculating average variables 30
4.2.3 Equation testing 32
4.3 Concluding remarks 34

5 Conclusions **37**

Bibliography **41**

II Published and submitted work **45**

A Operator splitting methods for systems of convection-diffusion equations: Nonlinear error mechanisms and correction strategies **47**

B A local streamline Eulerian-Lagrangian method for two-phase flow **75**

C A streamline front tracking method for two- and three-phase flow including capillary forces **85**

D Two phase flow including interfacial area as a variable **101**

E A pore network model for calculation of interfacial velocities **111**

Part I

Introduction

Chapter 1

Multi phase flow in porous media

In reservoir problems we consider some or all of the following phases: Oil, gas, water and solid. The solid phase is normally assumed to be immobile and non-deforming, but in general this does not need to be the case. By multi phase flow we will mean the flow of oil, gas and water. The phases are categorized according to their different physical quantities. A hydrocarbon phase, may consist of different hydrocarbon components, e.g., the oil phase can contain several oil and gas types. In this work the components are neglected and only the phases are considered. A porous medium is any solid phase, e.g. sand stone, that is permeable. The flow in a porous medium takes place through connected pores in the rock. Regions on a larger scale that contain oil or gas are called reservoirs. The typical size of a reservoir is kilometers in each direction while the pore scale size is millimeters or less. Solving the Navier-Stokes equation at the pore scale to obtain the transport on a larger scale is not numerically feasible because of the huge difference in scales. Therefore, some averaging is necessary to go from the pore scale (micro scale) to the reservoir scale (macro scale). In this process the Navier-Stokes equations are replaced by macro scale equations that are solved for macro scale variables. For a thorough introduction to reservoir modeling see [1].

The papers presented herein cover several topics in multi phase flow in porous media, and they address some central problems both on the micro scale as well as on the macro scale. In addition, operator splitting techniques have been developed for convection dominated non-linear transport equations.

1.1 From micro to macro scale

As noted above it is not numerically feasible to solve the equations on the micro scale to obtain macro scale behavior for the fluid phases. Therefore macro scale equations involving macro scale variables are used. Traditional multi-phase

flow models are usually based on direct extensions of one-phase flow equations like Darcy's law and conservation of mass. This leads to the introduction of constitutive relationships like capillary pressure curves and relative permeabilities. Although such models are generally accepted and have been successfully applied to numerous problems, one can easily list several difficulties. Most important, the models are *ad hoc* and involve hidden assumptions.

A rigorous approach to modeling of multi-phase flow is given by Gray and Hassanizadeh, e.g. [2]. This approach starts from first principles and is aiming at a more complete description of multi-phase flow phenomena. By using localization theorems the authors transform conservation equations for micro scale variables to conservation equations for macro scale variables. These theorems essentially transform averages of derivatives to derivatives of averages. In Paper D we use this approach to get a new formulation for two phase flow including interfacial area. Some of the proposed equations in this work are tested using a dynamical network model in Paper E.

Since geological data currently is available at a much finer scale than any practical computational grid, some kind of upscaling is necessary to get reasonable averaged physical parameters. This kind of parameter upscaling is not the subject of this work, and we refer to Dagan [3] for a thorough description of different scales in porous media flow and averaging techniques. However, we note that as the computers become faster and the methodologies improve, the computational scale may get closer to the geological scale. Some of the methods described in this work are aiming in that direction.

1.2 Non-linear equations and systems

Mathematical models for fluid flow often involve systems of convection-diffusion equations. This is the case for multi phase flow in porous media. These models are normally convection dominated with sharp fronts building up because of the non-linearities in the flow functions. Because of these fronts, difficulties will be experienced when standard numerical approximations are used. Thus, many different methods have been proposed to overcome the difficulties, see [4].

Mathematical models for three phase flow models often consist of a set of pressure–velocity equations and a pair of strongly coupled equations for phase saturations. The set of saturation equations is normally called a 2×2 system. The coupling of these strongly non-linear equations makes them hard to solve. In Paper C (and A), a solution procedure for a simplified 2×2 system is presented. We assume that one of the phases is independent of the other. This will normally be the gas and is due to the fact that the viscosity of the gas phase is usually at least an order less than the viscosities of the liquid phase(s). We call such a system

a triangular system because the Jacobi matrix for the flow functions is upper (or lower) triangular.

An important design principle for many numerical methods for convection-diffusion equations is *operator splitting* (OS). That is, one splits the time evolution into partial steps to separate the effects of convection and diffusion. It has been shown that the temporal splitting error can be significant when there is a shock present in the solution. This is well understood for scalar convection-diffusion equation. Paper A describes the error mechanisms and a correction strategy for systems.

In Paper A, B and C we concentrate on finding accurate and efficient solvers for the advection step in the OS method. To achieve high efficiency, i.e, long time steps, we have chosen to use the front tracking algorithm. The algorithm itself is *super fast* and has no numerical diffusion. However, front tracking relies on accurate Riemann solvers, and for systems in general they are hard to construct. For a certain class of triangular 2×2 systems the Riemann solver is not too complicated to construct, and in Paper C (and A) we have solved such systems by using front tracking.

Chapter 2

Corrected operator splitting for systems

This chapter describes the methods and ideas from Paper A, "*Operator splitting methods for systems of convection-diffusion equations: Nonlinear error mechanisms and correction strategies* ".

Mathematical models for fluid flow often involve systems of convection-diffusion equations as a main ingredient. As mentioned in Section 1.2, operator splitting (OS) is an important tool for developing numerical methods for such systems. In particular, OS methods are often used to solve convection-diffusion problems that are of convection dominated nature, see [5] and the references therein.

2.1 Introduction

The motivation for operator splitting methods is that it is easy to combine efficient methods for solving the convection step with efficient methods for solving the diffusive step. Especially for convection dominated systems, it is a major advantage to be able to use an accurate and efficient hyperbolic solver developed for tracking discontinuous solutions. By combining this with efficient methods for the diffusive step, we get a powerful and efficient numerical method which is well suited for solving parabolic problems with sharp gradients.

The obvious disadvantage of operator splitting methods is the temporal splitting errors. The temporal splitting error in OS methods can be significant in regions containing viscous shocks [6, 7, 8]. Thus, to resolve viscous shock profiles correctly, one must resort to very small splitting steps. This imposes a time step restriction that is not present in the underlying numerical methods for the convective and the diffusive step. To reduce the influence of temporal splitting errors in OS methods, and to allow for the use of large splitting steps, the *corrected oper-*

ator splitting (COS) method was introduced by Espedal and Ewing [9]. A recent mathematical description of this method was given by Karlsen and Risebro [10].

The main idea behind the scalar COS method is to take into account the unphysical entropy loss (due to Oleinik's convexification [11]) produced by the hyperbolic solver in the convective step. The COS approach uses the wave structure from the convective step to identify where the (nonlinear) splitting errors occur.

The purpose of the paper described here is to derive a thorough understanding of the nonlinear mechanisms behind the viscous splitting error typically appearing in operator splitting methods for systems of convection-diffusion equations. This mechanism is well understood in the scalar case.

2.2 Operator splitting

To describe our ideas in more detail, we consider the one dimensional Cauchy problem for $\ell \times \ell$ ($\ell \geq 1$) systems of convection-diffusion equations

$$\partial_t U + \partial_x F(U) = D\,\partial_x^2 U, \qquad U(x,0) = U_0(x) \tag{2.1}$$

where $x \in \mathbb{R}$ and $t > 0$. Here $U = (u_1, \ldots, u_\ell)^{\mathrm{T}}$ is the unknown state vector, $F(U) = (f_1(U), \ldots, f_\ell(U))^{\mathrm{T}}$ is a vector of flux functions for each variable, and $D = \mathrm{diag}(\varepsilon_1, \ldots, \varepsilon_\ell) > 0$ is a constant diagonal matrix. Let $\mathcal{S}_t$ denote the solution operator which takes the initial data $V_0(x)$ to a weak solution at time t of the purely hyperbolic problem

$$\partial_t V + \partial_x F(V) = 0, \qquad V(x,t) = V_0(x). \tag{2.2}$$

We write $V(x,t) = \mathcal{S}_t V_0(x)$ for this weak solution. Next, let $\mathcal{H}_t$ denote the operator which takes the initial data $W_0(x)$ to a weak solution at time t of the purely parabolic problem

$$\partial_t W = D\partial_x^2 W, \qquad W(x,t) = W_0(x). \tag{2.3}$$

We write $W(x,t) = \mathcal{H}_t W_0(x)$ for this solution.

In what follows, we consider a fixed final computing time $T > 0$. For simplicity we also choose a fixed splitting step $\Delta t > 0$ and an integer N_t such that $N_t \Delta t = T$. Then we define the semi-discrete OS algorithm by

$$U_{\Delta t}(\cdot, n\Delta t) := \left[\mathcal{H}_{\Delta t} \circ \mathcal{S}_{\Delta t}\right]^n U_0(\cdot), \qquad n = 0, \ldots, N_t. \tag{2.4}$$

In applications, the exact solution operators $\mathcal{S}_t$ and $\mathcal{H}_t$ in (2.4) are replaced by numerical methods. We will use front tracking as defined by Risebro [12, 13, 14] as an approximate solution operator for the hyperbolic part. For the parabolic part, we here use a simple explicit central difference method.

2.3 Nonlinear error mechanisms

In the introduction we stated that OS approximations can be too diffusive near viscous shocks when the splitting step Δt is large. To explain this in more detail let us first study the *scalar* case:

The entropy condition introduces a local linearization of $f(\cdot)$ once a shock is formed in the convection step. This linearization represents the entropy loss associated with the formation of a shock in the hyperbolic solution. Thus, the evolution of the *hyperbolic* solution is governed locally by some convex/concave envelope f_c of f between the left and right shock values, see Figure 2.1. A similar linearization can be introduced locally for the *parabolic* problem. That is, the flux function f can be decomposed into a convective part f_c and a self-sharpening part $f - f_c$ that tends to counteract the diffusive forces. Loosely speaking, we say that f_c governs the local translation and $f - f_c$ the shape (or structure) of the viscous front. In the OS algorithm, the local residual flux $f - f_c$ is disregarded in the hyperbolic step and the corresponding self-sharpening effects are therefore not taken into account in the splitting, resulting in a splitting error. Figure 2.1 gives an illustration of f, f_c, and the residual flux $f_{\text{res}} := f - f_c$ in the scalar case.

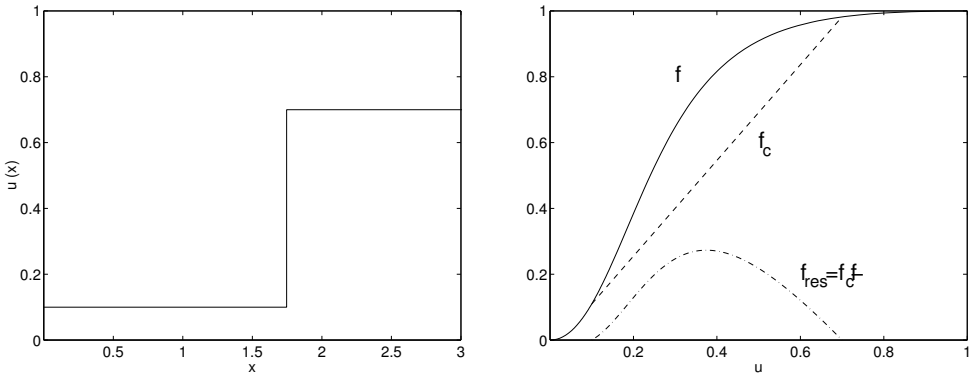


Figure 2.1: (Left) A single shock solution from a convection step. (Right) The corresponding residual flux function; flux function f (solid), convex envelope f_c, i.e., local linearization (dash), and residual flux f_{res} (dash-dot).

For a general system, the error mechanism is quite similar. To study it, we consider the propagation of a single viscous shock. Assume that the splitting step is sufficiently large so that a shock has developed in the hyperbolic substep (2.2), i.e., the solution $V(\cdot, t = \bar{t})$ consists of a single discontinuity at $x = \bar{x}$ with left and right shock values $V^l = (v_1^l, \ldots, v_1^l)^{\mathrm{T}}$ and $V^r = (v_1^r, \ldots, v_1^r)^{\mathrm{T}}$. Then the behavior (forward and backward in time) of $V(x,t)$ locally around $(\bar{x}, \bar{t})$ is governed by the

linearized hyperbolic problem

$$\partial_t V + \partial_x(\bar{\sigma} V) = 0, \qquad V(x,\bar{t}) = \begin{cases} V^l, & \text{for } x < \bar{x}, \\ V^r, & \text{for } x > \bar{x}, \end{cases} \tag{2.5}$$

where $\bar{\sigma}$ is the Rankine–Hugoniot shock speed satisfying

$$F(V^l) - F(V^r) = \bar{\sigma}(V^l - V^r).$$

We *claim* that a large part of the splitting error that occurs locally around $(\bar{x},\bar{t})$ in the standard OS algorithm can be understood in terms of the difference between the nonlinear system (2.1) and the linearized system (2.5) with right-hand side $D\partial_x^2 V$. In other words, in terms of the difference $\partial_x(F(U) - \bar{\sigma}U)$. In (2.1), the diffusion caused by the second order operator is perfectly balanced by the self-sharpening effects due to the nonlinearity in the convective operator. In the OS strategy, this self-sharpening disappears once a shock develops because $F(U)$ is in effect replaced by $\bar{\sigma}U$ locally. Thus, one step in OS effectively amounts to solving $\partial_t U + \partial_x(\bar{\sigma}U) = D\partial_x^2 U$ and not (2.1).

2.3.1 The COS strategy

To compensate for the loss of self-sharpening effects, the *scalar* COS approach proposes to include the residual flux F_{res} in the diffusion step of the splitting. The COS method therefore replaces the purely parabolic split problem (2.3) by

$$\partial_t W + \partial_x F_{\text{res}}(x,W) = D\partial_x^2 W, \qquad W(x,0) = W_0(x). \tag{2.6}$$

where $F_{\text{res}}(x,W)$ is the residual flux in the point x. Letting $\mathcal{P}_t$ denote the solution operator associated with (2.6), the COS solution may then be defined as

$$U_{\Delta t}(\cdot, n\Delta t) := \left[\mathcal{P}_{\Delta t} \circ \mathcal{S}_{\Delta t}\right]^n U_0(\cdot). \tag{2.7}$$

We have replaced the convection-diffusion equation (2.1) by a hyperbolic equation (2.2) and another convection-diffusion equation (2.6), where the flux term in (2.6) is seemingly more complicated than the one in (2.1). However, we see that while F contains convective *and* self-sharpening effects, F_{res} only contributes to self-sharpening effects. Thus, viscous shock fronts are moved to the correct location in the convective step and given a correct shape in the diffusive step.

When applied to systems of parabolic equations, the correction algorithm needs to be reformulated, since one cannot simply write down the solution of the hyperbolic step in terms of convex/concave envelopes. Instead, we identify the following term

$$\partial_x F_{\text{res}}(U) = \partial_x\big(F(U) - \bar{\sigma}U\big), \tag{2.8}$$

for each discontinuity in the solution from the hyperbolic step. Then, the parabolic subproblem (2.3) is modified locally by adding $F_{\text{res}}(U)$, which gives the new split problem (2.6). By integrating (2.8) with respect to x, we get the *residual flux*

$$F_{\text{res}}(U) = \big(F(U) - F(V^l)\big) - \bar{\sigma}\big(U - V^l\big), \tag{2.9}$$

where we have chosen the constant of integration such that

$$F_{\text{res}}(V^l) = F_{\text{res}}(V^r) \equiv 0.$$

2.4 A COS method

The operator splitting methods introduced above result in two different subproblems that each must be solved numerically. The convection part is solved by a front tracking method, and the diffusion part is solved by an explicit central finite difference scheme.

2.4.1 Convection solver

In this section we describe the front tracking method [12, 13, 14] for solving systems of conservation laws (2.2)

$$\partial_t V + \partial_x F(V) = 0, \quad V(x,0) = V_0(x).$$

Front tracking is an algorithm for computing a piecewise constant approximation to $V(x,t)$. The advantage of using a front tracking method is that it directly identifies the correct physical envelope, given that the Riemann solver is correct. We will not discuss Riemann solvers further here, but just note that finding good approximative Riemann solvers for systems is in general a difficult problem. Descriptions of the Riemann solvers used in the applications are given in Section 3 in Paper A.

First, V_0 is approximated by a step function so that a Riemann problem can be associated with each jump in the approximate initial data. The solution of each Riemann problem is approximated by step functions. In the front tracking approximation, rarefaction waves are approximated by step functions sampled along the wave curves (according to a pre-set, user-defined parameter δ), while the rest of the Riemann solution is left intact. By doing this each Riemann problem produces a sequence of jump discontinuities (fronts) that travel with a finite wave speed. The Riemann solution is represented by a list of fronts, sorted according to increasing wave speeds. A global solution (in x) is formed by connecting the local Riemann solutions. The solution consists of constant states separated by space-time rays, i.e., a list of fronts sorted from left to right. There will be a first time

at which two or more space-time rays intersect, i.e., two or more fronts collide. This collision defines a new Riemann problem which is solved and inserted into the list. The algorithm proceeds in this manner from collision to collision. The numerical method is *unconditionally* stable and very fast.

2.4.2 Diffusion solver

The parabolic step is a Cauchy problem of the form

$$\partial_t W + \partial_x G(W) = D\partial_x^2 W, \quad W(x,0) = W_0(x). \tag{2.10}$$

In applications G is the residual flux term, see (2.8), and D is still the diagonal matrix $\mathrm{diag}(\varepsilon_1,\dots,\varepsilon_\varrho) > 0$. To solve this system, one can for instance use the explicit, central finite difference method

$$\frac{W_j^{n+1} - W_j^n}{\tau} - \frac{G(W_{j+1}^n) - G(W_{j-1}^n)}{2\Delta x} = D\frac{W_{j+1}^n - 2W_j^n + W_{j-1}^n}{(\Delta x)^2}. \tag{2.11}$$

This scheme is stable provided the discretization parameters τ and Δx satisfy the following conditions

$$\tau \le 0.5\Delta x^2/\max\varepsilon_i, \qquad \Delta x \max|\lambda_G| \le 2\max\varepsilon_i,$$

where λ_G denotes the eigenvalues of G', the derivative of G. See Strikwerda [15] for an introduction to finite difference schemes and this one in particular. For the Cauchy problem (2.10), which has linear diffusion, convergence and error estimates for this scheme is shown in [16]. The stability conditions above may put severe restrictions on the discretization parameters, especially on Δx for small values of ε. However, both these conditions can be weakened or removed by using a more sophisticated scheme. The use of implicit schemes is discussed in the next section. To keep the technical details at a minimal level, we chose the simple explicit scheme.

2.4.3 Construction of the residual flux

Given a piecewise constant front tracking solution, $U^{n+1/2}$, of the hyperbolic equation (3.4), we can now construct the residual flux $F_{\mathrm{res}}(x,\cdot)$ appearing in (2.6). We assume that the discontinuities of $U^{n+1/2}(x)$ are located at the points $\{x_i\}$. Let $U_i = (u_1^i,\dots,u_\varrho^i)^{\mathrm{T}}$ and $U_{i+1} = (u_1^{i+1},\dots,u_\varrho^{i+1})^{\mathrm{T}}$ denote the values of $U^{n+1/2}(x)$ in the intervals $[x_{i-1},x_i)$ and $[x_i,x_{i+1})$, respectively. Locally, around the ith discontinuity emerging from (x_i,t_0) the nonlinear problem (2.2) is governed by the

linearized problem

$$\partial_t V + \partial_x(\sigma_i V) = 0, \qquad V(x_i, t_0) = \begin{cases} U_i, & \text{for } x < x_i, \\ U_{i+1}, & \text{for } x > x_i, \end{cases}$$

where σ_i is the Rankine–Hugoniot shock speed satisfying $F(U_i) - F(U_{i+1}) = \sigma_i(U_i - U_{i+1})$. Motivated by the discussion in Section 2.3, we define the residual flux F^i_{res} associated with the ith discontinuity, as

$$F^i_{\text{res}}(U) = \begin{cases} \big(F(U) - F(U_i)\big) - \sigma_i\big(U - U_i\big), & U \in (u^i_1, u^{i+1}_1) \times \cdots \times (u^i_\varrho, u^{i+1}_\varrho), \\ 0, & \text{otherwise.} \end{cases}$$

Note that $F^i_{\text{res}}(U_i) = F^i_{\text{res}}(U_{i+1}) \equiv 0$.

Although a residual flux term can be identified for every discontinuity in the front tracking solution, they should not be included for discontinuities which approximate rarefaction waves or for weak shocks. Therefore, in practice, we only include residual terms for shock waves with strength exceeding a user-defined threshold parameter γ. The process of identifying relevant residuals can be made more rigorous, and simplified, by tagging fronts in the front tracker according to wave type (shock/rarefaction/contact).

Having defined the residual fluxes in state space $(u_1, \ldots, u_\varrho)$, we need to specify where to apply them in physical space (i.e., intervals in x). For *explicit* discretizations we apply the following strategy: Observe that in each spatial interval where the solution is monotone in all its components (henceforth called *monotonicity interval*), all residual fluxes are defined on disjoint sets in state space. Therefore, the residual flux is set to zero outside (a subset of) the associated monotonicity interval, i.e.,

$$F_{\text{res}}(x, U) = \sum_i F^i_{\text{res}}(U)\chi_{D_i}(x),$$

where $\chi_I(x)$ denotes the indicator function of the interval $I \subset \mathbb{R}$ and D_i is the (subset of) the monotonicity interval. For implicit discretizations we use a much simpler approach where the user prescribes the length of the intervals where the correction is applied. Unfortunately, specifying a reasonable length for the correction intervals must be based on experience. For further details concerning the construction of the residual flux we refer to Paper A.

2.5 Concluding remarks

In Paper A results for two particular 2×2 systemsdescribing flow in porous media were presented. Simulations for a two-phase multicomponent model, a polymer

system, and for a triangular three-phase flow model were reported. In addition, the polymer system is solved in 2D using dimensional splitting.

We demonstrate numerically that operator splitting (OS) methods for systems of convection-diffusion equations in one space dimension, have a tendency to be too diffusive near viscous shock waves. The idea behind the scalar COS method is to use the wave structure from the convection step to identify where the nonlinear splitting error (or entropy loss) occurs. The potential error is then compensated for in the diffusion step (or in a separate correction step).

Similar to the scalar case, the splitting error is closely related to the local linearizations introduced implicitly in the convection steps due to the use of an entropy condition. A COS method for systems was proposed. The numerical examples demonstrate that the COS method is significantly more accurate than the corresponding OS method when the splitting step is large and the solution consists of (moving) viscous shock waves.

Chapter 3

Solving multi dimensional multi-phase flow by front tracking along streamlines

This chapter describes the results and ideas form Paper B, *"A local streamline Eulerian-Lagrangian method for two-phase flow "*, and Paper C, *"A streamline front tracking method for two- and three-phase flow including capillary forces "*.

Increased demands for assessment of uncertainties and history matching require fast and accurate flow simulations on multiple plausible geological models on a routinely basis. Conventional reservoir simulators fail to fulfill this need, and there seems to be a trend within the petroleum industry to simulate reduced sets of equations. Typically streamline and/or front tracking methods are used to solve the hyperbolic Buckley-Leverett equation for two-phase flow, see [17, 18, 19].

In the work described here we consider models of multi-phase flow which do include capillary forces. We also allow for three phases. In particular we shall investigate a streamline front tracking method. To account for capillary effects we use operator splitting [5]. We compare the streamline front tracking method (SFTM) with a fast marching method (FFM) and a modified method of characteristics (MMOC).

3.1 Governing equations

The basic equations describing three-phase immiscible flow in a porous medium, say water (w), gas (g) and oil (o), are mass balance equations and Darcy's law. Assuming that the flow is incompressible, and that gravity can be neglected, the equations can be written in a global pressure/total velocity Should this be moved

to Chapter 2? formulation, see [20, 21], as follows:

$$\nabla \cdot \mathbf{v} = q(\mathbf{x}, t), \tag{3.1}$$

$$\mathbf{v} = -\lambda_T(\mathbf{x}, S_\alpha)\mathbf{K}(\mathbf{x}) \cdot \nabla p, \tag{3.2}$$

$$\phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot [F_\alpha \mathbf{v} - \epsilon_\alpha \nabla \cdot D_\alpha(\mathbf{x}, S_\alpha) \nabla S_\alpha] = q_\alpha(\mathbf{x}, t). \tag{3.3}$$

Here ϕ and $\mathbf{K}(\mathbf{x})$ are the porosity and absolute permeability of the porous medium; S_α, $\mathbf{v}_\alpha$, $k_{r\alpha}$, μ_α and q/q_α are, respectively, the reduced phase saturation, Darcy velocity, relative permeability, viscosity of phase α and sink/source terms. A global pressure p is derived from the phase pressures and the capillary pressures, see [21]. We refer to Paper B and C for more details on this standard formulation.

3.2 Solution strategy

To decouple the Pressure/Velocity equations (3.1), (3.2), from the saturation equations (3.3), we use sequential time stepping. Thus, for a given saturation-field, say at time t^n, we calculate a new velocity field. What about p? The saturation field is then advanced to a new time-step t^{n+1} by solving (3.3), using the most recent velocity field. This is continued sequentially up to a predetermined time $t = T$.

To solve the parabolic saturation Equation (3.3) with a given velocity field, we again use operator splitting as described in Section 2.2 on page 8. In previous works a Modified-Method-of-Characteristics (MMOC) has been used to solve the two-phase flow problem based on the ideas outlined above, see [22]. This method works excellently when the wave structure of the solution is known *a priori*. We present two alternative methods to the MMOC method, which both preserve the shape of self-sharpening fronts and are more flexible than the MMOC method in the sense that no a priori knowledge of the wave structure is required.

3.2.1 A streamline front tracking method

For simplicity we assume that the computational domain is discretized by a regular Cartesian grid such that the velocity $\mathbf{v} = \mathbf{v}(\mathbf{x}) \in RT_0$ is given, where RT_0 is the lowest order Raviart-Thomas space. Furthermore, we assume that all variables are known at cell centers at time-level t^n.

To obtain saturation values S^{n+1} at time-level t^{n+1}, we split Equation (3.3) into a hyperbolic equation

$$S_t + \mathbf{v} \cdot \nabla F(S) = 0, \tag{3.4}$$

and a parabolic heat type equation

$$S_t = \epsilon \nabla \cdot (D \nabla S). \tag{3.5}$$

where S and $F(S)$ are vectors of saturations and fluxes, respectively. The equations are solved in a standard operator splitting fashion, see previous chapter.

Now, consider the solution of Equation (3.4). Observe that on streamlines $\mathbf{r} = \mathbf{r}(\xi)$ such that

$$\frac{d\mathbf{r}}{d\xi} = \mathbf{v}, \tag{3.6}$$

Equation (3.4) becomes one-dimensional

$$S_t + F_\xi(S) = 0. \tag{3.7}$$

We exploit this to obtain new saturation-values at cell centers $\mathbf{x}_I$, in the following way: First, trace streamlines (3.6) analytically for $\mathbf{r}(0) = \mathbf{x}_I$ and $-\xi_{\max} < \xi < \xi_{\max}$. Here $\xi_{\max} = |\lambda_{\max}|(t^{n+1} - t^n)$ with $\lambda_{\max}$ being an estimate of the maximum wave speed of the system, such that the streamline covers the domain of dependence for $(\mathbf{x}_I, t^{n+1})$. The streamline is only traced in the upstream direction if all the wave speeds are positive. The piecewise constant cell values of the saturations are then projected onto these local streamlines, thus defining piecewise constant initial conditions for Equation (3.7). This conveniently arranges for Equation (3.7) to be solved by the front tracking method. (See Section 2.4.1 on page 11 for a description of the method.)

The main advantages of the front tracking method, is that the method is *super fast* [23], and preserves the frontal structure of the solutions extremely well. On the other hand, since the method heavily depends on solving Riemann problems, it is not easy to extend the method to three-phase flow. In Section 3.3, we will discuss a solution strategy for so-called triangular systems which may be a step towards an applicable solution procedure for three-phase flow problems.

Equation (3.5) may be solved by finite element or finite difference based methods. In this work we have for convenience used a standard explicit central finite difference scheme. Note that a local time step, $\Delta t_{\text{diff}} \leq t^{n+1} - t^n$, is required to satisfy the stability constraint inherent in the explicit finite difference method.

3.3 Triangular systems

A possible extension of the SFTM approach to three-phase flow, is to use an approximate Riemann solver to generate front speeds. However, to our knowledge, the construction of accurate and reliable Riemann solvers for fully coupled three-phase flow is not a trivial task, and we have chosen a different approach here.

Observe that the viscosity of the gas phase is usually at least an order less than the viscosities of the liquid phases. Motivated by this fact, it seems reasonable to assume that the fractional flow function of the gas phase can be approximated by a flux function which only depends on the saturation of the gas phase. Thus, we may consider the following 2×2-triangular hyperbolic system, as an approximation to the hyperbolic part of the fully coupled system (3.3):

$$\frac{\partial S_g}{\partial t} + \frac{\partial}{\partial x} F_g(S_g) = 0, \tag{3.8}$$

$$\frac{\partial S_w}{\partial t} + \frac{\partial}{\partial x} F_w(S_g, S_w) = 0. \tag{3.9}$$

Systems of this type have been investigated in [24, 25, 26], and existence and uniqueness of the solution to the Riemann problem is shown under general conditions.

3.3.1 A Riemann solver for triangular systems

The numerical construction of a solution for a Riemann problem associated with equations (3.8)-(3.9) was developed by Gimse [24]. The idea is to solve the Riemann problem for (3.8) first. The approximate solution to (3.8) consists of a set of constant states, say $S_g^L = S_g^1 < S_g^2 < \ldots < S_g^{N+1} = S_g^R$, separated by jump discontinuities traveling with the Rankine-Hugoniot shock speed:

$$s_i = \frac{F_g(S_g^{i+1}) - F_g(S_g^i)}{S_g^{i+1} - S_g^i}, \quad i = 1, 2, \ldots, N. \tag{3.10}$$

Within each wedge of the solution fan, the fractional flow of the water phase depends only on the water saturation and is given as:

$$F_w^i(S_w) \stackrel{\text{def}}{=} F_w(S_g^i, S_w), \quad i = 1, 2, \ldots, N+1.$$

Thus, we may easily solve for the water saturation within each wedge once we know the left- and right-hand state of the water saturation within the wedge. Obviously, successive left- and right-hand states over the discontinuities in the gas phase must also satisfy the jump condition

$$\frac{F_w^{i+1}(S_w^{i+1}) - F_w^i(S_w^i)}{S_w^{i+1} - S_w^i} = s_i, \quad i = 1, 2, \ldots, N, \tag{3.11}$$

There are infinitely many states S_w^i, which satisfy conditions (3.11), leading to the definition of the so-called H-sets:

$H_{1,in}$ The set of S_w-values in region S_g^L, that can be reached from S_w^L with speed $\sigma \leq s_1$.

$H_{i,in}$ The set of S_w-values in region S_g^i, $i = 2,3,\ldots N$, that can be reached from $H_{i,out}$ with speed σ such that $s_{i-1} \leq \sigma \leq s_i$.

$H_{i+1,out}$ The set of S_w-values in region S_g^{i+1}, $i = 1,2,\ldots N+1$, that can be reached from $H_{i,in}$ by a shock with speed s_i.

The solution can now be assembled by connecting the right state, S_w^R, to the left state, S_w^L, by admissible waves given by the H-sets. This procedure is described schematically in the following diagram:

$$\begin{aligned} S_w^R \rightarrow H_{n+1,out} \rightarrow H_{n,in} \rightarrow \cdots \\ \rightarrow H_{i+1,out} \rightarrow H_{i,in} \rightarrow \cdots \\ \rightarrow H_{2,out} \rightarrow H_{1,in} \rightarrow S_w^L. \end{aligned}$$

Note that a jump from one set to the next always happens at the first possible value in the current H-set.

Gimse has shown [24] that the above construction gives a *unique* solution of the triangular system, if the following conditions hold:

$$(A)\ F_g(0) = 0, \quad (B)\ F_g(1) = 1, \quad (C)\ F_w(S_g, 1-S_g) = 1 - F_g(S_g)$$

$$(D)\ \frac{\partial F_w}{\partial S_w} \geq 0, \quad (E)\ \frac{\partial F_w}{\partial S_g} < 0, \quad (F)\ F_g'(S_g) \geq 0.$$

For a discussion of these conditions we refer to Paper C.

3.3.2 Approximation of a full three-phase flow system by a triangular system

Since the fractional flow function for the gas phase is nearly independent of the water saturation, it seems natural to decouple the gas phase from the other phases. This can be done simply by plugging in a value for the water saturation, S_w^0, in the gas fractional flow function:

$$F_g(S_w, S_g) \approx F_g(S_w^0, S_g) \stackrel{\text{def}}{=} F_g(S_g). \tag{3.12}$$

We can choose S_w^0 so that the decoupled fractional flow function is as close as possible to the complete function in some norm over the admissible section of state space, see [27]. However, in the experiments reported in Paper C, S_w^0 is chosen more or less arbitrary to be zero.

In situations where the full solution for the gas phase consists of two rarefaction waves connected by an intermediate shock wave, the triangular approximation must fail because it cannot produce such a wave structure. A possible solution to this problem is to let S_w^0 be defined locally for each Riemann problem which is solved. This will allow for some feedback, and can be combined with the fact that the total mobility λ_T acts as an approximate invariant for the transport. Thus $\lambda_T(S_w, S_g) \approx \lambda_T(S_w^0, S_g)$ may be used to eliminate S_w from F_g in a more accurate way. However, this idea was not pursued any further in the paper.

Another difficulty that arises from the approximation (3.12), is that condition (C) is violated since

$$F_w(1 - S_g, S_g) = 1 - F_g(1 - S_g, S_g) \neq 1 - F_g(S_w^0, S_g).$$

If condition (C) is not satisfied the construction of the solution may fail in two ways for values close to $S_o = 0$: Either the construction of appropriate H-sets will fail, or the tracking of admissible waves becomes impossible. Since $S_o \approx 0 \implies S_w \approx 1 - S_g$, we may circumvent the problem by replacing S_w^0 with $1 - S_g$ in (3.12) when the oil phase is close to residual. Again, this requires a local definition of S_w^0 which has not been implemented in our simulator yet.

3.4 Concluding remarks

For the detailed discussion of the numerical results, we refer to Paper B and C. A front tracking streamline method (SFTM) for multi-phase flow in porous media was presented. The main advantage of this method is to handle the advective part of the nonlinear transport using streamline information, and still be able to solve for diffusive/dispersive effects on a regular grid. The method has been compared with a Modified Method of Characteristics (MMOC) and a Fast Marching (FMM) approach for two-phase flow problems. The solutions obtained seem to be equally accurate. The SFTM and the MMOC are comparable when it comes to computational efficiency, whereas the FMM gives a much faster advection solver. However, compared to the MMOC and the FMM, the SFTM is more flexible and has less restrictions with respect to the complexity of the problems that may be solved.

Using the H-set method, the SFTM approach has been extended to solve three-phase flow problems which are triangular. Since most three-phase flow problems are fully coupled in both saturations, triangular systems can only be approximate. However, because the viscosity of the gas is much smaller than the viscosities of the liquid phases the gas phase is often nearly decoupled from the other phases. We showed that the solution of a *naive* Heavy sentence triangular

approximation of a fully coupled system, may and may not be a good approximation to the solution of the fully coupled system.

In these two papers standard operator splitting was applied, i.e, without correction in the diffusion step. Including a correction term would possibly give sharper fronts, as explained in Chapter 2. However, defining the residual flux is not straight forward for multi dimensional problems. A main difficulty is to find the correction domain. After obtaining the solution along the streamlines in the hyperbolic step, one needs to project not only the solution, but also the correction interval down onto the grid for the parabolic step. To find a local correction domain, information from several streamlines at a time must probably be used. Hence, it might also be hard to choose the correct shock values.

Chapter 4

Interfacial area and dynamic network models

This chapter covers ideas and results from Paper D, *"Two phase flow including interfacial area as a variable "*, and Paper E, *"A pore network model for calculation of interfacial velocities "*. The work presented is two folded; a generalization of the standard two-phase flow model and testing of this model using a dynamic network model.

Multi-phase porous media systems are characterized by fluid-fluid interfaces that exist at the pore scale. These interfaces define the spatial boundaries of each phase at any given instant in time. Interfaces also have properties such as interfacial tension, which allows each fluid to maintain a different pressure. The resulting difference between individual phase pressures is usually called capillary pressure. Mass is transferred from one fluid phase to another across the fluid-fluid interfaces. Fluid-solid interfaces provide similar surfaces for mass exchanges between fluid and solid phases. For these reasons, an understanding of interfacial behavior at the pore scale, and subsequent scaling of that behavior to the more practical continuum scale, is important to give proper descriptions of porous media flow systems.

4.1 A generalized two phase flow model

As mentioned in the introduction, Section 1.1 on page 3, traditional multi-phase flow models are usually based on direct extensions of one-phase flow equations. These models are *ad hoc* and involve hidden assumptions that may be difficult to uncover in a practical application. Furthermore, it may not be clear whether all important physical effects are appropriately accounted for in the mathematical formulations. This may also explain the apparent hysteresis in the constitutive

relationships.

A rigorous approach to modeling of multi-phase flow is given by Gray and Hassanizadeh, e.g. [2]. This approach starts from first principles and is aiming at a more complete description of multi-phase flow phenomena. In particular, their theoretical arguments imply that fluid-fluid interfacial areas, a^{wn}, should be primary variables in the mathematical formulation of two-phase flow. Their analysis, based on thermodynamics, also includes other variables, like contact lines and common points, but these will not be considered here. In this theory, it is conjectured that capillary-pressure hysteresis is a consequence of incompleteness of traditional models. A non-hysteretic constitutive relationship would require that the $P_c - s^w$ relationship should be extended by at least a third variable, e.g. P_c, s^w, and a^{wn}. This conjecture has been tested using computational (quasi-static) pore-scale network models, see [28, 29, 30]. The results obtained from these network models give some support to the theoretical conjecture. However, experimental validation is still required.

An important issue is whether the inclusion of interfacial area in a two-phase flow model changes the nature of the solution for the saturation. Of course there are situations where the interface itself is of interest, but in the oil reservoir setting the saturation is what really matters. In Paper D, which is presented in the following sections, this question is addressed and to some degree answered.

4.1.1 Assumptions and governing equations

Following the paper we will take a general set of equations based on the theory of Gray and Hassanizadeh, [2], and reduce them to a system of two equations for saturation and interfacial area using a standard fractional flow formulation. The major assumptions applied are:

1. No phase change occurs;
2. Isothermal system;
3. Immobile, non-deforming solid;
4. Common lines are neglected;
5. The solid-fluid interface dynamics are negligible (i.e no film flow);
6. Massless fluid-fluid interface;
7. Incompressible fluids;
8. Inertial terms are negligible in the momentum equations;
9. Interfacial tensions are all constant and specified.

These assumptions most certainly restrict the applicability of the model, but they are needed to reduce the complexity of the general set of equations. In a preliminary analysis such as this, the model is still useful because it enables us

to get some insight into the properties of two phase flow models that include interfacial area. Notice, that by using this general framework all the assumptions must be clearly stated.

A complete list of equations and variables is found in Paper D. Only the most important equations will be given here. The mass and momentum conservation for each phase is governed by standard equations. The interface was assumed massless and hence there is no mass conservation equation for the interface. The momentum conservation equation for a massless interface indicates that the interfacial velocity is a weighted sum of the velocities of the adjacent phases. Thus, the following equality is assumed:

$$(\mathbf{R}^w_{wn} + \mathbf{R}^n_{wn}) \cdot \mathbf{v}^{wn} = \mathbf{R}^w_{wn} \cdot \mathbf{v}^w + \mathbf{R}^n_{wn} \cdot \mathbf{v}^n \tag{4.1}$$

where $\mathbf{R}^\alpha_{wn}$ is the resistance term for the α-phase due to the wn-interface (fluid-fluid) and $\mathbf{R}^\alpha_{\alpha s}$ the resistance term for the α-phase due to the αs-interface (fluid-solid). $\mathbf{v}^\alpha$ is the velocity of α-phase and $\mathbf{v}^{wn}$ is the velocity of the wn-interface. Resistance terms can be viewed upon as the inverse of conductance and is a function of $\mathbf{K}^{-1}$.

In addition, a geometry equation for interfacial area has to be satisfied:

$$\frac{\partial a^{wn}}{\partial t} + \nabla \cdot [\mathbf{G}^{wn} a^{wn} \cdot \mathbf{v}^{wn}] = 0. \tag{4.2}$$

Here a^{wn} is the specific interfacial area (fluid-fluid) and $\mathbf{G}^{wn}$ is a geometric tensor. Also, the capillary pressure equation

$$\gamma^{wn} J^w_{wn} = p^w - p^n = -P_c \tag{4.3}$$

must hold, where J^w_{wn} is the average curvature and γ^{wn} is the surface tension of the wn-interface.

Equation (4.2) and (4.3) are in fact simplified versions of the corresponding equations arising from the general framework using the assumptions outlined in the start of this section. In Equation (4.2) the production term is neglected. In general, this is a significant assumption that limits the general applicability of the system of equations. However, it will be applied consistently here with the objective of performing a preliminary analysis of a simplified set of equations. Equation (4.3) is reduced from a dynamic capillary equation by neglecting the time derivative of the saturation. We refer to Section 2 in Paper D for the full equations and motivations. Notice, however, that adding a production term in Equation (4.2), might make it hard, if not impossible, to find a fractional flow formulation of the equations.

4.1.2 Fractional flow formulation

By eliminating all variables but saturation and interfacial area, and neglecting the gravity terms, we get the following fractional flow formulation

$$\begin{aligned} \varepsilon \frac{\partial s^w}{\partial t} + \mathbf{u} \cdot \nabla \cdot \mathbf{F}_w &= \nabla \cdot (\mathbf{D}_w \cdot \nabla P_c), \\ \frac{\partial a^{wn}}{\partial t} + \frac{1}{3} \left(\mathbf{u} \cdot \nabla \cdot (\mathbf{F}_{wn} a^{wn}) \right) &= \frac{1}{3} \nabla \cdot (a^{wn} \mathbf{D}_{wn} \cdot \nabla P_c) \end{aligned} \tag{4.4}$$

where $\mathbf{F}_\alpha$ is the fractional flow (of the phase or interface), $\mathbf{D}_\alpha$ is the diffusion function (for the phase or interface), $\mathbf{u}$ is the Darcy velocity and ε is the porosity. This constitutes a 2×2 coupled system for the saturation and the interfacial area.

To close the system we need a constitutive relationship for capillary pressure as a function of saturation and interfacial area. To our knowledge, no such relationship has been determined experimentally. Thus, we construct this relationship based on results from network models, see e.g. [29]. In addition, functional forms of the resistance terms are needed in the fractional flow and diffusion function. Details on this are given in Section 4 of Paper D.

There are two difficulties related to the system in (4.4), and both arise from the capillary pressure relationship. First of all the $P_c - s^w - a^{wn}$ surface is not defined for all value pairs of saturation and interfacial area. This might be solved by some kind of (smooth) extrapolation or extention of the surface. We avoided this problem by doing a careful choice of initial values. This is in general not enough since one cannot guarantee that the solution will stay on the surface. However, it is possible to reformulate the system as a set of equations for saturation and capillary pressure. We did not pursue this in the article, but it is our assumption that this might be less restrictive than the proposed system.

The second problem is that the system in some sense is overdetermined. If the system is solved sequentially, the (new) saturation and the (old) capillary pressure will be available when the interfacial area is solved for. Hence, we are then in the position to get the interfacial area from the capillary pressure relationship without actually solving the equation. This was not investigated any further, but we find it appropriate to point this problem out.

For a numerical example we refer to Paper D again. Some concluding remarks follow in Section 4.3.

4.2 Dynamic network models and equation testing

Network models have been used to investigate physical phenomena in multi phase flow that are hard to observe in laboratory experiments. Typically, these models

have been static in the sense that they just track the interface from one stable position to another without taking time into account. This is sufficient for studying fingering effects, functional relationships that do not depend on time and so on. One example that illustrates what can be achieved by such models, is the determination of a functional relationship between capillary pressure, saturation and interfacial area, see e.g. [29].

The view that interfacial area plays an important role in porous media flows is supported by the already mentioned theoretical developments reported by Gray and Hassanizadeh, [2]. In their work, a thermodynamic approach is used to show that the relationship between capillary pressure and saturation is incomplete, and that interfacial area should enter into the relationship.

One of the problems associated with tests of theories involving interfacial area, is the lack of experimental validation. It is very difficult to determine interfacial area, and even more difficult to determine interfacial dynamics, which tends to occur on very short time scales. Therefore, testing of these theories must rely on using specific kinds of computational models, including so-called pore-scale network models. Pore-scale network models typically represent the pore space of the medium using simplified geometries. Within this geometric representation equations are solved to explicitly track the location of all fluid-fluid interfaces within the network.

The models are often run to mimic typical laboratory experiments, such as pressure cell tests to determine the relationship between capillary pressure and relative fluid saturation. This is accomplished by using lattices that are sufficiently large to define meaningful continuum-scale measures, such as fluid saturation. In other words, the network should represent one or more REVs. Results of such simulations show all of the major features of experimental relationships, including finite entry pressures, residual saturations, and hysteresis.

Two general types of pore-scale network models may be identified: quasi-static models and dynamic models. In a quasi-static model, the location of any fluid-fluid interface is governed by equilibrium considerations only. Equilibrium states are determined from the Young-Laplace equation, which relates the capillary pressure to the interfacial tension and the interface curvature, viz.

$$P_c = \frac{\sigma}{R}. \tag{4.5}$$

In this equation, σ is the interfacial tension and R is the radius of curvature. For a given imposed capillary pressure there are rules to determine whether or not an interface is stable at a given location. If the interface is unstable, it is moved through the network until a stable position is found, or until it exits the network. No time dependence is included in the calculation. Examples of these kinds of models include those described by Dullien [31], Ferrand and Celia [32], and Reeves [28].

A second type of pore network model involves computation of transient behavior associated with interface movement. That is, unstable interfaces are tracked through the network until a stable position is reached, but the transient nature of the movement from one position to another is explicitly described and modeled. Most pore-scale network models reported in the literature are quasi-static, but there are several dynamic models that have been developed. These include the model of Blunt and King [33]; a series of models by Payatakes and coworkers (see, for example, Valavanides and Payatakes [34]), and more recent models by Mogensen and Stenby [35], Aker et al. [36], Hassanizadeh (Dijkstra et al. [37]) and by Dahle and Celia [38]. The models of Payatakes are the most comprehensive, including a focus on mobilization of trapped fluids and so-called drop traffic flows.

To study the equations in the generalized two phase flow model in Section 4.1, static network models are not sufficient; we need dynamic network models. In Paper E, we used a model based on the original work of Blunt and King [33] and calculated dynamic interfacial behavior, with focus on volume-averaged interfacial velocity. The most important features of the model are described, and a definition of average interfacial velocity is provided, in the next section. In the paper specific calculations were performed to demonstrate how continuum-scale measures of both interface velocity and phase velocity can be quantified. The model was then used to test a specific theoretical conjecture regarding interfacial velocity and its relationship to average phase velocities.

4.2.1 The dynamic network model

The pore-scale network model used in Paper E is an extension of the model of Blunt and King [33]. The pore network is a rectangular lattice, with spherical pore bodies and cylindrical pore throats, with pore-size distributions defined for the bodies and throats, see Figure 4.1 below. Following Blunt and King [33], the model is simplified by the following assumptions:

1. Local capillary pressure in the pore throats is assumed to be negligible, so that only one pressure exists within a pore body, independent of the local saturation of that pore body.
2. While the radius of a pore throat serves to define its hydraulic conductance, the volume contributed by the pore throat is assumed to be small relative to volumes of pore bodies. Therefore, movement of an interface through a pore throat is assumed to occur instantaneously.
3. Flow within pore throats is assumed to be laminar and given by Poiseuille's law, see Equation (4.8) and (4.7).
4. Both fluids are assumed to be incompressible.

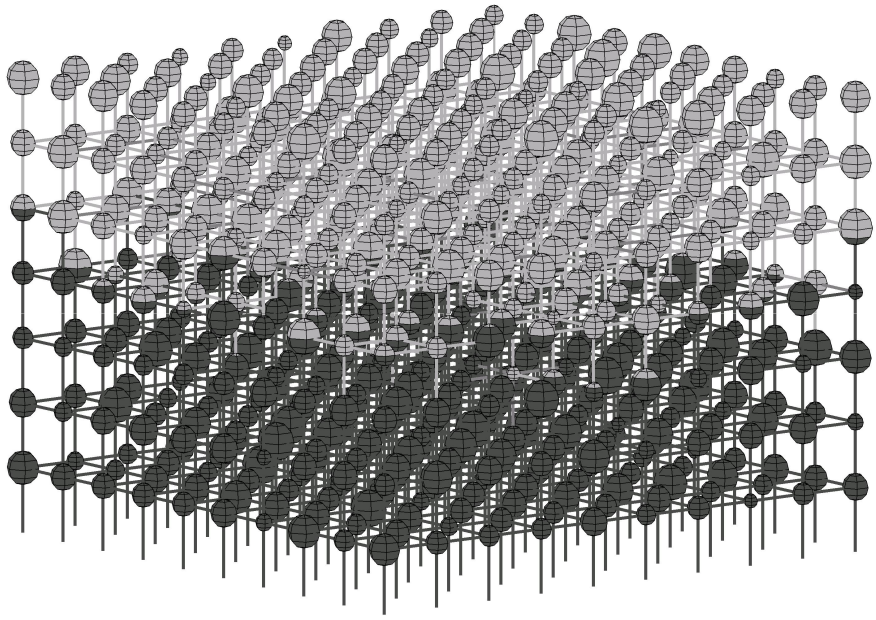


Figure 4.1: Example of a network model showing stable displacement.

With these assumptions, the set of governing equations is relatively simple. Each equation must obey volume conservation within each pore body, such that

$$V_i \frac{\partial S_i^\alpha}{\partial t} + \sum_{j \in N_i} Q_{ij}^\alpha = 0. \tag{4.6}$$

Here V_i represents the volume of pore body i, S^α represents local saturation (percent of V_i filled with fluid α), Q_{ij}^α is the volumetric flux from pore body i to it's neighbor j, and N_i is a list of all neighbor pore bodies for pore body i. This equation is written for both fluid phases, wetting ($\alpha = w$) and non-wetting ($\alpha = n$). The volumetric flux is related to pressures at the pore bodies by Poiseuille's law,

$$Q_{ij}^\alpha = G_{ij}^\alpha (p_i^\alpha - p_j^\alpha) \tag{4.7}$$

where p_l^α represents pressures, and G_{ij}^α represents hydraulic conductance in the pore throat connecting pore bodies i and j. Because the pore throats are cylindrical and interface movement through them is instantaneous, only one fluid can occupy a given pore throat at a given time. Therefore, the fluid occupying the pore throat has conductance

$$G_{ij}^\alpha = \frac{\pi r_{ij}^4}{8 \mu^\alpha l_{ij}}, \tag{4.8}$$

while the non-occupying fluid has zero conductance. Here r_{ij} and l_{ij} is the radius and the length of the pore throat, respectively. Summation of Equation (4.6) over the two phases gives the equation

$$\sum_{j \in N_i} (Q_{ij}^w + Q_{ij}^n) = 0 \tag{4.9}$$

Substitution of Equation (4.7) for each of the phase fluxes Q_{ij}^α provides a set of algebraic equations with the pore-body pressures as unknowns. These can be solved using standard matrix solution methods. Time steps are chosen so that during any time step, only one pore body reaches full non-wetting phase saturation.

As the model size increases the matrix solver becomes very important. The matrix is sparse, banded and also symmetric if contributions from boundary (and trapped) pore bodies are moved to the right hand side. However, having a condition number of 10^{-8} or more, the matrix system is hard to solve. Using a direct solver will slow the computations down dramatically even for matrices with narrow band width. We used a preconditioned conjugate gradient method with incomplete Cholesky factorization as preconditioner, and it worked very well. For a network with 5000 nodes the speed-up compared to a direct solver was only four, but for a network with 50000 nodes the speed-up was over 100. These results are for illustration only since we did not do a rigorous convergence test. The number of off diagonal rows that are non-zero is equal to the maximal coordination number of the nodes. For a three dimensional rectangular lattice the maximal coordination number is six, i.e, one node can be connected to six other nodes. For a irregular lattice this number may increase and having a good matrix solver is even more important.

Overall, the algorithm proceeds as an Implicit Pressure Explicit Saturation (IMPES) routine. The major unknowns are the pressure and saturation of each pore body. For a given distribution of fluids, phase conductances are calculated and substituted into Equation (4.9), which is solved for a new pressure field. That pressure field is then used in Equation (4.7) to compute fluxes through the pore throats. These fluxes are then used (in conjunction with knowledge of the current saturations in each pore body) to determine the minimum filling time for each of the pores, and this is set as the time step size. Then Equation (4.6) is used to update the saturations in each pore body. Newly created interfaces are tested for stability, conductances are updated, and the procedure is repeated.

4.2.2 Calculating average variables

Following the article we now define average variables over a representative volume. The volume might be chosen to correspond to the entire volume of the

network, or it might be defined as essentially two-dimensional slices through the network in the direction perpendicular to the macroscopic direction of displacement. In either case, the volume-averaged saturation is defined as

$$S^{\alpha} = \frac{\sum_{i \in N_{vol}} V_i S_i^{\alpha}}{\sum_{i \in N_{vol}} V_i} = \frac{1}{V} \sum_{i \in N_{vol}} V_i S_i^{\alpha} \tag{4.10}$$

where V is the volume of the chosen averaging region, and N_{vol} denotes the set of pore bodies within the chosen averaging volume. Capillary pressure is defined as the difference between volume-averaged phase pressures, such that

$$P_c = p_n - p_w = \frac{1}{V}\Big(\sum_{i \in N_{vol}} V_i S_i^n p_i - \sum_{i \in N_{vol}} V_i S_i^w p_i \Big). \tag{4.11}$$

Here p_i corresponds to the pressure in pore body i. Notice that while no local capillary pressure exists (by assumption in the model), a macroscopic capillary pressure is still well defined based on the average phase pressures.

Average properties associated with interfaces require somewhat more care in their definitions. Because no capillary pressure is associated with individual pore bodies, the shape (especially the curvature) of a particular interface is not specified. The only information for active (non-trapped) interfaces is that they reside in a specific pore body. The interfacial area in a pore body is of course a function of saturation, but for simplicity we assumed it to be constant equal to the cross section of the pore body. Those interfaces trapped at the entrance of pore throats are assigned the area associated with the diameter of the pore throat. Therefore, the specific interfacial area (defined as the amount of interfacial area per unit volume of porous medium) is defined as

$$a^{wn} = \frac{1}{V}\Big(\sum_{i \in N_{vol}} a_i^{wn} + \sum_{j \in M_{vol}} a_j^{wn} \Big) = \frac{A^{wn}}{V} \tag{4.12}$$

where M_{vol} denotes the set of pore throats that are contained within the averaging volume V.

The averaged variable that is the most difficult to define is the volume-averaged interfacial velocity. To define a finite local velocity that preserves proper global velocities, the pore filling associated with saturation changes is extended in length to cover the combined pore body – pore throat combination. The 'locally averaged' velocity for a specific interface is defined as

$$\|\mathbf{v}_i^{wn}\| = l_{i,j} \frac{\Delta S_i^n}{\Delta t}. \tag{4.13}$$

The double brackets signify magnitude of the interface velocity vector, the length $l_{i,j}$ denotes the length of the pore throat through which the entering interface travels plus the diameter of the pore body i, and ΔS_i^n denotes the saturation change over the time interval Δt. The direction assigned to the interfacial velocity vector is the average of the total inflow vector and total outflow vector. Finally, the volume-averaged velocity vector is given by the sum of each interfacial velocity weighted by the area of the interface,

$$\mathbf{v}^{wn} = \frac{1}{A^{wn}} \sum_{i \in N_{vol}} \mathbf{v}_i^{wn} a_i^{wn} \tag{4.14}$$

Volume-averaged phase velocities may be defined analogously to average interfacial velocities with volume replacing areas as the appropriate weights. Hence,

$$\mathbf{v}^{\alpha} = \frac{1}{V S^{\alpha}} \sum_{i \in N_{vol}} \mathbf{v}_i^{\alpha} V_i S_i^{\alpha} \tag{4.15}$$

where $\mathbf{v}_i^{\alpha}$ is given by an equation similar to Equation (4.13), with appropriate modification for the case of $S = 1$ or $S = 0$.

Another possible definition of velocities arises if we consider the flow in the pore throats and pore bodies separately. Let

$$\|\mathbf{v}_{jt}^{\alpha}\| = \frac{r_j^2}{8\mu^{\alpha} l_j} \|\Delta p_j\| \tag{4.16}$$

and

$$\|\mathbf{v}_{ib}^{\alpha}\| = 2R_i^2 \frac{\Delta S_i^n}{\Delta t} \tag{4.17}$$

be the phase velocity in pore throat j and pore body i, respectively. For the throats the actual flux is used to find the velocity, and for the pore bodies a 'locally averaged' velocity. Then we define the volume-averaged phase velocity as

$$\mathbf{v}^{\alpha} = \frac{1}{V S^{\alpha}} \Big(\sum_{i \in N_{vol}} \mathbf{v}_{ib}^{\alpha} V_{ib} S_i^{\alpha} + \sum_{j \in M_{vol}} \mathbf{v}_{jt}^{\alpha} V_{jt} \Big), \tag{4.18}$$

and the volume-averaged interface velocity analogously. This definition was not used in the article since it did not reveal any new information, i.e, the results from the equation testing were qualitatively the same.

4.2.3 Equation testing

In Paper E we wanted to test Equation (4.1). The following procedure was used: The phase velocities from the network model were used to get the theoretical interface velocity which then was compared to the interface velocity from the network

model. As the numerical results from the paper show, the match depends on the flow regime, i.e., stable or unstable, and whether or not the trapped interfaces are included.

In Paper D we did not discuss the freedom associated with Equation (4.1). For a scalar $\mathbf{K}$ we get the following expression for the interface velocity

$$\mathbf{v}^{wn} = F^w \mathbf{v}^w + F^n \mathbf{v}^n = \frac{f^w}{f^w + f^n}\mathbf{v}^w + \frac{f^n}{f^w + f^n}\mathbf{v}^n \tag{4.19}$$

where

$$f^w = \mu^w h^w(s^w)(s^w)^2,$$
$$f^n = \mu^n h^n(s^n)(s^n)^2,$$

when the definitions of the resistance terms from Paper D are used. For h^α we assume a more general form than was used in the article, namely

$$h^\alpha(s^\alpha) = \beta^\alpha(1 - s^\alpha)^p \tag{4.20}$$

To determine the β-parameters we put the following restrictions on F^α:

$$\lim_{s^w \to 1} F^w(s^w) = 0 \tag{4.21}$$
$$\lim_{s^w \to 0} F^w(s^w) = 1 \tag{4.22}$$
$$\lim_{s^w \to 1} F^n(s^n) = \lim_{s^n \to 0} F^n(s^n) = 1 \tag{4.23}$$
$$\lim_{s^w \to 0} F^n(s^n) = \lim_{s^n \to 1} F^n(s^n) = 0 \tag{4.24}$$

These restrictions are motivated by the observation that the interface velocity is close to the non-wetting phase velocity when the drainage starts and close to the wetting phase velocity when close to residual saturation. Below, the limits are displayed for different choices of p:

	$p = 1$	$p = 2$	$p > 2$
$\lim_{s^w \to 1} F^w(s^w)$	1	L^w	0
$\lim_{s^w \to 0} F^w(s^w)$	0	L^w	1
$\lim_{s^w \to 1} F^n(s^n)$	0	L^n	1
$\lim_{s^w \to 0} F^n(s^n)$	1	L^n	0

The entity L^α is given by $L^\alpha = \mu^\alpha\beta^\alpha/(\mu^w\beta^w + \mu^n\beta^n)$. We observe that p has to be larger than two in order to satisfy Equation (4.21) - (4.24). We do not get any constraints on the choice of β^α. Actually, several test cases showed that the best

choice (measured in the quality of the match between the theoretical and network model interface velocity) was $\beta^\alpha = 1/\mu^\alpha$. Together with the choice of p equal to three, this defined the relationship between the volume-averaged interface velocity and the volume-averaged phase velocities that we tested in the article.

4.3 Concluding remarks

In Paper D initial numerical solutions for an enhanced model of two phase flow which includes fluid-fluid interfacial area as a primary variable were presented. For the particular case studied, the saturation profiles show little dependency on interfacial area. Since the correct functional forms of the resistance terms used in the model are not known, more general studies are required to assess the practical importance of the interfacial area equations to flow modeling. In addition, numerical solutions need to be obtained for cases where the simplifying assumptions applied to the governing equations are systematically examined. The results in this paper represent a starting point for this more general analysis.

While we cannot reach any general conclusions based on our initial calculations, we can make a few general observations. First, inclusion of the interfacial area equations allows for direct calculation of the amount of interfacial area in the system. Equations written for the individual phases also incorporate effects of the interfaces through functional dependencies in the appropriate nonlinear coefficients. In addition, inclusion of a unique functional relationship between P_c, s^w, and a^{wn}, allows hysteresis within the $P_c - s^w$ plane to be incorporated into the algorithm with no additional effort. Arbitrary drainage and imbibition cycles can be simulated without redefining the constitutive curves. This appears to be a major advantage of this approach to multi phase flow modeling. The cost of such additional generality and flexibility is the need to identify and quantify additional parameters, as well as their functional dependencies on interfacial area. Additional equations also need to be solved compared to the traditional equations for multi phase flow.

The overall significance of interfacial areas in the mathematical description of multi phase flow in porous media, remains to be determined. The degree to which the governing equations can be simplified, must be explored numerically. This requires a systematic and comprehensive numerical approach. The results presented in Paper D represent a first contribution to the overall numerical study.

The dynamic network model presented in Paper E is relatively simple and leads to numerical calculations that are straight-forward to implement. The calculation of averages allows many continuum-scale variables to be calculated, and thereby facilitates testing of new theories that involve non-traditional variables. We focused on interfacial dynamics, and calculation of average interfacial ve-

locities. The results were used to test a specific conjecture that relates average interfacial velocity to average phase velocities.

In particular, we conclude that the proposed equation that relates interfacial velocity to phase velocities only hold under very specific conditions, namely piston-like stable displacements in which trapped interfaces are neglected. Under more general conditions the equation provides a poor prediction. To close the extended set of equations for two-phase flow, which include specific interfacial area as a primary variable, new constitutive equations need to be developed.

While Paper E focused on a specific constitutive equation, the general approach is illustrative for the kind of problems that can be solved with dynamic pore-scale network models. Because the local-scale information is highly detailed, many upscaled variables can be calculated. This greatly facilitates testing of new theories, and has the potential to provide significantly improved insight into fundamental behavior of two-phase porous media flows.

Chapter 5

Conclusions

This chapter contains a summary of conclusions and remarks made in the previous chapters. Since the presented work is done in collaboration with other researchers, I will also point out some of my main contributions.

Corrected operator splitting for systems. In Paper A results for two particular 2×2 systems describing flow in porous media were presented. Simulations for a two-phase multicomponent model, a polymer system, and for a triangular three-phase flow model were reported. In addition, the polymer system was solved in 2D using dimensional splitting.

We demonstrate numerically that operator splitting (OS) methods for systems of convection-diffusion equations in one space dimension, have a tendency to be too diffusive near viscous shock waves. The idea behind the scalar COS method is to use the wave structure from the convection step to identify where the nonlinear splitting error (or entropy loss) occurs. The potential error is then compensated for in the diffusion step (or in a separate correction step).

Similar to the scalar case, the splitting error is closely related to the local linearizations introduced implicitly in the convection steps due to the use of an entropy condition. A COS method for systems was proposed. The numerical examples demonstrate that the COS method is significantly more accurate than the corresponding OS method when the splitting step is large and the solution consists of (moving) viscous shock waves.

My contributions: The triangular three-phase flow solver was implemented by me, using the theory in [24]. To our knowledge this was the first attempt to solve this system using a COS method. In the process I also contributed to the general methodology and to the definition of the residual flux for systems.

Front tracking along streamlines. For the detailed discussion of the numerical results, we refer to Paper B and C. A front tracking streamline method (SFTM) for multi-phase flow in porous media was presented. The main advantage of this method is to handle the advective part of the nonlinear transport using streamline information, and still be able to solve for diffusive/dispersive effects on a regular grid. The method has been compared with a Modified Method of Characteristics (MMOC) and a Fast Marching (FMM) approach for two-phase flow problems. The solutions obtained seem to be equally accurate. The SFTM and the MMOC are comparable when it comes to computational efficiency, whereas the FMM gives a much faster advection solver. However, compared to the MMOC and the FMM, the SFTM is more flexible and has less restrictions with respect to the complexity of the problems that may be solved.

Using the H-set method, the SFTM approach has been extended to solve three-phase flow problems which are triangular. Since most three-phase flow problems are fully coupled in both saturations, triangular systems can only be approximate. However, because the viscosity of the gas is much smaller than the viscosities of the liquid phases the gas-phase is often nearly decoupled from the other phases. We showed that the solution of a *naive* Heavy sentence triangular approximation of a fully coupled system, may and may not be a good approximation to the solution of the fully coupled system.

In these two papers standard operator splitting was applied, i.e, without correction in the diffusion step. Including a correction term would possibly give sharper fronts, as explained in Chapter 2. However, defining the residual flux is not straight forward for multi dimensional problems. A main difficulty is to find the correction domain. After obtaining the solution along the streamlines in the hyperbolic step, one needs to project not only the solution, but also the correction interval down onto the grid for the parabolic step. To find a local correction domain, information from several streamlines at a time must probably be used. Hence, it might also be hard to choose the correct shock values.

My contributions: I implemented the three phase streamline solver, most of the two phase streamline solver and the streamline generator (for a given velocity field). I would like to thank Knut A. Lie for letting me use his front tracker code and Johnny Frøyen for the pressure/velocity solver. I contributed strongly to the development of these methods and in particular with questions related to triangular systems.

Interfacial area and dynamic network models. In Paper D initial numerical

solutions for an enhanced model of two phase flow which includes fluid-fluid interfacial area as a primary variable were presented. For the particular case studied, the saturation profiles show little dependency on interfacial area. Since the correct functional forms of the resistance terms used in the model are not known, more general studies are required to assess the practical importance of the interfacial area equations to flow modeling. In addition, numerical solutions need to be obtained for cases where the simplifying assumptions applied to the governing equations are systematically examined. The results in this paper represent a starting point for this more general analysis.

While we cannot reach any general conclusions based on our initial calculations, we can make a few general observations. Firstly, inclusion of the interfacial area equations allows for direct calculation of the amount of interfacial area in the system. Equations written for the individual phases also incorporate effects of the interfaces through functional dependencies in the appropriate nonlinear coefficients. In addition, inclusion of a unique functional relationship between P_c, s^w, and a^{wn}, allows hysteresis within the $P_c - s^w$ plane to be incorporated into the algorithm with no additional effort. Arbitrary drainage and imbibition cycles can be simulated without redefining the constitutive curves. This appears to be a major advantage of this approach to multi phase flow modeling. The cost of such additional generality and flexibility is the need to identify and quantify additional parameters, as well as their functional dependencies on interfacial area. Additional equations also need to be solved compared to the traditional equations for multi phase flow.

The overall significance of interfacial areas in the mathematical description of multi phase flow in porous media, remains to be determined. The degree to which the governing equations can be simplified, must be explored numerically. This requires a systematic and comprehensive numerical approach. The results presented in Paper D represent a first contribution to the overall numerical study.

The dynamic network model presented in Paper E is relatively simple and leads to numerical calculations that are straight-forward to implement. The calculation of averages allows many continuum-scale variables to be calculated, and thereby facilitates testing of new theories that involve non-traditional variables. We focused on interfacial dynamics, and calculation of average interfacial velocities. The results were used to test a specific conjecture that relates average interfacial velocity to average phase velocities.

In particular, we conclude that the proposed equation that relates interfacial

velocity to phase velocities only hold under very specific conditions, namely piston-like stable displacements in which trapped interfaces are neglected. Under more general conditions the equation provides a poor prediction. To close the extended set of equations for two-phase flow, which include specific interfacial area as a primary variable, new constitutive equations need to be developed.

While Paper E focused on a specific constitutive equation, the general approach is illustrative for the kind of problems that can be solved with dynamic pore-scale network models. Because the local-scale information is highly detailed, many upscaled variables can be calculated. This greatly facilitates testing of new theories, and has the potential to provide significantly improved insight into fundamental behavior of two-phase porous media flows.

My contributions: In Paper D the major part of the development of the general two phase flow model and the implementation of the numerical solver was done by me. In Paper E an existing network model was improved and extended by me to allow for calculation of several average variables and general rectangular geometry. In addition I was a major contributor in the discussions concerning the definition of average quantities based on the pore scale calculations.

Bibliography

[1] K. Aziz and A. Settari, *Petroleum Reservoir Simulation*. Elsevier Applied Science Publishers, 1985.

[2] W. Gray and S. Hassanizadeh, "Macroscale continuum mechanics for multiphase porous-media flow including phases, interfaces, common lines and common points," *Advances in Water Resources*, vol. 21, no. 4, pp. 261–281, 1998.

[3] G. Dagan, *Flow and transport in porous formations*. Springer-Verlag, 1989.

[4] K. Morton, *Numerical Solution of Convection-Diffusion Problems*. Chapman & Hall, 1996.

[5] M. Espedal and K. Karlsen, "Numerical solution of reservoir flow models based on large time step operator splitting algorithms," in *Filtration in porous media and industrial application (Cetraro, 1998)*, pp. 9–77, Berlin: Springer, 2000.

[6] H. Dahle, *Adaptive characteristic operator splitting techniques for convection-dominated diffusion problems in one and two space dimensions*. PhD thesis, Dept. of Mathematics, Univ. of Bergen, Norway, 1988.

[7] K. Karlsen, K. Brusdal, H. Dahle, S. Evje, and K.-A. Lie, "The corrected operator splitting approach applied to a nonlinear advection-diffusion problem," *Comput. Methods Appl. Mech. Engrg.*, vol. 167, no. 3-4, pp. 239–260, 1998.

[8] K. Karlsen and N. Risebro, "An operator splitting method for nonlinear convection-diffusion equations," *Numer. Math.*, vol. 77, no. 3, pp. 365–382, 1997.

[9] M. Espedal and R. Ewing, "Characteristic Petrov-Galerkin subdomain methods for two-phase immiscible flow," *Comput. Methods Appl. Mech. Engrg.*, vol. 64, no. 1-3, pp. 113–135, 1987.

[10] K. Karlsen and N. Risebro, "Corrected operator splitting for nonlinear parabolic equations," *SIAM J. Num. Anal.*, vol. 37, no. 3, pp. 980–1003 (electronic), 2000.

[11] O. Oleinik, "Discontinuous solutions of non-linear differential equations.," *Amer. Math. Soc. Transl. Ser. 2*, vol. 26, pp. 95–172, 1963.

[12] N. Risebro, "A front-tracking alternative to the random choice method," *Proc. Amer. Math. Soc.*, vol. 117, no. 4, pp. 1125–1139, 1993.

[13] N. Risebro and A. Tveito, "Front tracking applied to a non-strictly hyperbolic system of conservation laws," *SIAM J. Sci. Statist. Comput.*, vol. 12, no. 6, pp. 1401–1419, 1991.

[14] N. Risebro and A. Tveito, "A front tracking method for conservation laws in one dimension," *J. Comput. Phys.*, vol. 101, no. 1, pp. 130–139, 1992.

[15] J. Strikwerda, *Finite difference schemes and partial differential equations*. Pacific Grove, CA: Wadsworth & Brooks/Cole Advanced Books & Software, 1989.

[16] D. Hoff and J. Smoller, "Error bounds for finite-difference approximations for a class of nonlinear parabolic systems," *Math. Comp.*, vol. 45, no. 171, pp. 35–49, 1985.

[17] A. Datta-Gupta and M. King, "A semianalytic approach to tracer flow modeling in heterogeneous permeable media," *Adv. in Wat. Res.*, vol. 18, pp. 9–24, 1995.

[18] T. A. Hewett and T. Yamada, "Theory for the semi-analytical calculation of oil recovery and effective relative permeabilities using streamtubes," *Adv. in Wat. Res.*, vol. 20, pp. 279–292, 1997.

[19] M. King and A. Datta-Gupta, "Streamline simulation: A current perspective," *In Situ (Special Issue on Reservoir Simulation)*, vol. 22, no. 1, pp. 99–139, 1998.

[20] G. Chavent and J. Jaffre, *Mathematical models and finite elements for reservoir simulation*, vol. 17 of *Studies in mathematics and it's applications*. North Holland, Amsterdam, 1986.

[21] Z. Chen and R. Ewing, "Comparison of various formulations of three-phase flow in porous media," *Journal of Computational Physics*, vol. 132, pp. 362–373, 1997.

[22] H. Dahle, M. Espedal, and O. Sævareid, "Characteristic, local grid refinement techniques for reservoir flow problems," *Int. J. for Numerical Methods in Enginering*, vol. 34, no. 3, pp. 1051–1069, 1992.

[23] H. Holden, L. Holden, and R. Høegh-Krohn, "A numerical method for first order nonlinear scalar conservation laws in one dimension.," *Comp. Math. Appl.*, vol. 15, pp. 595–602, 1988.

[24] T. Gimse, "A triangular riemann solver.," Master's thesis, Dept. of Mathematics, Univ. of Oslo, Norway, 1988.

[25] T. Gimse, "A numerical method for a system of equations modelling one-dimensional three-phase flow in a porous medium," in *Nonlinear hyperbolic equations—theory, computation methods, and applications (Aachen, 1988)*, pp. 159–168, Braunschweig: Vieweg, 1989.

[26] L. Holden and R. Høegh-Krohn, "A class of n nonlinear hyperbolic conservation laws.," *Journal of Differential Equations*, vol. 84, pp. 73–99, 1990.

[27] H. Nordhaug, "On solving a three phase flow model with capillary forces," Master's thesis, Dept. of Mathematics, Univ. of Bergen, Norway, 1998.

[28] P. Reeves, *The Development of Pore-scale Network Models for the Simulation of Capillary Pressure - Saturation - Interfacial Area - Relative Permeability Relationships in Multi-fluid Porous Media*. PhD thesis, Department of Civil Engineering and Operations Research, Princeton University, 1997.

[29] P. Reeves and M. Celia, "A functional relationship between capillary pressure, saturation, and interfacial area as revealed by a pore-scale network model," *Water Resources Research*, vol. 32, no. 8, pp. 2345–2358, 1996.

[30] R. Held and M. Celia, "Modeling support of functional relationships between capillary pressure, saturation, interfacial area, and common lines," *Advances in Water Resources*, vol. 24, no. 3-4, pp. 325–343, 2001.

[31] F. Dullien, *Porous Media: Fluid Transport and Pore Structure*. Academic Press, 2 ed., 1992.

[32] L. Ferrand and M. Celia, "The effect of heterogeneity on the drainage capillary pressure - saturation relation," *Water Resources Research*, vol. 28, no. 3, pp. 859–870, 1992.

[33] M. Blunt and P. King, "Relative permeabilities from 2-dimensional and 3-dimensional pore-scale network modeling," *Transport in porous media*, vol. 6, no. 4, pp. 407–433, 1991.

[34] M. Valavanides and A. Payatakes, "True-to-mechanism model of steady-state two-phase flow in porous media, using decomposition into prototype flows," *Advances in Water Resources*, vol. 24, no. 3-4, pp. 385–407, 2001.

[35] K. Mogensen and E. Stenby, "A dynamic two-phase pore-scale model for imbibition," *Transport in Porous Media*, vol. 32, pp. 299–327, 1998.

[36] E. Aker, K. Maloy, A. Hansen, and G. Batrouni, "A two-dimensional network simulator for two-phase flow in porous media," *Transport in Porous Media*, vol. 32, pp. 163–186, 1998.

[37] T. Dijkstra, G. Bartelds, J. Bruining, and S. Hassanizadeh, "Dynamic pore-scale network for two-phase flow," in *Characterization and Measurement of the Hydraulic Properties of Unsaturated Soils* (van Genuchten et al., ed.), pp. 63–69, 1999.

[38] H. Dahle and M. Celia, "A dynamic network model for two-phase immiscible flow," *Computational Geosciences*, vol. 3, pp. 1–22, 1999.

Part II

Published and submitted work

Paper A

Operator splitting methods for systems of convection-diffusion equations: Nonlinear error mechanisms and correction strategies *

* To appear in Journal of Computational Physics.

OPERATOR SPLITTING METHODS FOR SYSTEMS OF CONVECTION-DIFFUSION EQUATIONS: NONLINEAR ERROR MECHANISMS AND CORRECTION STRATEGIES

K. HVISTENDAHL KARLSEN[A], K.–A. LIE[B,C],
J. R. NATVIG[C], H. F. NORDHAUG[A], AND H. K. DAHLE[A]

ABSTRACT. Many numerical methods for systems of convection-diffusion equations are based upon an operator splitting formulation, where convective and diffusive forces are accounted for in separate substeps. We describe the nonlinear mechanism of the splitting error in such numerical methods in the one-dimensional case, a mechanism that is intimately linked to the local linearizations introduced implicitly in the (hyperbolic) convection steps by the use of an entropy condition. For convection-dominated flows, we demonstrate that operator splitting methods typically generate a numerical widening of viscous fronts, unless the splitting step is of the same magnitude as the diffusion scale. To compensate for the potentially damaging splitting error, we propose a corrected operator splitting (COS) method for general systems of convection-diffusion equations with the ability of correctly resolving the nonlinear balance between the convective and diffusive forces. In particular, COS produces viscous shocks with correct structure also when the splitting step is large. A front tracking method for systems of conservation laws, which in turn relies heavily on a Riemann solver, constitutes an important part of our COS strategy. The proposed COS method is successfully applied to a system modelling two-phase, multicomponent flow in porous media and a triangular system modelling three-phase flow.

1. INTRODUCTION

Mathematical models for fluid flow often involve systems of convection-diffusion equations as a main ingredient. When a mathematical model is used for qualitative or quantitative studies, approximate numerical solutions must be constructed for the nonlinear system. An important design principle for many successful numerical methods for convection-diffusion equations is *operator splitting* (OS). That is, one splits the time evolution into partial steps to separate the effects of convection and diffusion. In particular, OS methods are often used to solve convection-diffusion problems that are of convection dominated nature, see [12] (and the references therein).

The motivation for operator splitting methods lies in that it is easy to combine modern methods developed within the hyperbolic community for tracking discontinuous solutions with efficient methods (e.g., multigrid) for solving implicit discretizations of the parabolic diffusive step, thus giving a powerful and efficient numerical method designed for resolving sharp gradients. From a software development point of view, this can be done in a stepwise (plug-and-play) manner, starting with a

Date: October 15, 2001.

Key words and phrases. systems of convection-diffusion equations, numerical methods, front tracking, finite difference method, operator splitting, splitting error, two-phase multi-component flow, three-phase flow.

[A]Department of Mathematics, University of Bergen, Johs. Brunsgt. 12, N–5008 Bergen, Norway.
E-mail: kenneth.karlsen@mi.uib.no, hans.nordhaug@mi.uib.no, helge.dahle@mi.uib.no.
[B]Department of Informatics, University of Oslo, P.O. Box 1080, Blindern, N-0316 Oslo, Norway.
[C]SINTEF Applied Mathematics, P.O. Box 124 Blindern, N-0314 Oslo, Norway.
E-mail: Knut-Andreas.Lie@math.sintef.no, Jostein.R.Natvig@math.sintef.no.

simple solver for each subproblem and then replacing each solver independently of the other by a more advanced solver until a suitable level of sophistication is reached.

The obvious disadvantage of operator splitting methods is the temporal splitting errors. Consider a scalar, convection-dominated problem $u_t + f(u)_x = \varepsilon u_{xx}$. If the equation possesses a viscous shock profile, this profile will appear on a spatial scale of order ε and move on a time scale of order $|f'(u)|$. Recent studies [2, 20, 21, 23, 24] have shown that unless the splitting step is of order ε, the temporal splitting error in OS methods can be significant in regions containing viscous shocks. The resulting incorrect balance between convective and diffusive forces appears as too wide shock layers in the numerical solution. Thus, to resolve viscous shock profiles correctly, one must resort to very small splitting steps, thereby imposing a time step restriction that is not present in the underlying numerical methods for the convective and the diffusive step. In fact, the splitting step needed to resolve the shock layers correctly may be much less than the one needed to resolve the interaction of viscous waves.

Small splitting steps should be avoided (if possible) for two reasons: computational efficiency and spatial accuracy. Increasing the number of splitting steps usually means increasing the runtime. Moreover, for numerical methods having stability restrictions, the highest *spatial* accuracy is often obtained when the time step is close to the stability limit. For these two reasons, one should try to pick the splitting step as large as possible. To reduce the influence of temporal splitting errors in OS methods and allow for the use of large splitting steps, the *corrected operator splitting* (COS) method was introduced by Karlsen and Risebro [23]. The COS method was further developed and successfully applied by Karlsen *et al.* in a series of papers [20, 21, 22, 13]. The forerunner for the scalar COS method was the modified method of characteristics for nonlinear scalar parabolic problems introduced by Espedal and Ewing [11] and Dahle [2], and further developed and analysed by Dahle, Espedal, and their collaborators [2, 3, 4, 5, 6] in the context of reservoir simulation. The relation between the modified method of characteristics and COS is discussed in the lecture notes [12].

The main idea behind the scalar COS method is to take into account the unphysical entropy loss (due to Oleinik's convexification) produced by the hyperbolic solver in the convective step. The COS approach uses the wave structure from the convective step to identify where the (nonlinear) splitting error occurs. This potential error is then compensated for in the diffusive step or in a separate correction step. As a result, the COS method exhibits the property of resolving accurately internal layers with steep gradients, gives very little numerical diffusion, and, at the same time, permits the use of large time steps. In addition, this numerical method seems to capture all potential combinations of convection and diffusion forces, ranging from convection dominated problems (including the purely hyperbolic case) to more diffusion dominated problems, all within the same application!

The purpose of this paper is to derive a thorough understanding of the nonlinear mechanisms behind the viscous splitting error typically appearing in operator splitting methods for systems of convection-diffusion equations. This mechanism is well understood in the scalar case. In Sections 2.1 to 2.3 we introduce in more detail the viscous splitting applied to one-dimensional systems, discuss the nonlinear mechanisms behind the temporal splitting error, and introduce a general correction strategy that generalizes the scalar COS algorithm [23]. Then in Section 2.4, we suggest particular numerical methods for solving the split problems introduced by the COS method and describe its numerical realisation in detail. In Section 3 we demonstrate the novel COS algorithm by applying it to two 2×2 systems of convection-diffusion equations modelling one-dimensional flow in porous media. We present a two-dimensional extension of the COS method based on dimensional splitting in Section 4. Finally, we make some concluding remarks in Section 5.

2. Operator Splitting Methods for General Systems

To describe of our ideas in more detail, we consider the Cauchy problem for $\ell \times \ell$ ($\ell \geq 1$) systems of convection-diffusion equations

$$\partial_t U + \partial_x F(U) = D\,\partial_x^2 U, \qquad U(x,0) = U_0(x) \tag{1}$$

where $x \in \mathbb{R}$ and $t > 0$. Here $U = (u_1, \ldots, u_\ell)^{\mathrm{T}}$ is the unknown state vector, $F(U) = (f_1(U), \ldots, f_\ell(U))^{\mathrm{T}}$ is a vector-valued function of class C^2, and $D = \mathrm{diag}(\varepsilon_1, \ldots, \varepsilon_\ell) > 0$ is a constant diagonal matrix. The linear diffusion operator is chosen for simplicity; all principles introduces below also apply to more complicated nonlinear (degenerate) diffusion operators. We always assume that the initial function $U_0(x)$ is of bounded total variation, i.e., $U_0 \in BV$. For a class of systems of the type (1), existence (and uniqueness) of classical solutions was established by Hoff and Smoller [17] using a finite difference scheme.

2.1. **Semi-Discrete OS.** Let $\mathcal{S}_t$ denote the solution operator taking the initial data $V_0(x)$ to a weak solution at time t of the purely hyperbolic problem

$$\partial_t V + \partial_x F(V) = 0, \qquad V(x,t) = V_0(x), \tag{2}$$

i.e., we write $v(x,t) = \mathcal{S}_t v_0(x)$ for this weak solution. For strictly hyperbolic $\ell \times \ell$ systems with initial data having small total variation, global existence of weak solutions of was proved by Glimm [15]. For stability and uniqueness of weak solutions, we refer to the recent paper by Bressan, Liu, and Yang [1].

Next, let $\mathcal{H}_t$ denote the operator taking the initial data $W_0(x)$ to a weak solution at time t of the purely parabolic problem

$$\partial_t W = D\partial_x^2 W, \qquad W(x,t) = W_0(x), \tag{3}$$

i.e., we write $W(x,t) = \mathcal{H}_t W_0(x)$ for this solution. For initial data of bounded total variation, the weak solution of (3) is a classical solution for $t > 0$, with the initial data U_0 taken in the strong L^1 sense on compact sets. This can be easily seen from the representation formula for the solution of the linear heat equation.

In what follows, we consider a fixed final computing time $T > 0$. For simplicity we also choose a fixed splitting step $\Delta t > 0$ and an integer N_t such that $N_t \Delta t = T$. Then we define the semi-discrete OS algorithm by

$$U_{\Delta t}(\cdot, n\Delta t) := \left[\mathcal{H}_{\Delta t} \circ \mathcal{S}_{\Delta t}\right]^n U_0(\cdot), \qquad n = 0, \ldots, N_t. \tag{4}$$

In the scalar case, it can be proved that $U_{\Delta t}$ converges in L^1 on compact sets to the unique classical solution U of (1) as the splitting step Δt tends to zero. The convergence proof is based on a uniform BV (space) estimate for the splitting approximation and Helly's theorem, see Karlsen and Risebro [24] for details. In general, it is not possible to prove BV estimates for systems and thus the Helly framework cannot be used to prove convergence of operator splitting methods. However, it is possible to prove a Lax–Wendroff type theorem for splitting methods for systems of parabolic equations:

Theorem 2.1. *If the splitting method* (4) *converges to a limit $U(\cdot,t) \in BV$, then this limit is the unique classical solution of* (1).

The proof goes along the lines of [18]. The theorem shows that (4) can indeed be used as a basis for constructing numerical solution algorithms for (1).

In applications, the exact solution operators $\mathcal{S}_t$ and $\mathcal{H}_t$ in (4) are replaced by numerical methods. We will use front tracking as defined by Risebro [27, 28, 29] as an approximate solution operator for the hyperbolic part. For the parabolic part, we here use a simple explicit central difference method.

The methods will be introduced in more detail in the next section. Meanwhile, let us stress that the method for the diffusion part is deliberately chosen as simple as possible to focus on the main ideas of our COS strategy and generally should be replaced by a more sophisticated method. We mention that Dawson, Wheeler, and collaborators [7, 8, 9, 10, 32] are using operator splitting algorithms similar to (4). In their splitting algorithm, the hyperbolic problem (2) is solved by $M \geq 1$ local time steps (for each splitting step) with an explicit Godunov type method, while the diffusion equation (3) is solved implicitly.

2.2. **Nonlinear Error Mechanisms.** In the introduction we stated that OS approximations can be too diffusive near viscous shocks when the splitting step Δt is large. Karlsen and Risebro [23] point out that this splitting error is simply a manifestation of the entropy condition that is imposed in the hyperbolic convection step. Let us consider the *scalar* case. The entropy condition introduces a local linearisation of $f(\cdot)$ once a shock is formed in the convection step and this linearisation represents the entropy loss associated with the formation of a shock in the hyperbolic solution. Thus the evolution of the *hyperbolic* solution is governed locally by some convex/concave envelope f_c of f between the left and right shock values. A similar linearisation can be introduced locally for the *parabolic* problem; that is, the flux function f can be decomposed into a convective part f_c and a self-sharpening part $f - f_c$ that tends to counteract the diffusive forces. Loosely speaking, we say that f_c governs the local translation and $f - f_c$ the shape (or structure) of the viscous front. In the OS algorithm, the local residual flux $f - f_c$ is disregarded in the hyperbolic step and the corresponding self-sharpening effects are therefore not taken into account in the splitting, resulting in a splitting error. Figure 1 gives an illustration of f, f_c, and the residual flux $f_{\text{res}} := f - f_c$ in the scalar case.

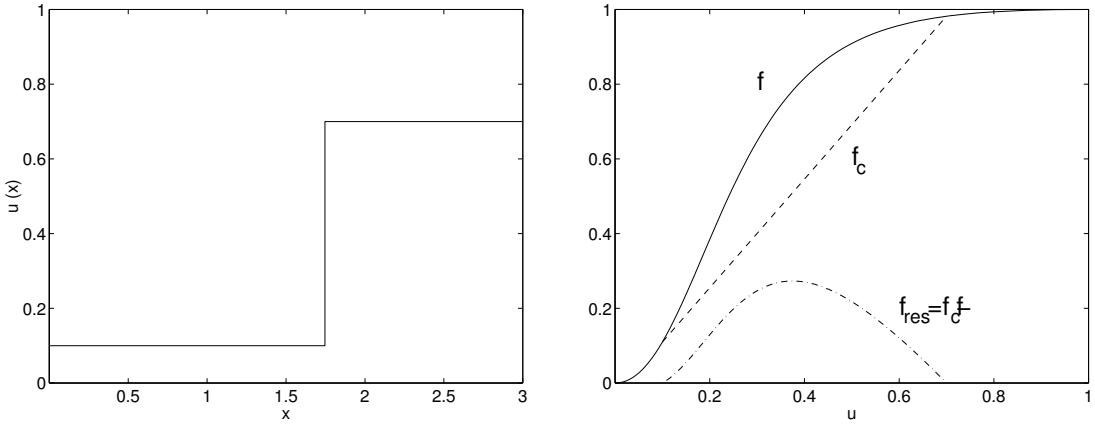


FIGURE 1. (Left) A single shock solution from a convection step. (Right) The corresponding residual flux function; flux function f (solid), convex envelope f_c, i.e., local linearisation (dash), and residual flux f_{res} (dash-dot).

For a general system, the error mechanism is quite similar. To study it, we consider the propagation of a single viscous shock. Assume that the splitting step is sufficiently large so that a shock has developed in the hyperbolic substep (2), i.e., the solution $V(\cdot, t = \bar{t})$ consists of a single discontinuity at $x = \bar{x}$ with left and right shock values $V^l = (v_1^l, \ldots, v_1^l)^{\mathrm{T}}$ and $V^r = (v_1^r, \ldots, v_1^r)^{\mathrm{T}}$. Then the behaviour (forward and backward in time) of $V(x,t)$ locally around $(\bar{x}, \bar{t})$ is governed by the linearized hyperbolic

problem

$$\partial_t V + \partial_x(\bar{\sigma} V) = 0, \qquad V(x,\bar{t}) = \begin{cases} V^l, & \text{for } x < \bar{x}, \\ V^r, & \text{for } x > \bar{x}, \end{cases} \tag{5}$$

where $\bar{\sigma}$ is the Rankine–Hugoniot shock speed satisfying

$$F(V^l) - F(V^r) = \bar{\sigma}(V^l - V^r).$$

We *claim* that a large part of the splitting error occurring locally around $(\bar{x},\bar{t})$ in the standard OS algorithm can be understood in terms of the difference between the nonlinear system in (1) and the linearized system in (5) with right-hand side $D\partial_x^2 V$, or in other words, in terms of the difference $\partial_x(F(U) - \bar{\sigma}U)$. In (1), the diffusion caused by the second order operator is perfectly balanced by the self-sharpening effects due to the nonlinearity in the convective operator. In the OS strategy, this self-sharpening disappears once a shock develops because $F(U)$ is in effect replaced by $\bar{\sigma}U$ locally. Thus, one step in OS effectively amounts to solving $\partial_t U + \partial_x(\bar{\sigma}U) = D\partial_x^2 U$ and not (1).

2.3. **A Novel COS Strategy.** To compensate for the loss of self-sharpening effects, the *scalar* COS approach proposes to include the residual flux F_{res} in the diffusion step of the splitting. The COS method therefore replaces the purely parabolic split problem (3) by

$$\partial_t W + \partial_x F_{\text{res}}(W;x) = D\partial_x^2 W, \qquad W(x,0) = W_0(x). \tag{6}$$

As an alternative, the residual flux can be included in a separate correction step, $\partial_t V + \partial_x F_{\text{res}}(V;x) = 0$, see [23, 21] for more details. Letting $\mathcal{P}_t$ denote the solution operator associated with (6), the COS solution may then be defined as

$$U_{\Delta t}(\cdot, n\Delta t) := \left[\mathcal{P}_{\Delta t} \circ \mathcal{S}_{\Delta t}\right]^n U_0(\cdot). \tag{7}$$

What we have done so far might seem a bit peculiar. We have taken the convection-diffusion equation (1) and replaced it by a hyperbolic equation (2) and another convection-diffusion equation (6), where the flux term in (6) is seemingly more complicated than the one in (1). However, we see that while F contains convective *and* self-sharpening effects, F_{res} only contributes self-sharpening effects. Thus, viscous shock fronts are moved to the correct location in the convective step and given a correct shape in the diffusive step. The solution process has also been simplified from a numerical point of view, i.e., with a fully discrete version of (7). Parabolic equations of the kind (1) and (6) will typically be solved by some implicit scheme, involving iterative solution of nonlinear systems of equations. The hyperbolic step can therefore be seen as some kind of preconditioner or a means for providing a good initial guess for the iteration, and the iteration process will converge faster for (6) than for (1). If the method used to solve the hyperbolic step is fast compared with the nonlinear iteration, we gain something in terms of efficiency, see, e.g., [4, 5, 11]. In the next subsection we introduce one such method, which employs a very fast, unconditionally stable, front tracking method for the convective step.

When applied to systems of parabolic equations, the correction algorithm needs to be reformulated, since one cannot simply write down the solution of the hyperbolic step in terms of convex/concave envelopes. Instead, we identify the following term

$$\partial_x F_{\text{res}}(U) = \partial_x\big(F(U) - \bar{\sigma}U\big), \tag{8}$$

for each discontinuity in the solution from the hyperbolic step. Then, the parabolic subproblem (3) is modified locally by adding $F_{\text{res}}(U)$, giving the new split problem (6). By integrating (8) with respect to x, we get the *residual flux*

$$F_{\text{res}}(U) = \big(F(U) - F(V^l)\big) - \bar{\sigma}\big(U - V^l\big), \tag{9}$$

where we have chosen the constant of integration such that

$$F_{\text{res}}(V^l) = F_{\text{res}}(V^r) \equiv 0.$$

2.4. **A Fully Discrete (C)OS Method.** The operator splitting methods introduced above result in two different subproblems that each must be solved numerically. Therefore, before we describe the OS and COS approach in more detail, we introduce numerical methods for solving the subproblems.

Convection solver. In this section we describe the front tracking method [27, 28, 29] for solving systems of conservation laws (2)

$$\partial_t V + \partial_x F(V) = 0, \quad V(x,0) = V_0(x).$$

The initial function $V_0(x)$ is assumed to be of bounded variation. Front tracking is an algorithm for computing a piecewise constant approximation to $V(x,t)$. First, V_0 is approximated by a step function so that a Riemann problem can be associated with each jump in the approximate initial data. The solution of each Riemann problem is approximated by a step function. This is most accurately achieved by using the Lax construction for the exact Riemann solution, utilizing the local system of coordinates formed by the wave curves around two constant states in state space. The Riemann solution consists of a set of constant states connected by simple waves. In the front tracking approximation, rarefaction waves are approximated by step functions sampled along the wave curves (according to a pre-set, user-defined parameter δ), while the rest of the Riemann solution is left intact. This way, each Riemann problem produces a sequence of jump discontinuities (fronts) that travel with a finite wave speed. The Riemann solution is represented by a list of fronts sorted according to increasing waves speed.

A global solution (in x) is formed by connecting the local Riemann solutions and consists of constant states separated by space-time rays, i.e., a list of fronts sorted from left to right. There will be a first time at which two or more space-time rays intersect, i.e., two or more fronts collide. This collision defines a new Riemann problem with states given by the rightmost and leftmost of the colliding fronts. The colliding fronts are taken out of the solution. Then the Riemann problem is solved and the emerging fronts are inserted into the front list. This way, the algorithm proceeds from collision to collision. Notice that no computations are necessary between collision times. To reduce the number of wave interactions, it is customary to perform some data reduction, i.e., remove weak waves, see [29]. The numerical method is *unconditionally* stable and very fast.

Diffusion solver. The parabolic step is a Cauchy problem of the form

$$\partial_t W + \partial_x G(W) = \partial_x^2 D(W), \quad W(x,0) = W_0(x), \tag{10}$$

where G is in applications the residual flux term (see (8)). The initial function $W_0(x)$ is assumed to be of bounded variation. To approximate the solution $W(x,t)$, we introduce a mesh in the (x,t) plane where the spatial grid points are denoted by x_j and the time levels by t_n. We denote the spacing in the x and t variables by Δx and τ, respectively; i.e., $(x_j,t_n) = (j\Delta x, n\tau)$. The value of the difference approximation at (x_j,t_n) is denoted by W_j^n. To solve this system, one can for instance use the explicit, central finite difference method

$$\frac{W_j^{n+1} - W_j^n}{\tau} - \frac{G(W_{j+1}^n) - G(W_{j-1}^n)}{2\Delta x} = \varepsilon \frac{D(W_{j+1}^n) - 2D(W_j^n) + D(W_{j-1}^n)}{(\Delta x)^2}. \tag{11}$$

This scheme is stable provided the discretization parameters τ and Δx satisfy the following conditions

$$\tau \le 0.5\Delta x^2/\varepsilon, \qquad \Delta x \max|\lambda_G| \le 2\varepsilon,$$

where λ_G denotes the eigenvalues of G'. In the case when $D(W) = W$, i.e., linear diffusion, convergence and error estimates for this scheme are shown in [17].

The stability conditions above may put severe restrictions on the discretization parameters, especially on Δx for small values of ε. However, both these conditions can be weakened or removed by using a more sophisticated scheme. The second condition on Δx is removed by using an upwind discretization of the flux G, and the first condition is weakened by using an implicit scheme. Generally, the most efficient method would therefore be an implicit discretization combined with an efficient (non)linear solver. In this context, the important point to keep in mind is that (10) (with G equal to the residual flux term) is much closer to being self adjoint than the original equation, since the front tracking will give almost "exact" information about the hyperbolic structure of the problem (see also the discussion after (7)). This means that any iterative procedure will be much more efficient for (10) than (1), and the numerical approximation properties will be better [11, 2, 3, 4, 5, 6]. However, to keep the technical details at a minimal level, we here choose simple explicit schemes. In Section 3 we use the scheme in (11), while in Section 4 we replace the discretization of the G-term by a suitable upwind difference to avoid the restriction on the grid size.

The splitting method. Given numerical methods for the two substeps, we can now describe the (C)OS method for the convection-diffusion problem (1). The construction of the residual flux F_{res} is described in detail later on.

Letting $\Delta x > 0$ denotes the grid spacing, we introduce the grid $\{x_j = j\Delta x\}_{j\in\mathbb{Z}}$. Our approximate solutions will be piecewise constant functions with respect to the grid cells $\{I_j = [x_j, x_{j+1})\}_{j\in\mathbb{Z}}$. Therefore we need the scalar projection operator π defined by

$$\pi z(x) = \frac{1}{\Delta x}\int_{I_j} z(\xi)\,dx, \qquad \text{for all } x \in I_j, \tag{12}$$

for any scalar function $z(x) \in BV$, and the operator Π defined by $\Pi Z = (\pi z_1, \dots, \pi z_\varrho)^{\mathrm{T}}$, for any vector-valued function $Z = (z_1, \dots, z_\varrho)^{\mathrm{T}} \in BV$. We next consider a fixed final computing time $T > 0$, and choose a splitting step $\Delta t > 0$ and an integer N_t such that $N_t\Delta t = T$. Using Δ as a short-hand notation for the discretization parameters $(\Delta x, \Delta t, \delta)$, we define our piecewise constant COS approximation $U_\Delta : \mathbb{R}\times[0,T] \to \mathbb{R}$ by

$$U_\Delta(x,t) := U^n(x), \qquad (x,t) \in \mathbb{R}\times((n-1)\Delta t, n\Delta t], \quad n = 1, \dots, N_t, \tag{13}$$

where $U_\Delta(x,0) := U^0(x)$ and $U^0 := \Pi U_0$. For notational convenience, we have suppressed the dependency on Δ in U^{n+1}. We next explain how to inductively construct the piecewise constant function $U^{n+1}(x)$ from the knowledge of the piecewise constant function $U^n(x)$. The construction consists of two main steps:

Step 1 (Convection): Let $\mathcal{S}_t V_0$ be the front tracking solution of the hyperbolic problem

$$\partial_t V + \partial_x F(V) = 0, \quad V(x,0) = V_0(x).$$

where we have suppressed the dependency on the discretization parameters Δx and δ in $\mathcal{S}_t$. We then define the intermediate solution

$$U^{n+1/2} = \mathcal{S}_{\Delta t} U^n.$$

Step 2 (Diffusion): Introduce a local time step τ satisfying $\tau \le 0.5\Delta x^2/\varepsilon$. Furthermore, we let N_τ be an integer such that $N_\tau\tau = \Delta t$. Let $\mathcal{P}_t W_0$ be the finite difference solution of

$$\partial_t W + \partial_x F_{\text{res}}(x, W) = D\partial_x^2 W, \quad W(x,0) = \Pi W_0(x),$$

where we have suppressed the dependency on the discretization parameters Δx and τ in $\mathcal{P}_t$. The residual flux $F_{\text{res}}(x,\cdot)$ depends on the hyperbolic solution $U^{n+1/2}$ and its construction is explained in detail below. Finally, the COS solution U^{n+1} is defined as

$$U^{n+1} = \mathcal{P}_{\Delta t} \circ \Pi U^{n+1/2},$$

or alternatively as

$$U^{n+1} = \Big[\mathcal{P}_{\Delta t} \circ \Pi \circ \mathcal{S}_{\Delta t}\Big] U^n.$$

Note that if the residual flux F_{res} is set to zero, the above algorithm reduces to a standard OS algorithm. In Section 3, we demonstrate numerically that the function U_Δ is a good approximation to the exact solution U of the convection-diffusion problem (1). In particular, when Δt is large, the COS method gives significantly more accurate treatment of viscous shocks than the corresponding OS method.

Remark. One should not confuse the splitting step Δt, which is also the time step used by the hyperbolic solver (i.e., the front tracking method), with the time step τ used by the parabolic solver. Since the parabolic step is here solved (for simplicity) by an explicit method, we have a stability constraint on τ. However, there is no CFL-constraint on the splitting step Δt! Since the front tracking method is unconditionally stable, we can therefore take $\Delta t = M\tau$ with $M >> 1$, and this is indeed what we do in practice.

Remark. Although the exact solution of (1) is a smooth function, our approximation $U_\Delta(\cdot,t)$ is merely piecewise constant. Increased accuracy in space can be obtained by replacing $U_\Delta(\cdot,t)$ with a piecewise linear interpolant.

Construction of the residual flux. Given a piecewise constant, front tracking solution $U^{n+1/2}$, we can now construct the residual flux $F_{\text{res}}(x,\cdot)$ appearing in (6). We assume that the discontinuities of $U^{n+1/2}(x)$ are located at the points $\{x_i\}$. Let $U_i = (u_1^i,\dots,u_\varrho^i)^{\mathrm{T}}$ and $U_{i+1} = (u_1^{i+1},\dots,u_\varrho^{i+1})^{\mathrm{T}}$ denote the values of $U^{n+1/2}(x)$ in the intervals $[x_{i-1},x_i)$ and $[x_i,x_{i+1})$, respectively. Locally around the ith discontinuity emerging from (x_i,t_0) the nonlinear problem (2) is governed by the linearized problem

$$\partial_t V + \partial_x(\sigma_i V) = 0, \qquad V(x_i,t_0) = \begin{cases} U_i, & \text{for } x < x_i, \\ U_{i+1}, & \text{for } x > x_i, \end{cases}$$

where σ_i is the Rankine–Hugoniot shock speed satisfying $F(U_i) - F(U_{i+1}) = \sigma_i(U_i - U_{i+1})$. Motivated by the discussion in Section 2.2, we define the residual flux F_{res}^i associated with the ith discontinuity as

$$F_{\text{res}}^i(U) = \begin{cases} \big(F(U) - F(U_i)\big) - \sigma_i\big(U - U_i\big), & U \in (u_1^i,u_1^{i+1}) \times \cdots \times (u_\varrho^i,u_\varrho^{i+1}), \\ 0, & \text{otherwise.} \end{cases}$$

Note that $F_{\text{res}}^i(U_i) = F_{\text{res}}^i(U_{i+1}) \equiv 0$.

We note that this is the same residual flux as Karlsen and Risebro [23] presented, since the shock speed σ_i has to fulfill the Rankine–Hugoniot relation. We also note that the constant of integration is of no importance since only derivatives of F_{res} are present in the COS equations.

Although a residual flux term can be identified for every discontinuity in the front tracking solution, they should not be included for discontinuities approximating rarefaction waves or for weak shocks. In practice we therefore only include residual terms for shock waves with strength exceeding a user-defined threshold γ. The process of identifying relevant residuals can be simplified by tagging fronts in the front tracker according to wave type (shock/rarefaction/contact).

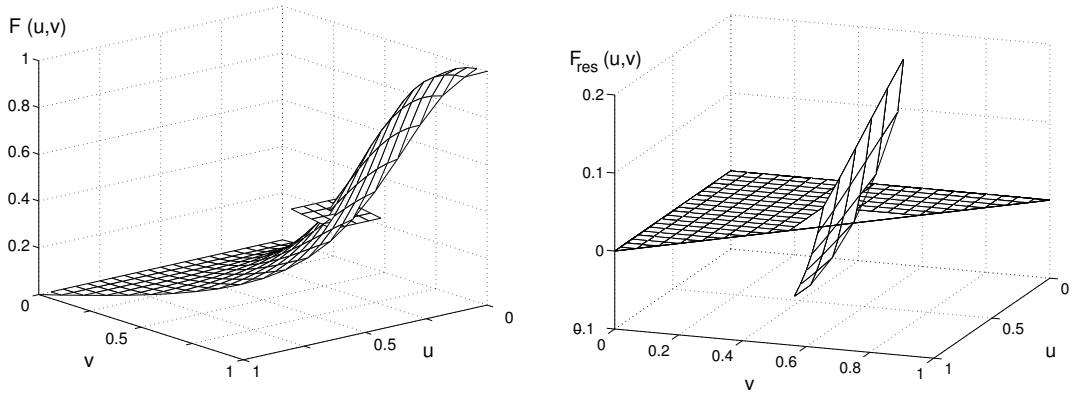


FIGURE 2. (Left) Shock plane and one component of the flux for a system of conservation laws. (Right) The corresponding residual flux component.

Having defined the residual fluxes in state space $(u_1, \ldots, u_\varrho)$, we need to specify where to apply them in physical space (i.e., intervals in x). This can be done in several ways. For *explicit* discretizations we apply the following strategy: We observe that in each spatial interval where the solution is monotone in all its components (henceforth called *monotonicity interval*), all residual fluxes are defined on disjoint sets in state space. Therefore, we set the residual flux to zero outside (a subset of) the associated monotonicity interval, i.e.,

$$F_{\text{res}}(x, U) = \sum_i F^i_{\text{res}}(U)\chi_{D_i}(x),$$

where $\chi_I(x)$ denotes the indicator function of the interval $I \subset \mathbb{R}$ and D_i is the (subset of) the monotonicity interval. Using the monotonicity of the solution, we can determine a unique residual flux also in regions of changing monotonicity. Although the monotonicity intervals may change throughout the diffusion step (as the discontinuity is smoothed out), they are always well-defined and easy to compute.

This approach works well for explicit schemes, but it does not apply to implicit discretizations when the discontinuity coincides with a change in monotonicity, i.e., when the left or right state of the discontinuity is a local extremum in one of its components. The reader is referred to Figure 15 for an example where such a situation occurs. For implicit discretizations we therefore use a much simpler approach where the user prescribes the length of the intervals where the correction is applied. To avoid overlap of residual flux domains, it might be necessary to clip the spatial correction intervals at each endpoint. To this end, one can use monotonicity information and for instance specify that the intervals should not exceed the midpoint between the discontinuities of two consecutive residual fluxes (i.e., not exceed $(x_i + x_{i+1})/2$).

Unfortunately, specifying a reasonable length for the correction intervals must be based on experience. For explicit discretizations, we have observed that the corresponding 'internal boundaries' introduced in the diffusion step may lead to unphysical effects in certain cases if for instance the length of the interval is underestimated by the user; one example is reflections breaking up the monotonicity of the viscous front. For explicit schemes we therefore advocate the approach based upon monotonicity intervals and for implicit schemes the approach based upon specified interval lengths. However, to show that both approaches work for explicit schemes, we construct residual fluxes based upon monotonicity intervals in Section 3.1 and based upon specified length in Section 3.2.

Remark. For scalar conservation laws, $f_{\text{res}}(\cdot, u)$ may be discontinuous for each fixed u, while $f_{\text{res}}(x, \cdot)$ is always continuous for each fixed x, see Figure 1. This is, however, not the case for systems of conservation laws. Each component $f^i_{\text{res}}(x, U)$ of $F_{\text{res}}(x, U)$ may be discontinuous for each fixed U and, more important, is always discontinuous for each fixed x, expect for the trivial case with no shock, see Figure 2 below.

3. Applications

In this section we present numerical realizations of the novel COS strategy for two particular 2×2 systems describing flow in porous media. Simulations for a two-phase, multicomponent model are reported in Section 3.1 and similar experiments for a triangular, three-phase model in Section 3.2.

3.1. **The Polymer System.**

We consider the initial value problem for the following 2×2 system of parabolic equations:

$$\begin{aligned} \partial_t s + \partial_x f(s,c) &= \varepsilon \partial_x^2 s \\ \partial_t [sc + a(c)] + \partial_x (c f(s,c)) &= \varepsilon \partial_x^2 [sc + a(c)], \end{aligned} \tag{14}$$

where (s, c) is the unknown state vector, $f = f(s, c)$, $a = a(c)$ are given functions

$$f(s,c) = \frac{s^2}{s^2 + \mu(1+\nu c)(1-s)^2}, \quad a(c) = \frac{0.2c}{1+c}, \tag{15}$$

and $\varepsilon > 0$ is a small scaling parameter. For all numerical examples in this section, $\mu = 0.5$ and $\nu = 2$.

The system (14) models a polymer process in enhanced oil recovery, see [30] and references therein for details. Existence, uniqueness and stability properties for a smooth solution of the Cauchy problem for (14) have been established by Tveito [30]. The Riemann problem for the corresponding inviscid, nonstrictly hyperbolic system is solved in [19] and a front tracking method is presented in [28]. One special feature of the inviscid system is that the eigenvalues $\lambda_s = f_s$ and $\lambda_c = f/(s + a_c)$ coincide along a curve $\mathcal{T}$ in state space.

Introducing the quantity $b = sc + a(c)$, the mathematical model (14) takes the form

$$\begin{aligned} \partial_t s + \partial_x f(s,c) &= \varepsilon \partial_x^2 s \\ \partial_t b + \partial_x (c f(s,c)) &= \varepsilon \partial_x^2 b. \end{aligned} \tag{16}$$

The Riemann problem. The solution of the Riemann problem for (16) (with $\varepsilon = 0$) is a composition of four simple waves; s- and c-rarefactions and s- and c-shocks. For Riemann problems with no jump in the concentration c, the solution reduces to that of the Buckley–Leverett equation, which is well known. Solutions of this kind will simply be termed s-waves, although they may be compositions of shocks and rarefactions.

For Riemann problems with a jump in c, the system behaviour is more complicated since the eigenvalues of (16) coincide along a curve $\mathcal{T}$ in phase space. In general, if $c^L > c^R$ the solution will consist of one c-shock and possibly some s-waves. If $c^L < c^R$, the solution will be made up of s-waves an c-rarefactions.

For data completely on one side of $\mathcal{T}$, the solution is a composition of one s-wave and one c-wave. However, when the left and right states $\mathbf{u}^L$ and $\mathbf{u}^R$ are on different sides of $\mathcal{T}$, the solution can consist of up to five (or six) constant states, separated by simple waves. This case is shown in Example 2 of this section.

Due to the nonstrict hyperbolicity, there is one special case, called an overcompressive shock, where both the s- and c-characteristics go into the shock. The corresponding solution can be interpreted as the exact superposition of an s- and a c-shock, see Example 3 of this section.

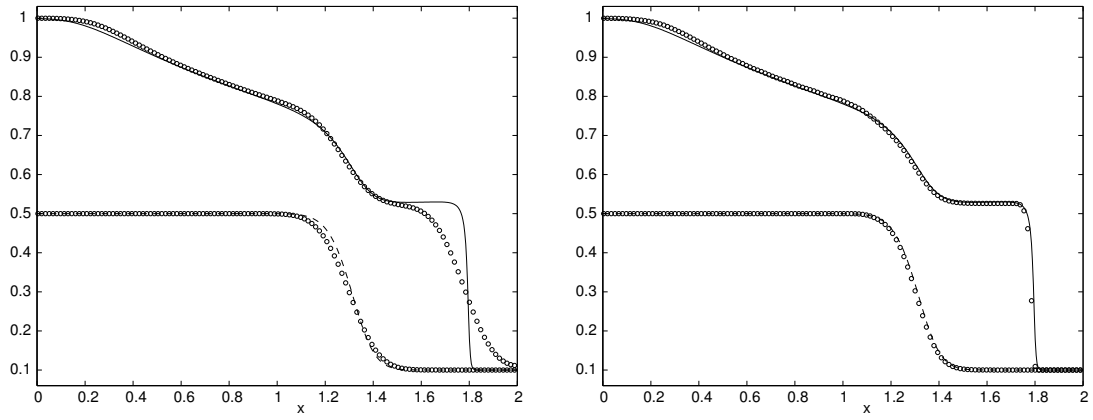


FIGURE 3. Example 1. Solution at time $t = 1.0$ computed using one OS step (left) and one COS step (right). The solution is plotted in primitive variables (s, c) at every second grid point. A reference solution computed by the central difference scheme (11) is included; the solid lines gives the s-component and the dashed line the c-component.

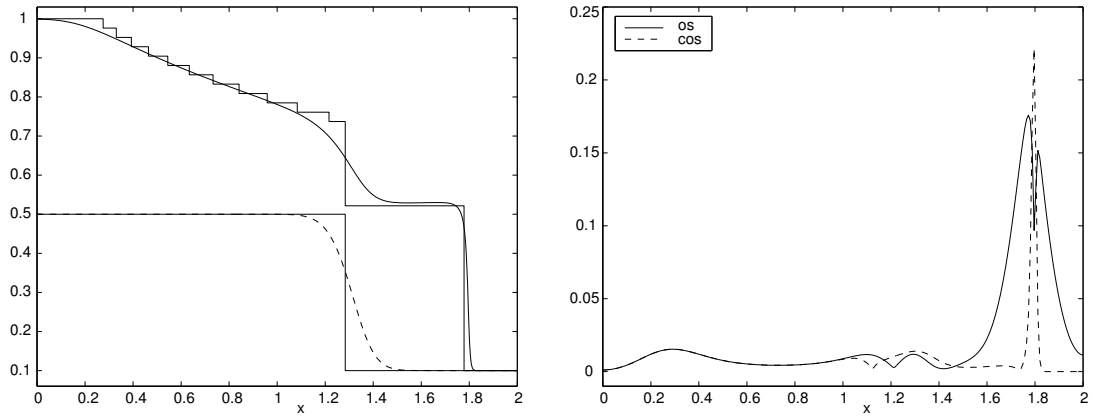


FIGURE 4. Example 1. (Left) The solution after the hyperbolic substep compared with the reference solution. (Right) Pointwise error in L^∞ norm relative to the reference solution.

Example 1. In our first example, we consider discontinuous initial data

$$(s_0, c_0)(x) = \begin{cases} (1.0, 0.5), & x \leq 0.1, \\ (0.1, 0.1), & x > 0.1. \end{cases}$$

In the inviscid case, the initial data correspond to a Riemann problem, which is solved by an s-shock, followed by a c-shock and an s-rarefaction wave.

Figure 3 shows the approximate solution at time $t = 1.0$ for $\varepsilon = 0.005$ computed using OS and COS with one time step on a 256 grid. Since the time step is much larger than the diffusion scale, OS gives a considerable smearing of the s-shock, whereas the c-shock contains little self-sharpening and is therefore represented quite accurately. When correction effects are included, both shock fronts are resolved almost perfectly. In this example, residual fluxes are defined for both the s-shock and the c-shock (see Figure 4), although the latter gives little effect. No fluxes are defined for the discontinuities in the rarefaction wave.

Figure 4 reports pointwise L^∞ errors for OS and COS. As expected, the major error contribution is around the s-shock. However, both OS and COS overshoot the s-component in the interval from $x = 0.1$ to $x = 0.5$. Because of mass conservation, the s-shock is therefore placed a bit to the left compared with the reference solution; hence the high peak in the pointwise error for COS.

To study the convergence of the splitting methods, we fix the spatial discretization to 2^{10} blocks on the interval $(-0.25, 2.25)$ and increase the number of splitting steps by powers of two. Errors for this convergence study are reported in Table 1 for $\varepsilon = 0.01$ and $\varepsilon = 0.001$. Here the error is defined as

$$E = \sum_{i=1}^{\varrho} \frac{\|u_i - u_i^r\|_1}{\|u_i^r\|_1}, \tag{17}$$

where $(u_1, \dots, u_\varrho)$ denotes the splitting solution and $(u_1^r, \dots, u_\varrho^r)$ the reference solution.

TABLE 1. *Example 1. Estimated errors* (17) *and convergence rates for OS and COS with fixed spatial discretization; N_t denotes the number of splitting steps.*

	ε=0.01				ε=0.001			
N_t	OS	rate	COS	rate	OS	rate	COS	rate
1	4.42e-02	—	1.96e-02	—	1.88e-02	—	3.47e-03	—
2	2.90e-02	0.61	1.61e-02	0.29	1.26e-02	0.57	5.18e-03	-0.58
4	1.97e-02	0.56	1.22e-02	0.40	8.16e-03	0.63	4.30e-03	0.27
8	1.27e-02	0.63	8.53e-03	0.51	5.69e-03	0.52	3.27e-03	0.40
16	7.57e-03	0.75	5.65e-03	0.59	4.18e-03	0.44	2.52e-03	0.38
32	4.16e-03	0.86	3.58e-03	0.66	3.27e-03	0.35	2.22e-03	0.18
64	2.30e-03	0.86	2.45e-03	0.55	2.78e-03	0.23	2.17e-03	0.03
128	1.48e-03	0.63	1.53e-03	0.68	2.85e-03	-0.03	2.58e-03	-0.25
256	1.30e-03	0.19	1.31e-03	0.23	3.61e-03	-0.34	3.52e-03	-0.45

For $\varepsilon = 0.001$, the error for OS decreases up to $N_t = 64$, but then starts to increase as a result of increasing numerical diffusion introduced by the projection in the hyperbolic steps. For COS, the error increases when going from one to two splitting steps, because with two splitting steps the c-shock is not fully formed in the second hyperbolic step, and hence the residual flux is weaker. With four (or more) splitting steps, a residual flux for the c-shock is only formed in the first step.

For $\varepsilon = 0.01$, the OS error decreases for all N_t (but it increases for $N_t = 512$). By introducing corrections, we remove most of the splitting error around the s-shock, but not in the interval from $x = 0$ to $x = 1.5$, see Figure 5. Therefore, the COS error also decreases with increasing number of splitting steps. Since the c-shock is completely smoothed out by the diffusive forces, the corresponding residual flux is formed only in the first hyperbolic step.

Example 2. The next example demonstrates that the corrected splitting method also applies to non-monotone data. To this end, consider the Riemann initial data

$$(s_0, c_0)(x) = \begin{cases} (0.45, 0.0), & x \le 0.1, \\ (0.2, 1.0), & x > 0.1. \end{cases}$$

In the inviscid case, this problem gives the maximum number of intermediate states in the Riemann solution, i.e., a solution of the form $u^L \xrightarrow{c} u^1 \xrightarrow{s} u^2 \xrightarrow{c} u^3 \xrightarrow{s} u^R$, where $\xrightarrow{s}$ denotes an s-wave and $\xrightarrow{c}$ a c-wave, see e.g., [19, Ex. 8.2]. The solutions in the inviscid case and for $\varepsilon = 0.0025$ are shown in Figure 6.

Figure 7 shows approximate solutions computed by OS and COS on a 2^{10} grid; estimated L^1 errors in each grid cell are shown in Figure 8. With one splitting step, COS resolves the leading s-wave

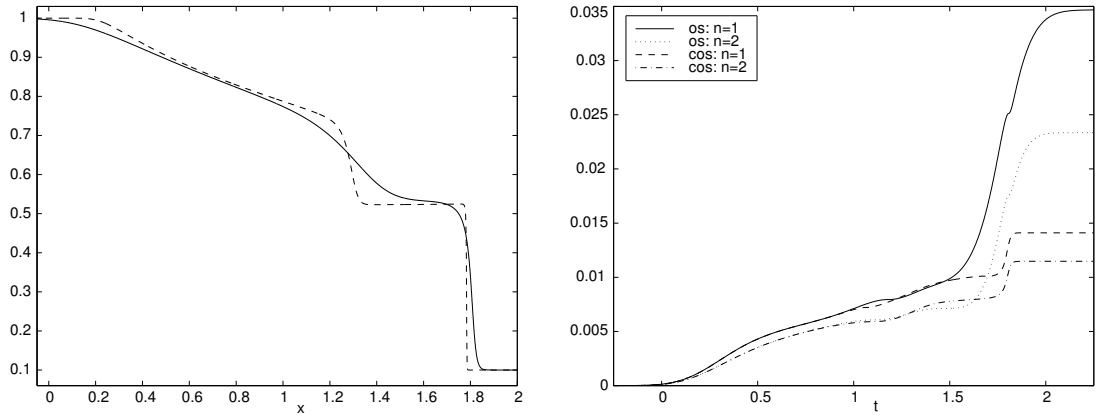


FIGURE 5. Example 1. (Left) The s-component of the reference solution for $\epsilon = 0.01$ (solid) and 0.001 (dashed). (Right) Cumulative L^1 error of the s-component for OS and COS with one and two splitting steps for $\epsilon = 0.01$.

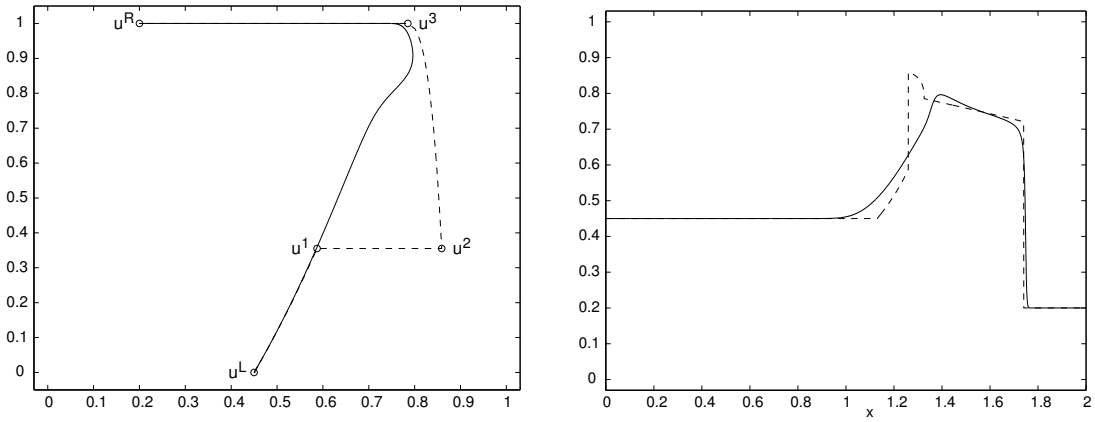


FIGURE 6. Example 2. (Left) The solution in (s, c)-space; solid line represents the solution for $\epsilon = 0.0025$ at time $t = 1.0$ and the dashed line the inviscid solution. (Right) The s-component as a function of spatial coordinate x.

accurately. However, since a residual flux is also identified for the s-shock at $x \approx 1.26$, COS produces an extra peak in the solution (resembling the inviscid solution). With two splitting steps, the OS and COS solutions coincide along the trailing edge. With four splitting steps, COS gives a fairly accurate approximation of the exact solution. Notice that four splitting steps corresponds to a CFL number around 300!

Example 3. In the last polymer example, we consider a Riemann problem corresponding to an (over)compressive shock $u^L \xrightarrow{c} u^R$ in the inviscid case; compressive shock means that both the s- and the c-characteristics go into the shock and contribute to the self-sharpening.

The Riemann problem is given by

$$(s_0, c_0)(x) = \begin{cases} (0.75, 0.8), & x \le 0.25, \\ (0.839619\ldots, 0.4), & x > 0.25. \end{cases}$$

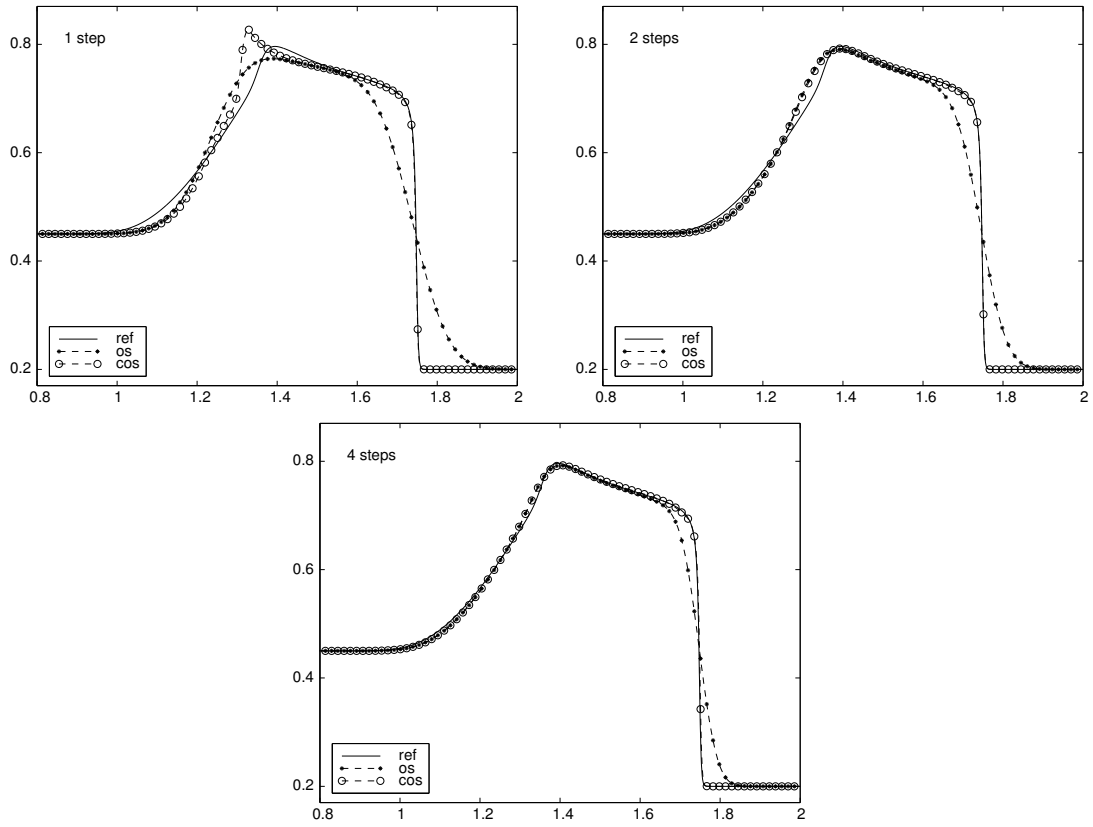


FIGURE 7. Example 2. Comparison of OS (small dots) and COS (large dots) with one, two, and four splitting steps. The plots show the s-component versus x at time $t = 0.5$ with markers at every eight grid point.

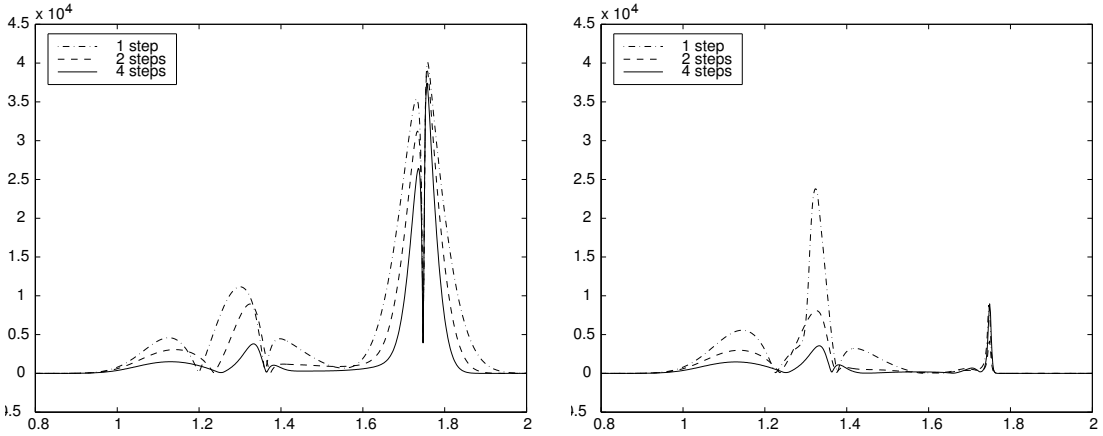


FIGURE 8. Example 2. L^1 error versus the cell center in each grid cell for OS (left) and COS (right).

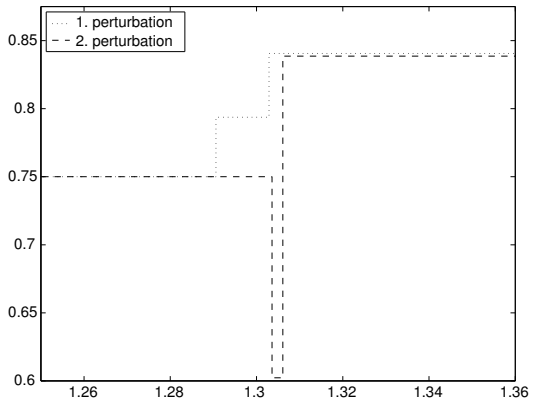


FIGURE 9. Example 3. Wave patterns in the s-component as a function of x for two slightly perturbed right states ($s_1^R = s_0^R + 0.001$ and $s_2^R = s_0^R - 0.001$) for overcompressive shock case.

If this Riemann problem is perturbed slightly, the solution changes from a single shock to a composition of waves moving with almost the same speed, as shown in Figure 9. There are two possible results of a perturbation; either a monotone solution $u^L \xrightarrow{s} u^1 \xrightarrow{c} u^R$ or a non-monotone solution $u^L \xrightarrow{c} u^2 \xrightarrow{s} u^R$ (as shown Figure 9).

In the viscid case, the Riemann problem will be perturbed instantly. The result is a truly nonlinear phenomenon, where monotone data gives non-monotone solutions. Therefore, the solution of the inviscid case is a poor approximation to the true solution, and both the OS and COS strategy will fail to resolve the non-monotonicity in the first time step. Figure 10 shows the exact solution at time $t = 1.0$ along with the OS and COS approximations for one time step. The exact (reference) solution has a dip due to the presence of the diffusion term, which instantly perturbs the initial Riemann problem. The OS approximation is monotone and does not resolve the nonlinear dynamics between the convective and diffusive forces, thereby giving the usual smearing of the steep gradient. COS, on the other hand, does not produce the same excessive smearing and gives a very small dip introduced by the nonlinear residual. In both cases the solution misses on the position of the shock because of mass conservation.

With two time steps, as in Figure 11, the difference is more distinct. Neither OS nor COS are able to resolve the dip in the solution properly. This is due to splitting errors from the coarse time discretization. However, when comparing OS and COS, we see that COS is much closer to the reference solution in the steep part (s-shock).

In Table 2 we report the estimated errors for a number of runs with decreasing splitting steps, along with convergence rates. COS and OS have more or less the same rates of convergence. We note, however, that COS in general has lower error than OS. This is caused by the sharpening effect of the residual flux.

When the number of splitting steps becomes large, the solutions deteriorate due to the many projections that are performed; see the discussion in Example 2.

Remark. In the above examples we have not elaborated on how to choose the threshold parameter γ used to identify residual fluxes. When choosing this parameter there are two points to consider: First, γ should be sufficiently small, so that relevant residual fluxes are identified. Second, γ should be chosen so large that no residual fluxes are identified for small-scale oscillations appearing in the hyperbolic solutions (e.g., post-shock oscillations, or oscillations for data near the transition curve [31]). Using these rules of thumb, a suitable parameter can easily be identified in each case, without

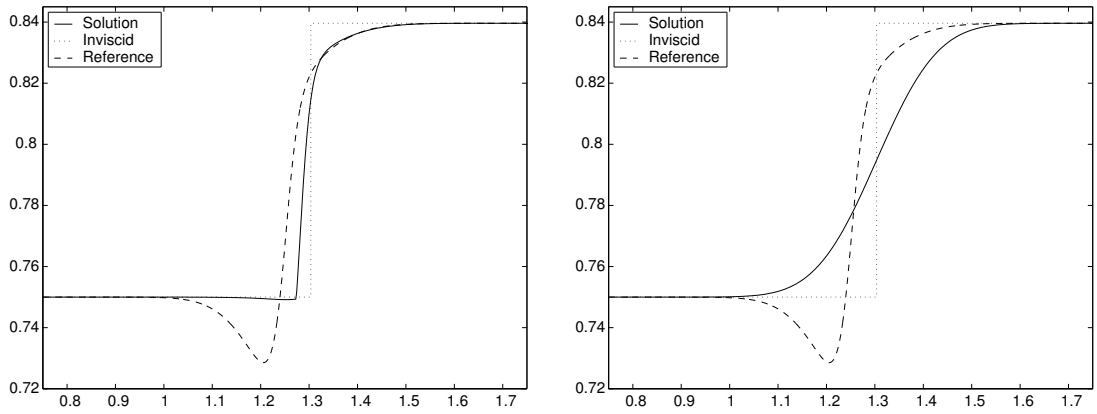


FIGURE 10. Example 3. Comparison of the s-component as a function of x for splitting solutions at time $t = 1.0$ with $\epsilon = 0.005$. The approximate solution is computed with one step of COS (left) and one step of OS (right).

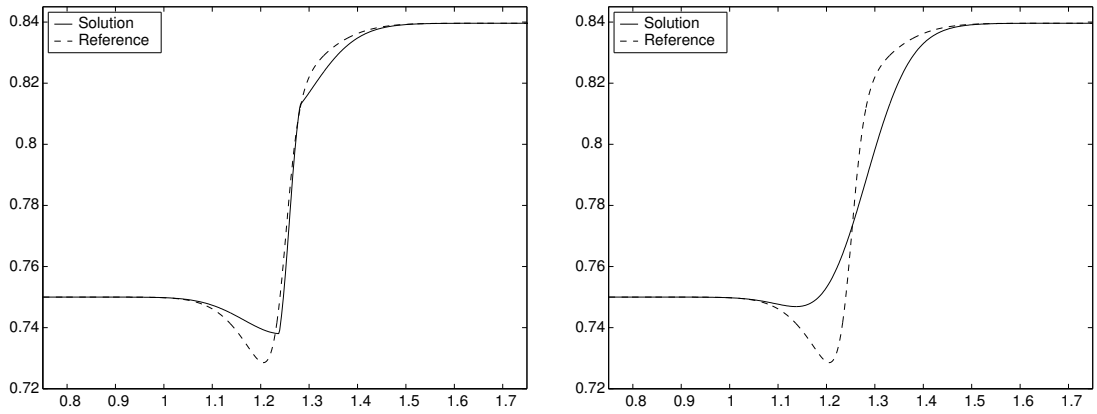


FIGURE 11. Example 3. Comparison of the s-component as a function of x for splitting solutions computed with two steps of COS (left) and OS (right).

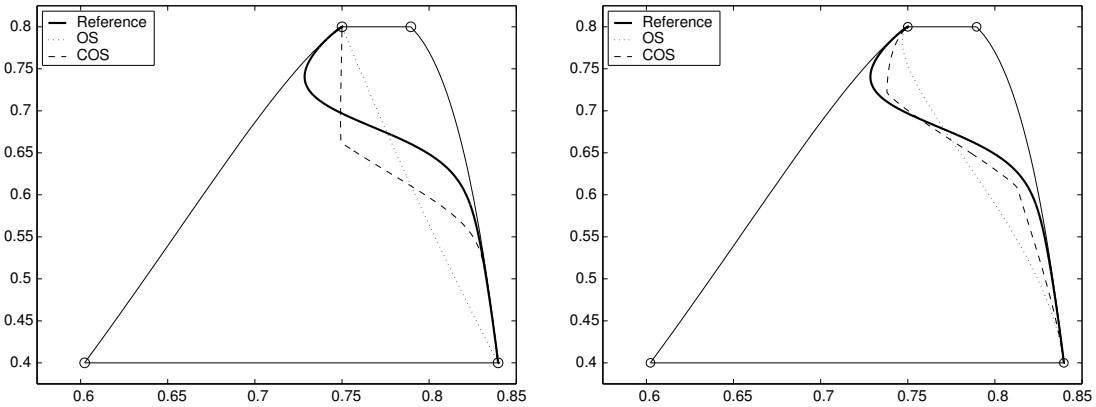


FIGURE 12. Example 3. Splitting solutions in state space after one step (left) and two steps (right). The thin solid lines represents the waves of the slight perturbations (s_1^R and s_2^R).

TABLE 2. *Example 3. Estimated errors* (17) *and convergence rates for OS and COS with fixed spatial discretization* $\Delta x = 2^{-9}$*;* N_t *denotes the number of splitting steps.*

	ε=0.005				ε=0.01			
N_t	OS	rate	COS	rate	OS	rate	COS	rate
1	6.93e-03	—	4.31e-03	—	8.11e-03	—	5.05e-03	—
2	3.94e-03	0.57	2.95e-03	0.38	4.40e-03	0.61	3.60e-03	0.34
4	2.15e-03	0.61	2.67e-03	0.10	2.26e-03	0.67	2.15e-03	0.51
8	1.13e-03	0.64	9.94e-04	0.99	1.16e-03	0.67	1.02e-03	0.74
16	6.23e-04	0.60	5.84e-04	0.53	6.22e-04	0.62	5.86e-04	0.56
32	3.89e-04	0.47	3.77e-04	0.44	3.82e-04	0.49	3.71e-04	0.46
64	3.04e-04	0.25	2.97e-04	0.24	2.80e-04	0.31	2.76e-04	0.30
128	2.85e-04	0.07	2.82e-04	0.05	2.48e-04	0.12	2.46e-04	0.11
256	2.89e-04	-0.02	2.88e-04	-0.02	2.47e-04	0.00	2.46e-04	0.00

any special tuning necessary. In all the above runs, we used a parameter value of $\gamma = 0.1$ (as the size of the oscillations typically were below 0.05).

3.2. **A Triangular Three-Phase Flow Model.** We consider the initial value problem for the 2×2 system of parabolic equations

$$\begin{aligned} \partial_t u + \partial_x f(u) &= \varepsilon \partial_x^2 d(u), \\ \partial_t v + \partial_x g(u,v) &= \varepsilon \partial_x^2 d(v), \end{aligned} \tag{18}$$

where u and v are phase saturations (gas and water, respectively), f and g the fractional flow functions, d the diffusion function, and ε is a small scaling factor. The Jacobian matrix of the fractional flow functions is (lower) triangular, hence a triangular system. The system (18) models flow of oil, gas, and water in a porous medium. Generally, no systems arising from three-phase flow models are triangular, but by assuming that one of the phases, typically the gas phase, is decoupled from the other two we get the system above. In many cases this is a reasonable assumption. Here we use flux functions [25]

$$f(u) = \frac{u^2}{u^2 + (1-u)^2/10},$$

$$g(u,v) = \frac{(1-u)^2 + u^2/10}{10u^2 + (1-u)^2} \cdot \frac{v^2}{v^2 + (1-v)^2/10},$$

and the diffusion function is given as

$$d(x) = 4x(1-x) + 0.01,$$

which thus has the usual bell shape typically seen in reservoir models. For simplicity, a small constant has been added to avoid degeneration of the system. Alternatively, we could have treated the degeneracy by using upwind discretizations of the convective terms.

We present two numerical examples, one simple Riemann problem and a Cauchy problem with non-monotone solution. In both cases, we compare COS and OS solutions against a reference solution computed with a central difference scheme on a very fine discretization.

The Riemann solver. The essential part of the hyperbolic solution operator in our operator splitting method is the Riemann solver whose construction depends heavily on the triangular nature of the system. The procedure for constructing the solution of the Riemann problem is found in Gimse [14]. Let us very briefly recall the basic steps in this procedure.

We solve the Riemann problem for (18) with linearised flux functions (the linearisation parameter is denoted by $\delta > 0$). The scalar decoupled equation for u (the first equation in (18)) is easily solved, and the solution u consists of a finite number (say M) of constant states. Rarefaction waves are approximated by step functions. Then we solve the (coupled) equation for v (the second equation in (18)). In each region in the (x,t) plane where u is constant, the coupled equation reduces to a scalar equation. In order to solve the coupled equation, we must therefore solve M scalar equations and connect together the solutions from the neighbouring regions. To find legal connection values, i.e., legal v-values on each side of the interface between two regions, we must construct so-called H-sets. We mention that the construction of the H-sets is the most time consuming part of the Riemann solver. For further details on the Riemann solver, we refer to [14].

There are three wave types in the solution of the triangular Riemann problem. From the scalar Riemann problem, we have rarefaction waves and shock waves. In addition, there are connection waves, i.e., waves connecting the v-regions in the (x,t) plane. The connection waves will always have the same speed as the corresponding discontinuities in u.

In the numerical examples below, the flux functions are linearized with $\delta = 0.01$. This sets a limit on the resolution of the front tracker and hence the expected accuracy of the solution.

Example 1. As for the polymer system, we consider first a simple Riemann problem:

$$(u_0, v_0)(x) = \begin{cases} (0.4, 0.6), & x \le 1.0, \\ (0.0, 0.0), & x > 1.0. \end{cases} \tag{19}$$

The diffusion coefficient ε is set equal to 0.1, and we seek solutions at time $t = 0.5$. The finite difference operator uses a grid with $\Delta x = 0.01$.

Figure 13 shows solutions computed using one step of OS and COS compared with a reference solution. The COS method sharpens the shock front in u nicely. In addition, the COS method incorrectly sharpens the quite smooth front in v. As a result of this sharpening, the error in v close to the shock is larger for COS than for OS. The left plot in Figure 14 shows local L^1 error in each grid cell as a function of the position of the cell center. The remedy for this erroneous sharpening is to decrease the time step. Figure 14 (right plot) shows cumulative L^1 errors for COS and OS with one and two steps. We observe that already when two time steps are used instead of one, the erroneous sharpening in v (at $x \approx 1.5$) is almost gone, and COS outperforms OS. This is because the front in v is smoother in the second time step and therefore no residual flux is defined.

Example 2. In the second example, we solve the Cauchy problem for (18) with initial data

$$u_0(x) = v_0(x) = \begin{cases} 0.0, & x < 0.4, \\ x - 0.4, & 0.4 < x < 0.8, \\ 0.4, & x > 0.8. \end{cases} \tag{20}$$

Final computing time is $t = 1.0$ and we set $\varepsilon = 0.1$. The finite difference operator uses a grid with $\Delta x = 0.01$.

The performance of COS in this example is very good. In Figure 15, OS and COS solutions computed with one time step are compared with the reference solution. The OS solution is nowhere near catching the shock fronts in u and v. On the other hand, the correction achieved by adding the residual fluxes enables the COS method to resolve the shock fronts almost perfectly.

When going from one to two time steps in COS, we observe a nice decrease in the error, see Figure 16 (left plot). However, when increasing the number of splitting steps to four, the error increases almost to the level observed for one step, see Figure 16 (right plot). The reason is the way the residual

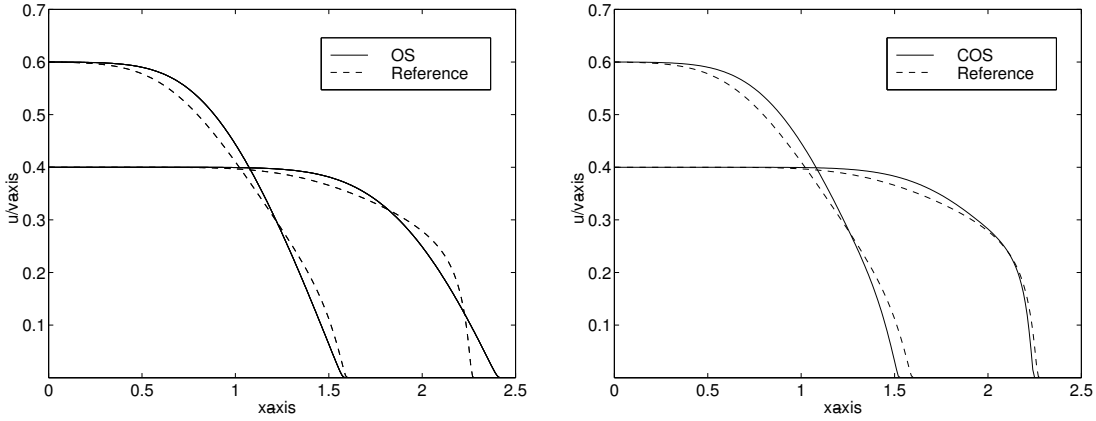


FIGURE 13. Example 1. Solution of Riemann problem (19) at time $t = 0.5$ for $\varepsilon = 0.1$ computed using one step of OS (left) and one step of COS (right).

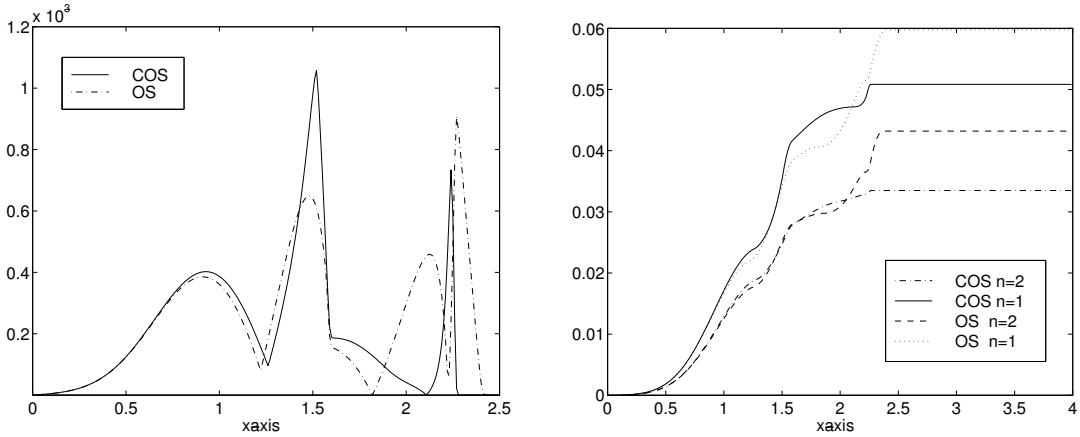


FIGURE 14. Example 1. Error measurements for Riemann problem (19) at time $t = 0.5$ for $\varepsilon = 0.1$. (Left) L^1 error in each grid cell for COS and OS with one time step. (Right) Cumulative L^1 error for COS and OS with one and two steps.

fluxes are defined. If the distance between two shocks is less than the (prescribed) length of the correction intervals, the residual fluxes are only defined to the midpoint between the two shocks. When the time step decreases, the distance between the two v-shocks also decreases. As a result, the correction interval for each residual flux also decreases, and the sharpening effect is reduced. Thus the improvement of the OS solution is smaller, but COS still gives a lower error than OS with the same number of time steps.

To study the convergence of OS and COS, we fix the spatial discretization Δx to 0.01 on the interval $(0, 4)$ and increase the number of splitting steps by powers of two. Table 3 gives errors and convergence rates for this convergence study for $\varepsilon = 0.1$. The error is defined as

$$E = \sum_{i=1}^{\varrho} \|u_i - u_i^r\|_1, \tag{21}$$

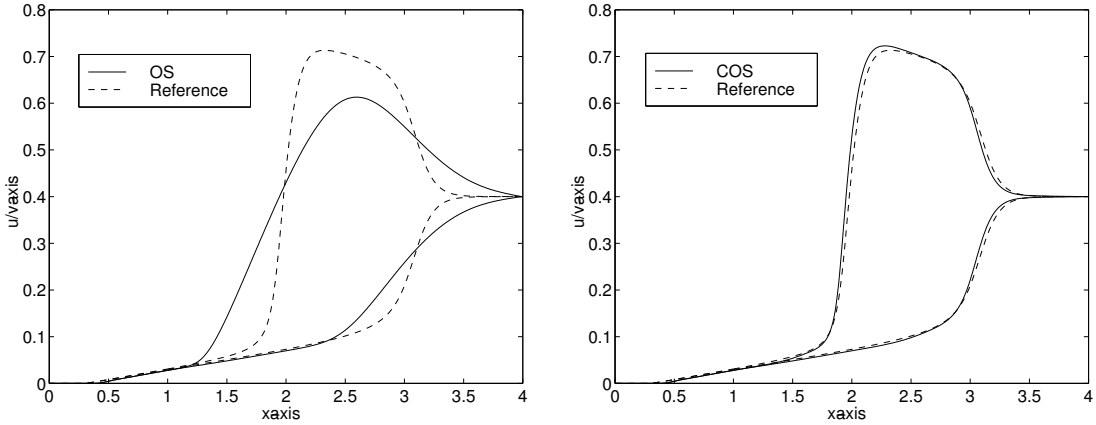


FIGURE 15. Example 3. Solutions of Cauchy problem (20) at time $t = 1.0$ for $\varepsilon = 0.1$ computed using one step of OS (left) and one step of COS (right).

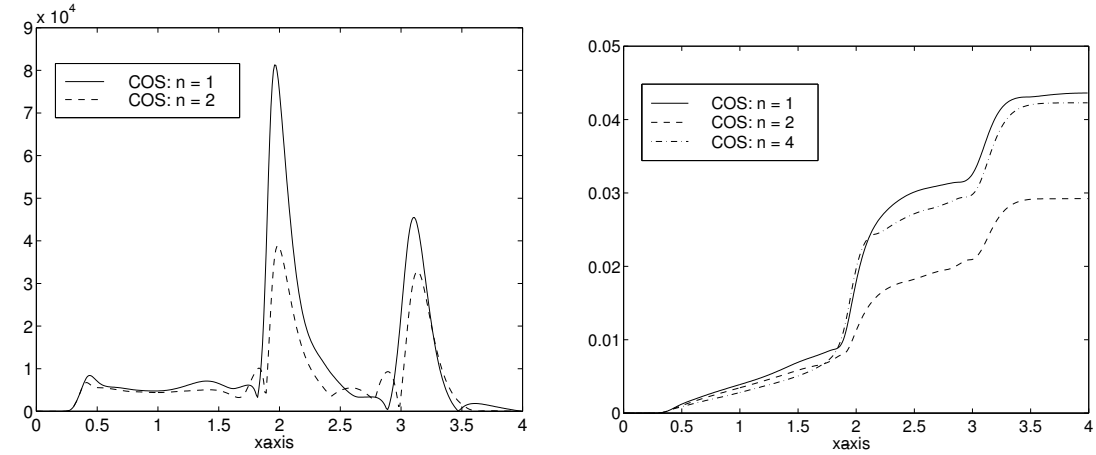


FIGURE 16. Example 3. Error measurements for COS for Cauchy problem (20) at time $t = 1.0$ with $\varepsilon = 0.1$; (left) L^1 error in each grid cell, (right) cumulative L^1 error.

where $(u_1, \ldots, u_\varrho)$ denotes the splitting solution and $(u_1^r, \ldots, u_\varrho^r)$ the reference solution. We see that the error for OS decreases up to $N_t = 64$, but then starts to increase as a result of numerical diffusion. The error for COS first decreases and then increases for $N_t = 4, 8$ as a result of decreasing distance between the two v-shocks (as explained above). For N_t larger than 8, COS performs as OS.

Remark. The COS method gives very good results. Using an implicit diffusion solver would make the runtimes go down and thus make the COS method very efficient. We have not done this here, since we were merely interested in testing whether COS reduces the temporal splitting error compared to OS. In the COS method we have chosen to prescribe the length of the correction intervals as $K\varepsilon$ with $K = 10$. Moreover, as for the polymer system in Section 3.1, we chose $\gamma = 0.1$ as the value for the threshold parameter used to pick out the shocks that define the residual fluxes.

TABLE 3. *Example 3. Estimated errors* (21) *and convergence rates for OS and COS with fixed spatial discretization* $\Delta x = 0.01$*;* N_t *denotes the number of splitting steps.*

	ε=0.1			
N_t	OS	rate	COS	rate
1	2.76e-01	—	4.36e-02	—
2	1.75e-01	0.65	2.92e-02	0.58
4	9.76e-02	0.84	4.23e-02	-0.53
8	5.51e-02	0.82	5.59e-02	-0.40
16	3.19e-02	0.79	2.97e-02	0.91
32	2.15e-02	0.56	2.11e-02	0.49
64	1.80e-02	0.26	1.80e-02	0.24
128	1.99e-02	-0.14	1.99e-02	-0.15
256	2.30e-02	-0.20	2.31e-02	-0.21

4. A TWO-DIMENSIONAL EXTENSION

In this section we present a two-dimensional extension of the COS idea for the polymer system using dimensional splitting. The same idea can be applied to other systems and higher dimensions. To add more realism to the equation, we also include a driving velocity field. That is, we consider

$$\begin{aligned} \partial_t s + V(x)\cdot\nabla f(s,c) &= \varepsilon\Delta s \\ \partial_t b + V(x)\cdot\nabla(c f(s,c)) &= \varepsilon\Delta b, \end{aligned} \tag{22}$$

where f is given by (15) with $\mu = 0.25$ and $\nu = 9$, $x = (x_1, x_2)$, $V = (V_1, V_2)$, $\nabla = (\partial_{x_1}, \partial_{x_1})$, and $\Delta = \partial_{x_1}^2 + \partial_{x_2}^2$. The velocity field is given by a pressure equation combined with Darcy's law:

$$-\nabla(\lambda(s,c)\nabla p) = 0, \quad V = -\lambda(s,c)\nabla(p), \tag{23}$$

where λ denotes a total mobility.

A common strategy to solve the overall system (22) and (23) is to decouple the equations by operator splitting: First solve the pressure equation (23) with coefficients given by the initial fluid distribution. Then the velocity field is computed using Darcy's law, and this velocity is held fixed while the saturation s and polymer concentration c are advanced forward according to (22). Then the process is repeated. In the following we will assume that the velocity field is a given quantity and concentrate on the solution of (22).

Let $\Delta x > 0$ and $\Delta t > 0$ denote the spatial and temporal discretization parameters associated with our discrete splitting method. Let $\mathcal{S}_t^{x_j}$ denote the front tracking solution operator associated with the following one-dimensional system of conservation laws

$$\begin{aligned} \partial_t s + V_j(x)\partial_{x_j} f_j(s,c) &= 0 \\ \partial_t b + V_j(x)\partial_{x_j}(c f_j(s,c)) &= 0. \end{aligned} \tag{24}$$

The front tracking algorithm can easily be modified to incorporate varying velocity fields, see e.g., Lie [26] or Haugse *et al.* [16]. Moreover, residual fluxes are identified as outlined above for $V \equiv 1$. Now, let $\mathcal{P}_t^{x_j}$ denote the finite difference solution operator of the following parabolic system

$$\begin{aligned} \partial_t s + V_j(x)\partial_{x_j} f_1^{\mathrm{res},j}(x,s,c) &= \varepsilon\partial_{x_j}^2 s \\ \partial_t b + V_j(x)\partial_{x_j} f_2^{\mathrm{res},j}(x,s,c) &= \varepsilon\partial_{x_j}^2 b. \end{aligned} \tag{25}$$

Note that x_i, $i \neq j$, act only as parameters in (25). We then introduce the following operator splitting solution

$$(s,b)(x,n\Delta t) \approx (s^n, b^n) \equiv \left[\mathcal{P}_{\Delta t}^{x_2} \circ \Pi \circ \mathcal{S}_{\Delta t}^{x_2} \circ \mathcal{P}_{\Delta t}^{x_1} \circ \Pi \circ \mathcal{S}_{\Delta t}^{x_1}\right]^n \Pi(s_0, b_0). \tag{26}$$

Example 1. In the first example, we consider a standard test case from reservoir simulation, the quarter five-spot case, which consists of a unit square with an injection well in the lower left corner and a production well in upper right. A no-flow condition is specified along the boundaries. We consider a reservoir initially filled with pure oil into which pure water with a polymer concentration 0.1 is injected. The diffusion coefficient ε is set to 0.005. To solve the parabolic systems (25) we use a componentwise upwind method.

Figure 17 shows the saturation component of the solution at time $t = 0.4$ (corresponding to 0.4 pore volumes of fluid injected) computed by COS and OS. A reference solution computed on a 257×257 grid is given in Figure 18. Due to very large velocities near the wells, we use small splitting steps (CFL number 2.0) up to time $t = 0.02$ to stabilize the profile and then four splitting steps to reach final time $t = 0.4$. The splitting steps have been intentionally chosen large to demonstrate the error mechanisms in the shock layers, hence the lack of symmetry due to dimensional splitting errors. Much of the dimensional splitting error is removed by increasing the number of splitting steps to e.g., 16. Furthermore, we see that OS and COS with 16 steps coincide except near the diagonal, where the velocity field is sufficiently strong to form discontinuities in the hyperbolic steps and hence residual fluxes.

Example 2. In the second example, we add a heterogeneous permeability field to the quarter five-spot. The permeability field is realised as a log-Gaussian random field. Due to the heterogeneity, the water front will contain viscous fingers. To resolve the fingering properly, one has to reduce the splitting steps compared with the example above. Figure 19 shows saturation and polymer concentration profiles at time $t = 0.3$ for three different values of ε. The length of the splitting steps varies according to a CFL target 4.0 for the hyperbolic steps. Although residual fluxes are seldom identified in these simulations (for $\varepsilon = 0.05$ and 0.005), the mechanism is embedded and is automatically invoked whenever necessary to prevent viscous splitting errors.

5. Concluding Remarks

We have demonstrated numerically that operator splitting (OS) methods for systems of convection-diffusion equations in one space dimension have a tendency to be too diffusive near viscous shock waves. In the scalar case [23, 20], this is related to the fact that the entropy condition (Oleinik's convexification criterion) forces the hyperbolic solver in the convection step to throw away information about the structure of the viscous shock waves, at least when the splitting step is large, thereby creating an entropy loss. To reduce this temporal splitting error, Karlsen and Risebro [23] (see also [20, 12]) introduced the corrected operator splitting (COS) method for scalar convection-diffusion equations. The idea behind the scalar COS method is to use the wave structure from the convection step to identify where the nonlinear splitting error (or entropy loss) occurs. The potential error is then compensated for in the diffusion step (or in a separate correction step).

In the present paper, we have outlined the mechanisms behind the splitting error for systems of convection-diffusion equations. Similarly to the scalar case, the splitting error is intimately related to the local linearisations introduced implicitly in the convection steps due to the use of an entropy condition. Moreover, we have proposed a working COS method for systems. A front tracking method [27, 28, 29] for systems of conservation laws, which in turn relies on a Riemann solver, is an important part of this COS method. The proposed COS method has been applied to a 2×2 system modelling two-phase, multicomponent flow and a triangular 2×2 system modelling three-phase flow. The numerical examples demonstrate that the COS method is significantly more accurate than the corresponding OS method when the splitting step is large and the solution consists of (moving) viscous shock waves. We have extended the COS method to two-dimensional systems of convection-diffusion

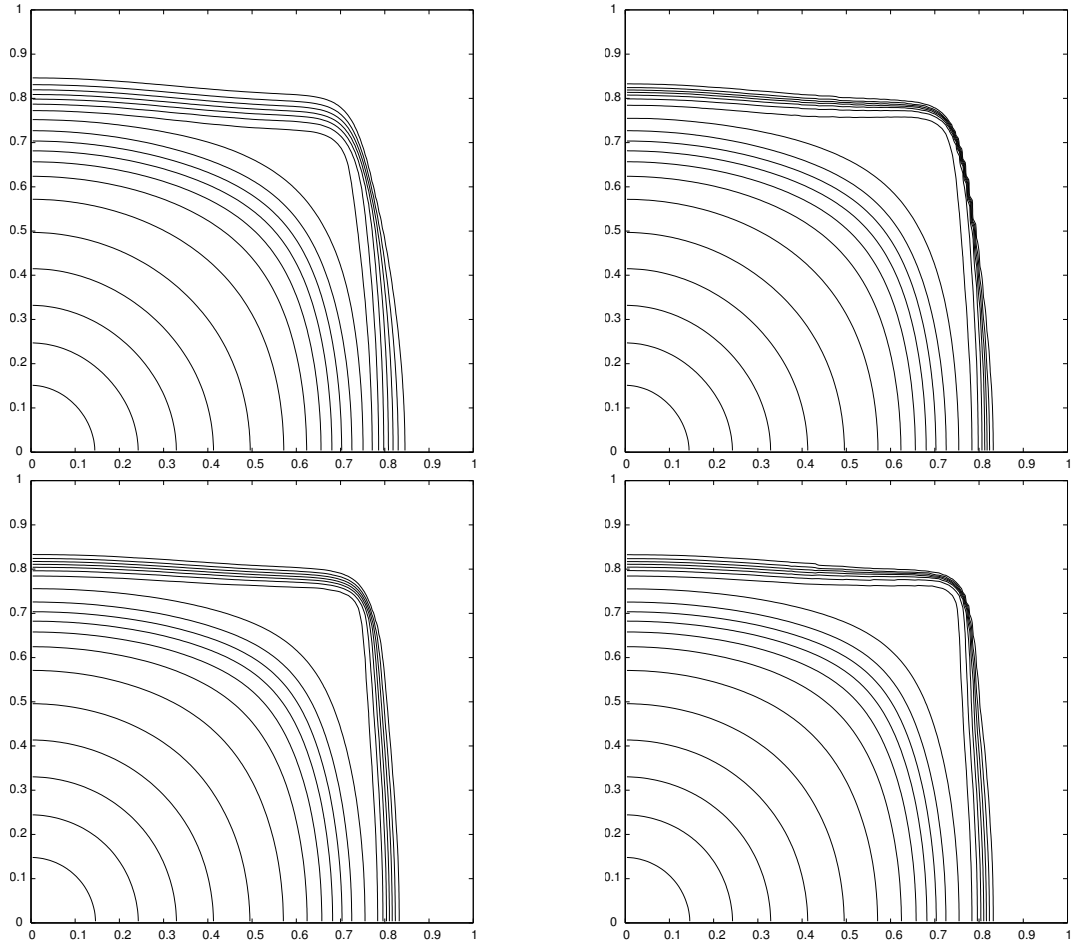


FIGURE 17. Example 1. Saturation component of the solution at time $t = 0.4$ computed by OS (left) and COS (right) with 4 steps (upper row) and with 16 steps (lower row).

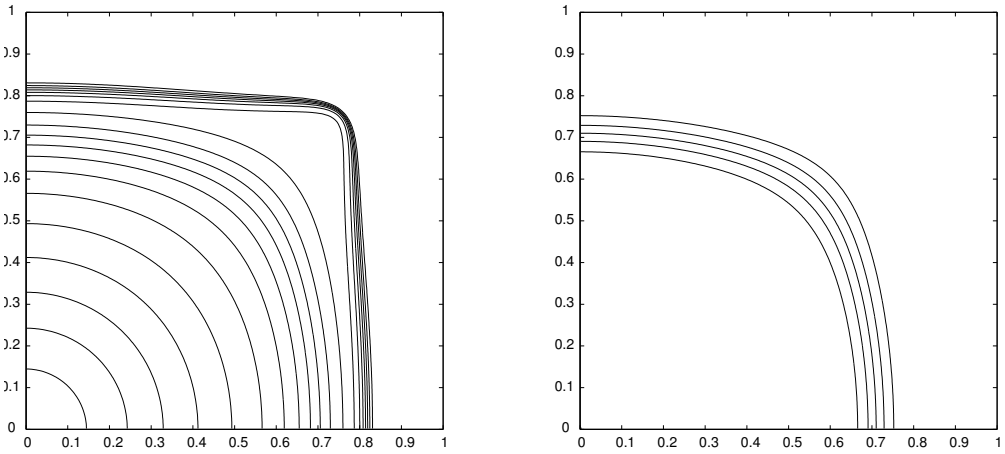


FIGURE 18. Example 1. Saturation (left) and polymer concentration (right) computed on a 257×257 grid with CFL number 2.0.

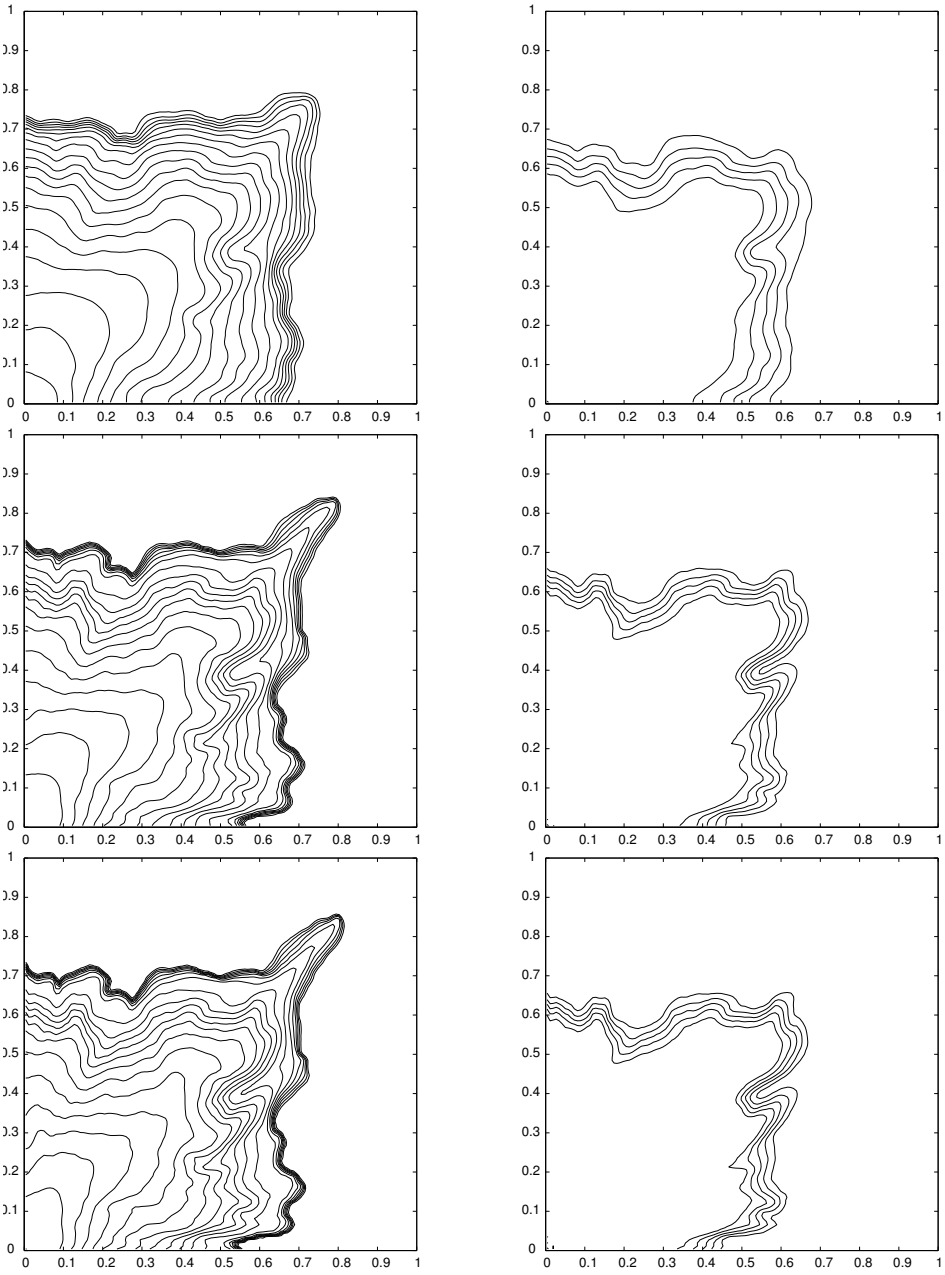


FIGURE 19. Example 2. Saturation and concentration profiles at time $t = 0.3$ computed on a 129×129 grid. The diffusion coefficient is $\varepsilon = 0.05$, 0.005, and 0.0005 from top to bottom.

equations by means of dimensional splitting and applied it to a polymer system with a driving velocity field. In closing, we mention that the COS approach can be implemented for other systems. Moreover, we believe that one can replace the front tracking method in the COS approach by, e.g., the Godunov method.

REFERENCES

[1] A. Bressan, T.-P. Liu, and T. Yang. L^1 stability estimates for $n \times n$ conservation laws. *Arch. Ration. Mech. Anal.*, 149(1):1–22, 1999.

[2] H. K. Dahle. *Adaptive characteristic operator splitting techniques for convection-dominated diffusion problems in one and two space dimensions*. PhD thesis, Department of Mathematics, University of Bergen, Norway, 1988.

[3] H. K. Dahle, M. S. Espedal, and R. E. Ewing. Characteristic Petrov-Galerkin subdomain methods for convection-diffusion problems. In *Numerical simulation in oil recovery (Minneapolis, Minn., 1986)*, pages 77–87. Springer, New York, 1988.

[4] H. K. Dahle, M. S. Espedal, R. E. Ewing, and O. Sævareid. Characteristic adaptive subdomain methods for reservoir flow problems. *Numer. Methods Partial Differential Equations*, 6(4):279–309, 1990.

[5] H. K. Dahle, M. S. Espedal, and O. Sævareid. Characteristic, local grid refinement techniques for reservoir flow problems. *International Journal for Numerical Methods in Enginering*, 34:1051–1069, 1992.

[6] H. K. Dahle, R. E. Ewing, and T. F. Russell. Eulerian-Lagrangian localized adjoint methods for a nonlinear advection-diffusion equation. *Comput. Methods Appl. Mech. Engrg.*, 122(3-4):223–250, 1995.

[7] C. Dawson. Godunov-mixed methods for advection-diffusion equations in multidimensions. *SIAM J. Numer. Anal.*, 30(5):1315–1332, 1993.

[8] C. Dawson. High resolution upwind-mixed finite element methods for advection-diffusion equations with variable time-stepping. *Numer. Methods Partial Differential Equations*, 11(5):525–538, 1995.

[9] C. N. Dawson. Godunov-mixed methods for advective flow problems in one space dimension. *SIAM J. Numer. Anal.*, 28(5):1282–1309, 1991.

[10] C. N. Dawson and M. F. Wheeler. Time-splitting methods for advection-diffusion-reaction equations arising in contaminant transport. In *ICIAM 91 (Washington, DC, 1991)*, pages 71–82. SIAM, Philadelphia, PA, 1992.

[11] M. S. Espedal and R. E. Ewing. Characteristic Petrov-Galerkin subdomain methods for two-phase immiscible flow. *Comput. Methods Appl. Mech. Engrg.*, 64:113–135, 1987.

[12] M. S. Espedal and K. H. Karlsen. *Numerical solution of reservoir flow models based on large time step operator splitting algorithms*. Filtration in Porous Media and Industrial Applications (Cetraro, Italy 1998), Lecture Notes in Math., 1734, Springer, 2000.

[13] S. Evje, K. H. Karlsen, K.–A. Lie, and N. H. Risebro. Front tracking and operator splitting for nonlinear degenerate convection–diffusion equations. In *Parallel solution of partial differential equations*, Eds. P. Bjørstad and M. Luskin, *IMA Vol. Math. Appl.*, Vol. 120, pp. 209–228, Springer Verlag, 2000.

[14] T. Gimse. A numerical method for a system of equations modelling one-dimensional three-phase flow in a porous medium. In *Nonlinear hyperbolic equations—theory, computation methods, and applications (Aachen, 1988)*, pages 159–168. Vieweg, Braunschweig, 1989.

[15] J. Glimm. Solutions in the large for nonlinear hyperbolic systems of equations. *Comm. Pure Appl. Math.*, 18:697–715, 1965.

[16] V. Haugse, K. H. Karlsen, K.-A. Lie, and J. R. Natvig. Numerical solution of the polymer system by front tracking. *Transport in Porous Media*. To appear.

[17] D. Hoff and J. Smoller. Error bounds for finite-difference approximations for a class of nonlinear parabolic systems. *Math. Comp.*, 45(171):35–49, 1985.

[18] H. Holden, K. H. Karlsen, K.–A. Lie, and N. H. Risebro. Numerical solution of nonlinear partial differential equations using operator splitting methods: An L^1 convergence theory. Preprint (in preparation).

[19] T. Johansen and R. Winther. The solution of the Riemann problem for a hyperbolic system of conservation laws modelling polymer flooding. *SIAM J. Math. Anal.*, 19:541–566, 1988.

[20] K. H. Karlsen, K. Brusdal, H. K. Dahle, S. Evje, and K.–A. Lie. The corrected operator splitting approach applied to a nonlinear advection–diffusion problem. *Comput. Methods Appl. Mech. Engrg.*, 167(3-4):239–260, 1998.

[21] K. H. Karlsen and K.–A. Lie. An unconditionally stable splitting scheme for a class of nonlinear parabolic equations. *IMA J. Numer. Anal.*, 19(4):609–636, 1999.

[22] K. H. Karlsen, K.–A. Lie, N. H. Risebro, and J. Frøyen. A front-tracking approach to a two-phase fluid-flow model with capillary forces. *In Situ (Special issue on reservoir simulation)*, 22(1):59–89, 1998.

[23] K. H. Karlsen and N. H. Risebro. Corrected operator splitting for nonlinear parabolic equations. *SIAM J. Numer. Anal.* Vol. 37 (2000), No. 3, pp. 980–1003.

[24] K. H. Karlsen and N. H. Risebro. An operator splitting method for convection-diffusion equations. *Numer. Math.*, 77(3):365–382, 1997.

[25] J. A. Kok. Front tracking for three phase flow in porous media. Final report of the postgraduate program 'Mathematics for industry', Eindhoven, Netherlands, 1994.
[26] K.-A. Lie. A dimensional splitting method for quasilinear hyperbolic equations with variable coefficients. *BIT*, 39(4):683–700, 1999.
[27] N. H. Risebro. A front-tracking alternative to the random choice method. *Proc. Amer. Math. Soc.*, 117(4):1125–1139, 1993.
[28] N. H. Risebro and A. Tveito. Front tracking applied to a nonstrictly hyperbolic system of conservation laws. *SIAM J. Sci. Stat. Comput.*, 12(6):1401–1419, 1991.
[29] N. H. Risebro and A. Tveito. A front tracking method for conservation laws in one dimension. J. Comp. Phys., 101(1):130–139, 1992.
[30] A. Tveito. Convergence and stability of the Lax-Friedrichs scheme for a nonlinear parabolic polymer flooding problem. *Advances In Applied Mathematics*, 11:220–246, 1990.
[31] A. Tveito and R. Winther. The solution of the nonstrictly hyperbolic conservation law may be hard to compute. *SIAM J. Sci. Comp.*, 16:320–329, 1995.
[32] M. F. Wheeler, W. A. Kinton, and C. N. Dawson. Time-splitting for advection-dominated parabolic problems in one space variable. *Comm. Appl. Numer. Methods*, 4(3):413–423, 1988.

Paper B

A local streamline Eulerian-Lagrangian method for two-phase flow *

* Published in Computational Methods in Water Resources (L.R Bentley, J.F Sykes, C.A Brebbia, W.G gray, and Pinder G.F, eds.), vol. 2, A.A Balkema, 2000, pp. 645–650.

A Local Streamline Eulerian-Lagrangian Method for Two-Phase Flow

A.K. Ask, H.K. Dahle, K.H. Karlsen and H.F Nordhaug
Dept. of Mathematics, University of Bergen, NORWAY

ABSTRACT: A method for solving the saturation equation for two-phase flow is presented. The method may be viewed as an operator splitting method or as an Eulerian-Lagrangian backtracking procedure or as a modified method of characteristics. For each time step, the method consists of an advection step and a diffusion step. The advection step requires the tracking of streamlines locally around certain integration points. On each streamline we have to solve a nonlinear hyperbolic equation. This is done using a front tracking method. The solution from the tracking step is projected back onto a fixed grid, and we finally have to solve a parabolic heat-type equation.

1 INTRODUCTION

The numerical solution of advective-diffusive transport problems arise in many important applications in science and engineering, e.g. oil reservoir flow, transport of solutes in ground water and surface water, the movement of aerosols and trace gases in the atmosphere, to mention a few examples. The difficulty of solving such problems, especially if advection dominates, have long been recognized as one of the more challenging tasks in scientific computing, see (Morton 1996) for an overview.

An important class of numerical schemes for solving such problems are the Eulerian-Lagrangian localized adjoint methods (ELLAM) (Celia et al. 1990; Herrera et al. 1993; Wang et al. 1999). These schemes have been successfully applied to linear transport problems of various types, and give a framework for devising Eulerian-Lagrangian type methods that are both mass conservative and able to handle boundary conditions in a fairly systematic way.

A more difficult problem is the solution of multiphase transport processes. This leads to nonlinear advection which greatly complicates the tracking part of the algorithm. Usually this difficulty has been overcome by some kind of linearization, e.g., (Dahle et al. 1995; Douglas et al. 1997). However, such linearizations may put artificial restrictions on the time steps that are not dictated by the physical processes investigated.

The tracking algorithm is a major part of any Eulerian-Lagrangian type method. Essentially, this part consists of tracing flow lines from certain integration points through a fixed spatial grid. Usually the velocity field is calculated as a part of the model, and the calculation of streamlines may then be done efficiently by analytical methods (Pollock 1988; Goode 1990; Datta-Gupta & King 1995; Lu 1994; Russell & Healy 2000), or less efficiently by ode-solvers like the Runge-Kutta methods. On the other hand, standard ode-solvers do not depend on any particular representation of the velocity field, and are a lot easier to implement.

A number of Eulerian-Lagrangian type methods have been suggested to solve the saturation equation for two-phase immiscible flow, e.g., (Dahle et al. 1990; Dahle et al. 1992; Dawson 1991; Douglas et al. 1997; Espedal & Karlsen 1999). In petroleum research, streamline/streamtube-methods have become very popular for solving this type of problems (Datta-Gupta & King 1995; Bratvedt et al. 1996; Hewett & Yamada 1997; King & Datta-Gupta 1998). The main difficulties encountered are the nonlinearity of the fractional flow (or flux) function f (typically s-shaped), which leads to self-sharpening fronts, and the possible degenerate nature of the small capillary-diffusion term.

Here we focus on the first difficulty and will present a scheme based on the following steps: *Lagrangian-step*: Calculate the streamlines locally around certain integration points on a fixed grid. Since the transport equation is nonlinear, the calculation of the foot of the streamline, i.e. the point where the streamline cross the previous time level, becomes a nonlinear hyper-

bolic equation. Thus, to advect the solution along the local streamlines we use front-tracking (Holden et al. 1988) as a fast hyperbolic solver. This step involves a projection of the solution from the previous time step, onto the streamline. *Eulerian-step*: The advected solution is projected back onto a fixed grid, and then used as initial condition for a heat-type equation. This parabolic equation is solved on a fixed grid, using a finite difference scheme.

In the following, we first introduce a simple model for two-phase flow. Then, a fairly detailed description of the numerical method is given, followed by some preliminary numerical results and conclusions.

2 MODEL

The problem to be considered here is the saturation equation for two-phase flow (Chavent & Jaffre 1986):

$$u_t + \nabla \cdot (f(u)\mathbf{v}) = \epsilon \nabla \cdot (D(u)\nabla u), \tag{1}$$

where $u \in [0,1]$ is the saturation of the wetting phase. For ease of presentation, we restrict the problem to a two-dimensional domain. Furthermore, we have neglected gravity in this model, since gravity gives rise to an additional advection term (Karlsen et al. 1998). The velocity field $\mathbf{v}$ is normally calculated as part of the model, involving Darcy's law and a compressibility condition. For the purpose of this work, we shall simply assume that $\mathbf{v}$ is a given vector-valued function which satisfy:

$$\nabla \cdot \mathbf{v} = 0. \tag{2}$$

The fractional flow (or flux) function f is given as the relative permeability of the wetting phase divided by the sum of the relative permeabilities. A simple analytic expression for this function is given by

$$f(u) = \frac{u^2}{u^2 + (1-u)^2}. \tag{3}$$

Note that $f' \geq 0$ for $u \in [0,1]$. The capillary diffusion coefficient $D(u)$ is generally a nonlinear (bell-shaped) function of u which becomes zero at the end points $u = 0, 1$. Thus, (1) is an example of a parabolic degenerate equation. This degeneracy add extra difficulties that will be avoided here by setting $D = 1$. Finally, the parameter ϵ determines the relative importance of advective and diffusive forces, and is small for advection dominated problems.

To close this model we need to specify an initial state given by

$$u(x, y, 0) = u_0(x, y), \tag{4}$$

and boundary conditions. Generally, boundary conditions add greatly to the complexity of any numerical scheme, and is one reason why the ELLAM-methodology was originally devised. However, since the main purpose of this work is to demonstrate a tracking concept for nonlinear transport problems, we shall assume that the saturation u is defined on an infinite domain. The numerical solver will then be supplied with zero Dirichlet- or Neumann-conditions, whenever required.

3 NUMERICAL FORMULATION

Let the computational domain be discretized by a rectangular Cartesian grid, and assume that an approximate solution, U_{ij}^{n-1}, of equation (1), is given at each node $\mathbf{x}_{ij}$ on the grid and time t^{n-1}. The problem is to find a new approximation at time-level t^n, where $t^n - t^{n-1} = \Delta t$. To do so, observe that a directional derivative

$$\frac{d}{d\xi} = \mathbf{v} \cdot \nabla, \tag{5}$$

may be defined along streamlines

$$\frac{d\mathbf{r}}{d\xi} = \mathbf{v}. \tag{6}$$

It follows that the hyperbolic part of the problem will greatly simplify if we can construct a new orthogonal coordinate system based on streamlines and velocity equipotentials, see e.g. (King & Datta-Gupta 1998).

The approach taken here is to split equation (1) into a *hyperbolic part*:

$$u_t + \mathbf{v} \cdot \nabla f(u) \equiv u_t + f_\xi(u) = 0, \tag{7}$$

using (2) and (5), and a *parabolic part*:

$$u_t = \epsilon \nabla \cdot (D \nabla u). \tag{8}$$

The procedure is then first to advect the solution from one time level to the next by solving (7) with the solution at the previous time level as initial condition. Secondly, this solution is diffused by solving (8) to obtain a final solution at the new time level. This operator-splitting algorithm is analyzed by (Karlsen & Risebro 1997).

3.1 *Calculating Streamlines*

Let $\mathbf{x}_q$ denote fixed integration points. The placement and number of integration points are somewhat arbitrary, but is chosen to be the cell-centers in this work. We have to approximate streamlines backwards from $\mathbf{x}_q$ since $f' \geq 0$. An ode-solver denoted $RK(\cdot)$ is used to solve equation (6). Let $\overline{\mathbf{x}}_q = \overline{\mathbf{x}}_q(\xi)$ be the (approximate) streamline such that $\overline{\mathbf{x}}_q(0) = \mathbf{x}_q$. Furthermore, let $\lambda = \max_u |f'|$, and $\Delta\xi_q$ be the (largest) Runge-Kutta step. The streamline has to be traced for $\xi \in [-\lambda\Delta t, 0]$. The following quasi-algorithm describes the tracking:

Algorithm 1:

$\bar{\mathbf{x}}_q = \mathbf{x}_q$;
$\xi = 0$;
while $\xi > -\lambda\Delta t$;
  $\Delta\xi \leftarrow \Delta\xi_q$;
  $\mathbf{x} \leftarrow RK(\Delta\xi, \bar{\mathbf{x}}_q)$;
  while $\mathbf{x} \notin neigh(\bar{\mathbf{x}}_q)$;
    $\Delta\xi \leftarrow \Delta\xi/2$;
    $\mathbf{x} \leftarrow RK(\Delta\xi, \bar{\mathbf{x}})$;
  end while;
  $\xi \leftarrow \xi - \Delta\xi$;
  $\bar{\mathbf{x}}_q \leftarrow \mathbf{x}$;
end while;

The second while-loop is introduced to avoid that a streamline is tracked more than one grid-cell in each Runge-Kutta step. Here, $neigh(\mathbf{x})$ denote the grid-cell to which $\mathbf{x}$ belongs, and the eight neighbour cells to this grid-cell.

Algorithm 1 produces points $\bar{\mathbf{x}}_q^k = \bar{\mathbf{x}}_q(\xi_k) = (x_q^k, y_q^k)$, $k = 1, 2, \ldots, K$, on the streamline. The entire streamline is constructed by introducing straight lines between such points. Thus,

$$\bar{\mathbf{x}}_q(\xi) = \frac{1}{\Delta\xi_k}\left(\bar{\mathbf{x}}_q^{k-1}(\xi_k - \xi) - \bar{\mathbf{x}}_q^k(\xi - \xi_{k-1})\right),$$

for $\xi_k \le \xi \le \xi_{k-1}$. It is now easy to determine points ξ_c^l, $l = 1, 2, \ldots, L$, where the streamline crosses grid-lines. Assume that $x_q^{k-1} \le x_i \le x_q^k$, where x_i is the i-th grid-line orthogonal to the x-axis, then

$$\xi^c = \frac{\Delta\xi_k}{\Delta x_k} x_i + x_q^k \xi_{k-1} - x_q^{k-1}\xi_k.$$

The points where a streamline crosses grid-lines orthogonal to the y-axis, are found by a similar calculation. The ordered set of all such points on a given streamline, $\{\xi_c^l\}$, is of course generated as part of Algorithm 1.

3.2 *Advection*

We wish to determine the solution of (7) at the integration points $\mathbf{x}_q$ and time-level t^n, given the approximate solution U^{n-1}.

Equation (7) is a one-dimensional hyperbolic conservation law. The basic waves of the scalar equation are rarefaction waves and shocks. There exist numerous efficient solver for such problems, see for example (Toro 1999). Here, we will use a front tracking approach. The front tracking method was first presented in (Dafermos 1972), and later developed into a numerical method by (Holden et al. 1988).

The front tracking method requires that the data is replaced by piecewise constants along the streamline. Here, we choose to place the discontinuities at the crossing points $\{\xi_c^l\}$. Then the initial data v_0 can be computed by

$$v_0(\xi) = \frac{1}{\Delta\xi^l}\int_{\xi_c^l}^{\xi_c^{l-1}} U^{n-1}(\xi)d\xi, \qquad (9)$$

for $\xi_c^l < \xi < \xi_c^{l-1}$. To simplify the last calculation, U^{n-1} is replaced by some cell average $\tilde{U}^{n-1}$ on each grid-cell, so that $v_0(\xi) = \tilde{U}^n$ on the segment of the streamline contained in the respective grid-cell. Note that the way the streamlines are defined, states to the left of ξ_q^K and right of $\xi_q^0 = 0$ do not interact with the solution at $\mathbf{x}_q$ for $t \in [t^{n-1}, t^n]$. These states are therefore replaced by the values at the endpoints.

Front tracking proceeds by calculating the position of the discontinuities (fronts) at time t^n. This also imply the solution values at integration points $\mathbf{x}_q$ to be used in the diffusion step.

3.3 *Diffusion*

The solution of Equation (8), given the solution of (7) at the integration points, are open for many choices of discretization. Based on the ELLAM-methodology finite-element or finite-volume type techniques are natural choices. Here, a simple second order accurate explicit central-difference scheme have been used, since the main purpose of this work is to test the tracking concepts. The initial data $\bar{U}_{ij}^{n-1}$ of (8) at nodes $\mathbf{x}_{ij}$ is taken to be

$$\bar{U}_{ij}^{n-1} = \frac{1}{4}\sum_{k=1}^{4} \bar{U}^{n-1}(\mathbf{x}_{q_k}),$$

where $\mathbf{x}_{q_k}$, $k = 1, \ldots, 4$, are the integration points of the four cells that have $\mathbf{x}_{ij}$ as a common vertex.

Note that when solving (8) using an explicit difference scheme, there is a stability constraint on the time step. Consequently, we may have to take many substeps to propagate the solution of (8) from time t^{n-1} to t^n.

4 NUMERICAL EXAMPLES

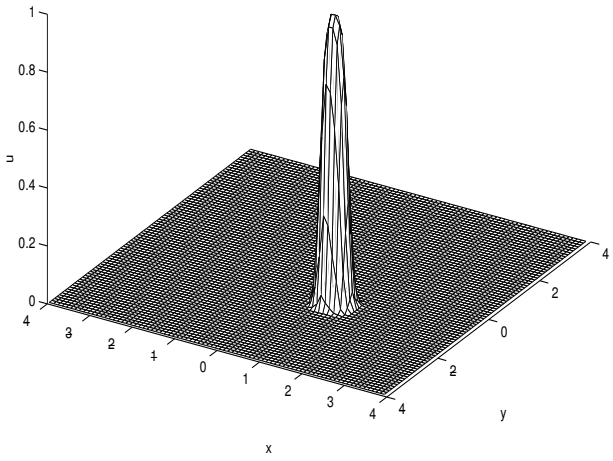
Some preliminary experiments are performed with a rotating velocity field

$$\mathbf{v}(\mathbf{x}) = 2\pi[y, -x].$$

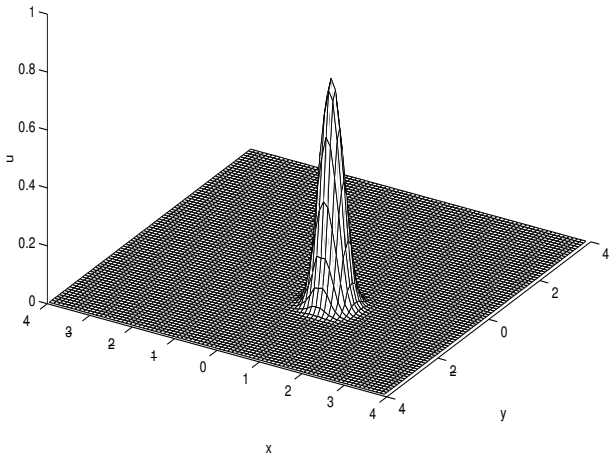
The initial condition (4) is chosen to be the cylindrical profile:

$$u_0(x) = \begin{cases} 0, & (x-1)^2 + (y+1)^2 > 0.3^2, \\ 1, & (x-1)^2 + (y+1)^2 \le 0.3^2. \end{cases}$$

The computational domain is set to be the square $[-4, 4] \times [-4, 4]$. This domain is discretized using a uniform grid of 160 grid-cells in each direction ($\Delta x = \Delta y = 0.05$).

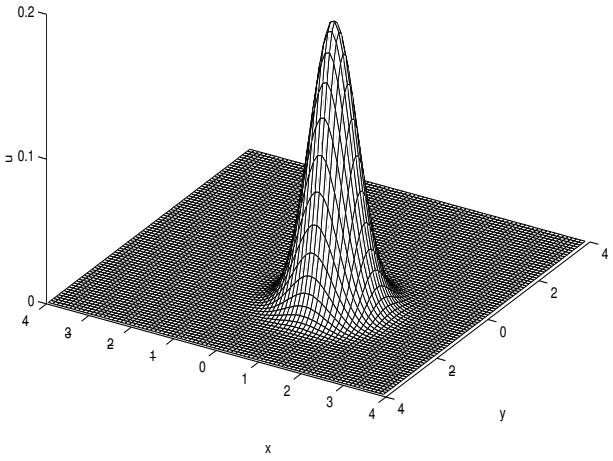


(a) $\Delta t = 0.2$

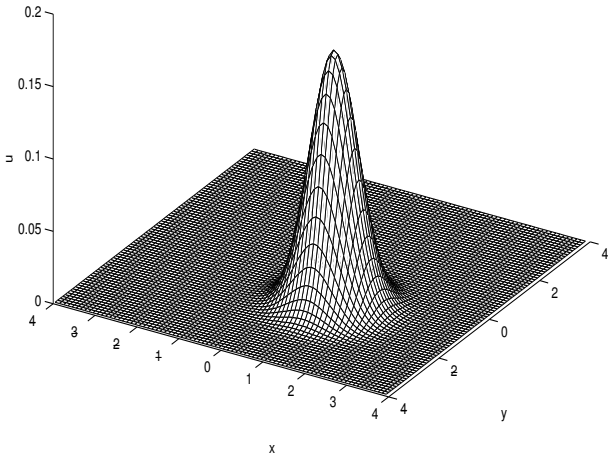


(b) $\Delta t = 0.05$

Figure 1: Solutions with linear flux, $\epsilon = 0$.

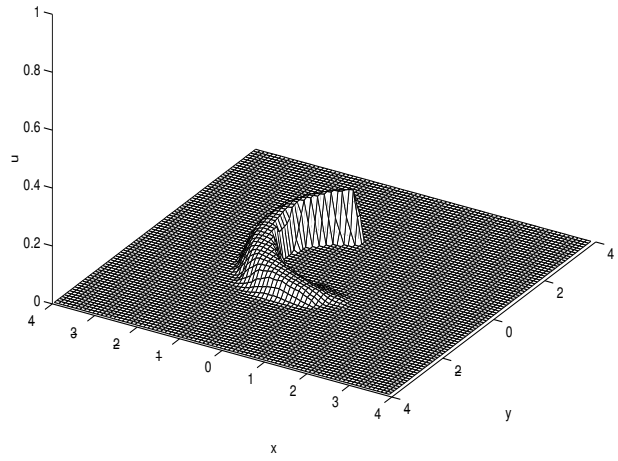


(a) $\Delta t = 0.2$

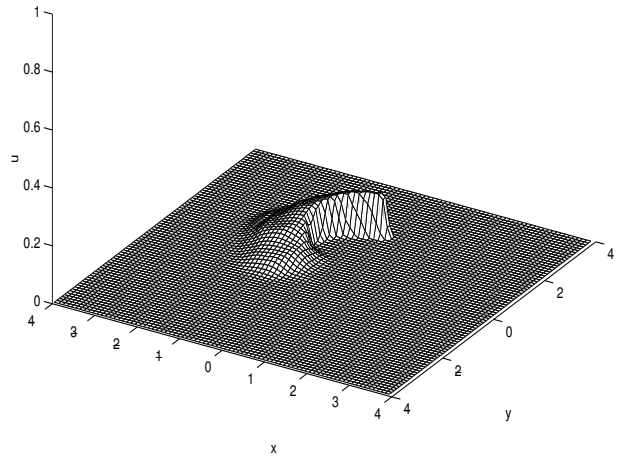


(b) $\Delta t = 0.05$

Figure 2: Solutions with linear flux, $\epsilon = 0.1$.

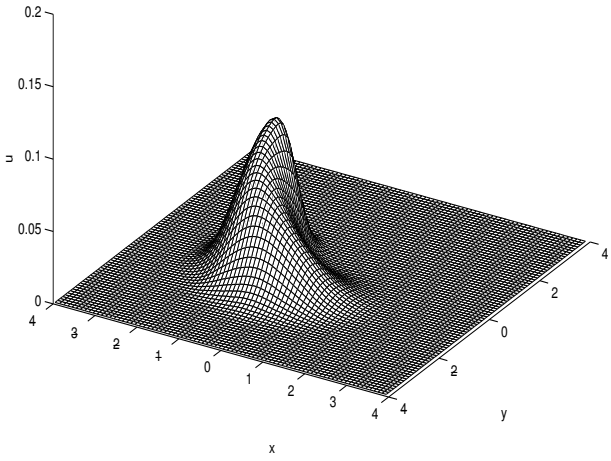


(a) $\Delta t = 0.2$, $\epsilon = 0$

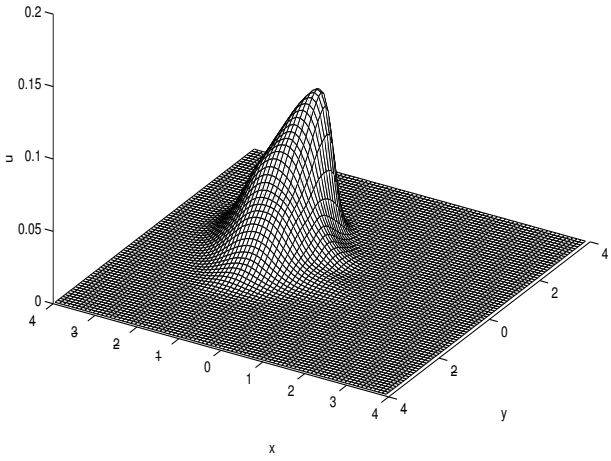


(b) $\Delta t = 0.05$, $\epsilon = 0$

Figure 3: Solutions with nonlinear flux, $\epsilon = 0$.



(a) $\Delta t = 0.2$, $\epsilon = 0.1$



(b) $\Delta t = 0.05$, $\epsilon = 0.1$

Figure 4: Solutions with nonlinear flux, $\epsilon = 0.1$.

We have run two sets of experiments. The first with linear advection, $f(u) \equiv 1$, and the second with a nonlinear flux given by Equation (3). For each experiment the final time is $T = 1$ (one rotation in the linear case), and we have varied the time step as $\Delta t = 0.05, 0.2$. The diffusion parameter is set to be $\epsilon = 0$ (no diffusion) and $\epsilon = 0.1$.

The experiments show that some numerical diffusion is introduced by the projection step, see Figure 1 and Figure 3. In particular, we see that the nonlinear hyperbolic waves are slightly distorted by the number of projection steps, see Figure 3. However, the front-tracking method is itself diffusion free, and reproduces the waves correctly.

The introduction of a diffusion term, seems to alleviate the effect of the projection steps, see Figure 2 and Figure 4 and note that the u-axis is scaled differently form Figures 1 and 3. In fact, the combination of front tracking and local streamlines seems to advect the solution accurately and without grid orientation effects.

5 CONCLUSIONS

The experiments reported here are in agreement with what should be expected: The calculation of local streamlines leads to negligible grid orientation effects and allow us to use fast hyperbolic solvers. Some numerical diffusion is introduced by the projection steps, whereas the front tracker introduce no numerical diffusion.

The method seems to be fairly flexible. However, before any definite conclusions can be drawn, more extensive experiments have to be performed. In particular, semianalytical tracking methods should be implemented, and different projection strategies should be investigated. Comparisons with other methods and/or analytical results must also be done.

Work is in progress on the following modifications/extensions of the method: Most important, a finite-element or finite-volume type method will be implemented for the parabolic step. This should allow us to use concepts from the ELLAM-methodology and enable the method to be tested on more realistic problems. Furthermore, the calculation of streamlines using ode-solvers seems to be computationally very time consuming and will be replaced by semianalytical methods. Finally, splitting errors caused by the nonlinearity of the flux will appear when too large time-steps are taken, see (Karlsen & Risebro ; Karlsen et al. 1998; Brusdal et al. 1998). The use of a front tracker allow us to construct correction terms that may compensate for splitting errors.

ACKNOWLEDGEMENT

We thank Knut-Andreas Lie for letting us use his front tracking code.

REFERENCES

Bratvedt, F., Gimse, T., & Tegnander, C. (1996). Streamline computations for porous media flow including gravity. *Transport in Porous Media 25*(1), 63–78.

Brusdal, K., Dahle, H. K., Karlsen, K. H., & Mannseth, T. (1998). A study of the modelling error in two operator splitting algorithms for porous media flow. *Computational Geosciences 2*, 23–36.

Celia, M. A., Russell, T. F., Herrera, I., & Ewing, R. E. (1990). An Eulerian-Lagrangian localized adjoint method for the advection-diffusion equation. *Adv. Water Resources 13*, 187–205.

Chavent, G. & Jaffre, J. (1986). *Mathematical models and finite elements for reservoir simulation*, Volume 17 of *Studies in mathematics and it's applications*. North Holland, Amsterdam.

Dafermos, C. M. (1972). Polygonal approximation of solutions of the initial value problem for a conservation law. *J. Math. Anal. Appl. 38*, 33–41.

Dahle, H. K., Espedal, M. S., Ewing, R. E., & Sævareid, O. (1990). Characteristic adaptive subdomain methods for reservoir flow problems. *Numerical Methods for Partial Differential Equations 6*, 279–309.

Dahle, H. K., Espedal, M. S., & Sævareid, O. (1992). Characteristic, local grid refinement techniques for reservoir flow problems. *International Journal for Numerical Methods in Engineering 34*, 1051–1069.

Dahle, H. K., Ewing, R. E., & Russell, T. F. (1995). Eulerian-Lagrangian localized adjoint methods for a nonlinear advection-diffusion equation. *Comput. Methods Appl. Mech. Eng. 122*, 223–250.

Datta-Gupta, A. & King, M. J. (1995). A semianalytic approach to tracer flow modeling in heterogeneous permeable media. *Adv. in Wat. Res. 18*, 9–24.

Dawson, C. N. (1991, October). Godunov-mixed methods for advective flow problems in one space dimension. *SIAM J. Num. Anal. 28*(5), 1282–1309.

Douglas, J., Furtado, F., & Pereira, F. (1997). On the numerical simulation of waterfloding of heterogeneous petroleum reservoirs. *Computational Geosciences 1*, 19–48.

Espedal, M. S. & Karlsen, K. H. (1999). Numerical solution of reservoir flow models based on large time step operator splitting algorithms. In A. Fasano & H. van Duijn (Eds.), *Springer Lecture Notes in Mathematics: Filtration in Porous Media and Industrial Applications*. Springer.

Goode, D. (1990). Particle velocity interpolation in block-centered finite difference groundwater flow models. *Water Resources Research 28*(5), 925–940.

Herrera, I., Ewing, R. E., Celia, M. A., & Russell, T. F. (1993). Eulerian-lagrangian localized adjoint methods: the theoretical framwork. *Numerical Methods for PDE's 9*, 431–458.

Hewett, T. A. & Yamada, T. (1997). Theory for the semianalytical calculation of oil recovery and effective relative permeabilities using streamtubes. *Adv. in Wat. Res. 20*, 279–292.

Holden, H., Holden, L., & Høegh-Krohn, R. (1988). A numerical method for first order nonlinear scalar conservation laws in one-dimension. *Comput. Math. Applic. 15*(6–8), 595–602.

Karlsen, K. H., Brusdal, K., Dahle, H. K., Evje, S., & Lie, K.-A. (1998). The corrected operator splitting approach applied to a nonlinear advection-diffusion problem. *Comput. Methods Appl. Mech. Eng. 167*, 239–260.

Karlsen, K. H. & Risebro, N. H. Corrected operator splitting for nonlinear parabolic equations. *SIAM J. Num. Anal.*. To appear.

Karlsen, K. H. & Risebro, N. H. (1997). An operator splitting method for convection-diffusion equations. *Numer. Math. 77*(3), 365–382.

King, M. J. & Datta-Gupta, A. (1998). Streamline simulation: A current perspective. *In Situ (Special Issue on Reservoir Simulation) 22*(1), 99–139.

Lu, N. (1994). A semianalytical method of path line computation for transient finite-difference groundwater flow models. *Water Resources Research 30*(8), 2449–2459.

Morton, K. (1996). *Numerical Solution of Convection-Diffusion Problems*. London, UK: Chapman & Hall.

Pollock, D. W. (1988). Semianalytical computation of path lines for finite difference models. *Groundwater 26*(6), 743–750.

Russell, T. F. & Healy, R. W. (2000). Analytical tracking along streamlines in temporally linear raviart-thomas velocity fields. In *Proc. 11th Int. Conf. on Comp. Meth. in Water Resources, Calgary, Canada, June 2000*.

Toro, E. F. (1999). *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction* (2nd ed.). Germany: Springer.

Wang, H., Dahle, H. K., Espedal, M. S., Ewing, R. E., Sharpley, R. C., & Man, S. (1999). An ELLAM scheme for advection-diffusion equations in two dimensions. *SIAM J. Sci. Comput. 20*, 2160–2194.

Paper C

A streamline front tracking method for two- and three-phase flow including capillary forces*

* Submitted to the Proceedings of the Joint Summer Research Conference on Fluid Flow and Transport in Porous Media, fall 2001.

Contemporary Mathematics

A Streamline Front Tracking Method for Two- and Three-Phase Flow Including Capillary Forces

Inga Berre, Helge K. Dahle, Kenneth H. Karlsen, and Hans F. Nordhaug

1. Introduction/Outline

Currently, there seems to be a trend within the petroleum industry to use fast and accurate methods to simulate reduced sets of equations. Typically streamline and/or front tracking methods are used to solve the hyperbolic Buckley-Leverett equation for two-phase flow [**8, 12, 14**]. This allows for estimates of reservoir performance on more complete geological data, possibly without using upscaling. However, important physical phenomena are neglected which may greatly influence the overall fluid distribution.

In this work we consider models of multi-phase flow which do include capillary forces. We also allow for three phases. In particular we shall investigate a streamline front tracking method (SFTM) [**1**]. This method is based on calculating streamlines locally around grid points. The hyperbolic part of the problem is then calculated using front tracking. Since the front tracking method requires a Riemann solver, we do currently only consider a triangular model for three phase flow, i.e., the advective flux of the gas phase does only depend on gas-saturation and not on the other saturations. To account for capillary effects we use operator splitting [**6**]. The hyperbolic part of the solution is projected onto a regular grid, and a complete solution in each time step is obtained by solving a parabolic equation.

The paper is organized as follows: In Section 2 we state a standard model for two- and three-phase flow. In Section 3 the streamline front tracking method (SFTM) [**1**] is developed. A modified method of characteristics (MMOC) [**5**] is also briefly discussed and we we give an outline of a fast marching method (FMM) [**7**]. These methods are used in the two-phase case as comparisons with the SFTM-method proposed here. In Section 4 we discuss a possible extension of the SFTM-method to three phase flow. This extension is based on a triangular approximation of the fully coupled three-phase problem, and the so-called H-set method is used to solve associated Riemann problems. Some consequences of the triangular approximation are also discussed. Finally, in Section 5 we present some numerical experiments for two- and three-phase flow to demonstrate the methodology, and a summary and conclusions are given in Section 6.

2000 *Mathematics Subject Classification.* Primary 65M99; Secondary 76S05.

This work is partly supported by the Norwegian Research Council (NFR) under grant 116153/431.

©2000 American Mathematical Society

2. Governing equations

The basic equations describing multi-phase immiscible flow in a porous medium, say water (w), gas (g) and oil (o), is given by mass balance equations and Darcy's law. Assuming that the flow is incompressible, and that gravity can be neglected, the equations can be written in a global pressure/total velocity formulation, see [**3, 4**], as follows:

$$\nabla \cdot \mathbf{v} = q(\mathbf{x}, t), \tag{1}$$

$$\mathbf{v} = -\lambda_T(\mathbf{x}, S_\alpha)\mathbf{K}(\mathbf{x}) \cdot \nabla p, \tag{2}$$

$$\phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot [F_\alpha \mathbf{v} - \epsilon_\alpha \nabla \cdot D_\alpha(\mathbf{x}, S_\alpha) \nabla S_\alpha] = q_\alpha(\mathbf{x}, t). \tag{3}$$

Here ϕ and $\mathbf{K}$ are the porosity and absolute permeability of the porous medium; S_α, $\mathbf{v}_\alpha$, $k_{r\alpha}$ and μ_α are, respectively, the reduced saturation, Darcy velocity, relative permeability, and viscosity of phase α; p is a global pressure derived from the phase pressures and the capillary pressures, see [**4**]; $\lambda_\alpha = k_{r\alpha}/\mu_\alpha$ denote the mobility ratio, $\lambda_T = \sum_\alpha \lambda_\alpha$ is the total mobility, and $F_\alpha = \lambda_\alpha/\lambda_T$ is the fractional flow of phase α; $\mathbf{v} = \sum_\alpha \mathbf{v}_\alpha$ is the total velocity; D_α is a simplified capillary diffusion/dispersion term and ϵ_α is (typically) a small parameter which gives the relative importance of advective and capillary/dispersive forces; finally q, q_α account for injection and production wells. An additional constraint is given by $\sum_\alpha S_\alpha = 1$, and we note that $\sum_\alpha F_\alpha = 1$. To close the above system we need constitutive relationships for the capillary pressures and relative permeabilities. We shall assume that the concentration of gas in oil phase is constant (no phase transfer), which means that do not have to calculate phase pressures. Thus, it suffices to specify D_α directly rather than capillary pressures. In the experiments reported below we shall use $D_\alpha(\mathbf{x}, S_\alpha) = \mathbf{K}$. For the relative permeabilities use quadratic forms $k_{r\alpha} = S_\alpha^2$, combined with Stone models for the relative permeability of oil in the three-phase case, see Section 5. In the numerical experiments, we consider a quarter-of-a-five-spot problem. Thus, no-flow conditions are imposed on the boundaries.

3. Solution Strategy

To decouple the Pressure/Velocity equations (1), (2), from the saturation equations (3), we use sequential time stepping. Thus, for a given saturation-field, say at time t^n, we calculate a new velocity field. The saturation field is then advanced to a new time-step t^{n+1} by solving (3), using the most recent velocity field. This is continued sequentially up to a predetermined time $t = T$. To recover conservative and accurate fluxes we have used a control volume finite element method [**5**]. From these fluxes we obtain a velocity field in the lowest order Raviart-Thomas space RT_0, on a regular Cartesian grid.

To solve the parabolic saturation Equation (3) with a given velocity field, we again use operator splitting. First the purely hyperbolic part of Equation (3) is solved to advect the solution up to a new time level, then diffusion/dispersion is accounted for by solving a heat type equation on a regular grid, see [**6**] and references therein.

In previous work a Modified-Method-of-Characteristics (MMOC) has been used to solve the two-phase flow problem based on the ideas outlined above, see [**5**]. This method works excellent when the wave structure of the solution is known *a priori*. For an established front it has been shown that when the front is resolved this method gives the correct front-width of $O(D/b)$, where D and b is respectively an estimate of the diffusion and the residual of the linearized fractional flow function. Below we present two alternative methods to the MMOC method, which both preserve the shape of self-sharpening fronts and are

more flexible than the MMOC method in the sense that no a priori knowledge of the wave structure is required. However, the fast marching method presented below is based on the restriction that the initial data are monotone and that level sets propagate outwards from their source.

3.1. A streamline front tracking method (SFTM). For simplicity we assume that the computational domain is discretized by a regular Cartesian grid such that the velocity $\mathbf{v} = \mathbf{v}(\mathbf{x}) \in RT_0$ is given. Furthermore, assume that approximate saturation values S^n is known at the cell centers at time-level t^n. Saturation values refer to water-saturation S_w in case of two-phase flow, and water- and gas-saturation S_w, S_g in case of three-phase flow. To obtain saturation values S^{n+1} at time-level t^{n+1}, we split Equation (3) into a hyperbolic part:

$$S_t + \mathbf{v} \cdot \nabla F(S) = 0, \tag{4}$$

and a parabolic heat type equation

$$S_t = \epsilon \nabla \cdot (D \nabla S). \tag{5}$$

In a standard operator-splitting fashion, equation (3) is solved in two steps: First, Equation (4) is solved with S^n as initial condition to obtain a first approximation, say $\bar{S}^{n+1}$ at the next time-level. Then, saturation values at the next time-level are obtained by solving (5), with $\bar{S}^{n+1}$ as initial condition. Now, consider the solution of Equation (4): Observe that on streamlines $\mathbf{r} = \mathbf{r}(\xi)$ such that

$$\frac{d\mathbf{r}}{d\xi} = \mathbf{v}, \tag{6}$$

Equation (4) becomes one-dimensional:

$$S_t + F_\xi(S) = 0. \tag{7}$$

We exploit this to obtain new saturation-values at cell centers $\mathbf{x}_I$, in the following way: First, trace streamlines (6) analytically for $\mathbf{r}(0) = \mathbf{x}_I$ and $-\xi_{\max} < \xi < \xi_{\max}$. Here $\xi_{\max} = |\lambda_{\max}|(t^{n+1} - t^n)$ with $\lambda_{\max}$ being an estimate of the maximum wave speed of the system, such that the streamline cover the domain of dependence for $(\mathbf{x}_I, t^{n+1})$. The streamline is only traced in the upstream direction if all the wave speeds are positive. The piecewise constant cell values of the saturations are then projected onto these local streamlines, thus defining piecewise constant initial conditions for Equation (7). This conveniently arrange for Equation (7) to be solved by the front tracking method: For a two-phase (scalar) problem the flux function F, is simply replaced by a piecewise linear approximation F^δ. We then solve the Riemann problem associated with each of the initial discontinuities. Since F^δ is piecewise linear, the rarefaction part of the solution is approximated by discontinuities. Thus, the solutions will always consist of a set of fronts traveling with distinct speeds. We then have to keep track of the position of each such front, and solve new Riemann problems whenever fronts collide, thereby obtaining a global solution $\bar{S}$ for any time $t > t^n$ on the streamline. In particular we will obtain approximate saturation values $\bar{S}^{n+1}$ at the cell centers. We refer to [**11**] for more details about the front tracking method.

The main advantages of the front tracking method, is that the method is *super fast* [**11**], and preserves the frontal structure of the solutions extremely well. On the other hand, since the method heavily depends on solving Riemann problems, it is not easy to extend the method to three-phase flow. In Section 4, we will discuss a solution strategy for so-called triangular systems which may be a step towards this end. Equation (5) may be solved by finite element or finite difference type of methods. In this work we have for

convenience used a standard explicit central finite difference scheme. Note that a local time step, $\Delta t_{diff} \leq t^{n+1} - t^n$, is required to satisfy the stability constraint inherent in the explicit finite difference method.

3.2. A fast marching method (FMM). An alternative to the front-tracking approach is the fast marching method for reservoir simulation developed in [**7**]. This methodology has shown to give very fast and accurate results for Buckley-Leverett type problems. Here we extend the work in [**7**] by including diffusion/dispersion-effects using operator splitting as described above. To outline this procedure we consider the hyperbolic part of the transport equation (3):

$$\frac{\partial S_w}{\partial t} + \mathbf{v} \cdot \nabla F_w(S_w) = 0, \qquad S(\mathbf{x}, 0) = S_0(\mathbf{x}).$$

We will assume that the initial function S_0 is monotone, and that the velocity field is so that the level-sets of the saturation function never pass a point twice. This is not a restrictive assumption for the reservoir problem, since the flow will be directed from injection wells towards production wells. Following [**7**], we consider a level set $S(x,t) = k$ of the saturation function. Let the front be defined by $\Xi_k(t) = \{\mathbf{x} : S(\mathbf{x},t) = k\}$. According to the classical method of characteristics, the front will move with a characteristic speed given by $\mathbf{V}(\mathbf{x}) = \frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x})F'(k)$. Let $T_k(\mathbf{x})$ be the time $\Xi_k(t)$ crosses the point $\mathbf{x}$. The arrival time satisfies the Eikonal equation

$$\nabla T_k \cdot \mathbf{v}(\mathbf{x}) = \frac{1}{F'(k)}, \qquad F'(k) > 0. \tag{8}$$

The evolution problem for the propagating front is hence reformulated in an Eulerian framework.

The equation (8) can be solved rapidly by means of the fast marching method (see, e.g., [**17, 16**]). Since the flow is directed from injection towards production wells, information will flow in one direction, that is, from regions with smaller to regions with larger arrival times. Hence, the arrival time at a grid point can only depend on points with smaller values. This fact motivates an upwind discretization for the Eikonal equation. For the comparison results presented later we have applied a first order 9-point scheme based on linear approximation of streamlines locally through each grid point. When the Eikonal equation is solved to a prescribed final time, fronts corresponding to larger values of S may have crossed fronts corresponding to smaller values of S. This corresponds to the development of shocks. However, since the level sets have been transported according to the characteristic speed, an approximation to the correct entropy weak solution can be obtained from the multivalued solution by means of the collapse transform, i.e., a proper vertical averaging of the multivalued solution, see [**2**], $S(\mathbf{x}) = \sum_{i=0}^{2p(\mathbf{x})} (-1)^i \hat{S}_i(\mathbf{x})$. Here the multivalued solution $\hat{S}_i$ is obtained by considering the positions of the level sets. The resulting method is not subject to CFL-conditions. The method described in this section will be described in detail in a forthcoming paper.

4. Triangular systems

A possible extension of the SFTM approach to three-phase flow, is to use an approximate Riemann solver to generate front speeds. However, to our knowledge, the construction of accurate and reliable Riemann solvers for fully coupled three-phase flow is not a trivial task, and we have chosen a different approach here: Observe that the viscosity of the gas phase is usually at least an order less than the viscosities of the liquid phases.

Motivated by this fact it seems reasonable to assume that the fractional flow of the gas-phase can be approximated by a flux function which only depends on the saturation of the gas phase. Thus, we may consider the following 2×2-triangular hyperbolic system, as an approximation to the hyperbolic part of the fully coupled system (3):

$$\frac{\partial S_g}{\partial t} + \frac{\partial}{\partial x} F_g(S_g) = 0, \tag{T1}$$

$$\frac{\partial S_w}{\partial t} + \frac{\partial}{\partial x} F_w(S_g, S_w) = 0. \tag{T2}$$

Systems of this type has been investigated in [**9, 10, 13**], and existence and uniqueness of the solution to the Riemann problem is shown under general conditions.

4.1. A Riemann solver for triangular systems. The numerical construction of a solution for a Riemann problem associated with equations (T1)-(T2) was developed by Gimse [**9**], and is rather involved. The idea is to solve (T1) first using the front tracking method. Thus, the approximate solution to (T1) consists of a set of constant states, say $S_g^L = S_g^1 < S_g^2 < \ldots < S_g^{N+1} = S_g^R$, separated by jump discontinuities traveling with the Rankine-Hugoniot shock speed:

$$s_i = \frac{F_g(S_g^{i+1}) - F_g(S_g^i)}{S_g^{i+1} - S_g^i}, \ \ i = 1, 2, \ldots, N. \tag{9}$$

Within each wedge of the solution fan, the fractional flow of the water phase depends only on the water saturation and is given as:

$$F_w^i(S_w) \stackrel{def}{=} F_w(S_g^i, S_w), \ \ i = 1, 2, \ldots, N+1.$$

Thus, we may easily solve for the water saturation within each wedge once we know the left- and right-hand state of the water saturation within the wedge. Obviously, successive left- and right-hand states over the discontinuities in the gas-phase must also satisfy the jump condition:

$$\frac{F_w^{i+1}(S_w^{i+1}) - F_w^i(S_w^i)}{S_w^{i+1} - S_w^i} = s_i, \ \ i = 1, 2, \ldots, N, \tag{10}$$

There are infinitely many states S_w^i, which satisfy conditions (10), leading to the definition of the so-called H-sets:

- $H_{1,in}$**:** The set of S_w-values in region S_g^L, that can be reached from S_w^L with speed $\sigma \leq s_1$.
- $H_{i,in}$**:** The set of S_w-values in region S_g^i, $i = 2, 3, \ldots N$, that can be reached from $H_{i,out}$ with speed σ such that $s_{i-1} \leq \sigma \leq s_i$.
- $H_{i+1,out}$**:** The set of S_w-values in region S_g^{i+1}, $i = 1, 2, \ldots N+1$, that can be reached from $H_{i,in}$ by a shock with speed s_i.

The solution can now be assembled from these sets by connecting S_w^R to S_w^L by tracking through admissible waves given by the H-sets. This procedure is described schematically in the following diagram:

$$\begin{aligned} S_w^R \to H_{n+1,out} \to H_{n,in} \to \cdots \\ \to H_{i+1,out} \to H_{i,in} \to \cdots \\ \to H_{2,out} \to H_{1,in} \to S_w^L \end{aligned}$$

Note that a jump from one set to the next always happens at the first possible value in the current H-set.

Gimse has shown [**9**] that the above construction gives a *unique* solution of equation (T2) if the following conditions hold:

$$(A)\ \ F_g(0) = 0, \quad (B)\ \ F_g(1) = 1, \quad (C)\ \ F_w(S_g, 1 - S_g) = 1 - F_g(S_g)$$

$$(D)\ \ \tfrac{\partial F_w}{\partial S_w} \geq 0, \qquad (E)\ \ \tfrac{\partial F_w}{\partial S_g} < 0, \qquad (F)\ \ F_g'(S_g) \geq 0.$$

Conditions (A) and (B) are always satisfied if the transport equations are properly scaled. Conditions (D) and (F) are usually satisfied if gravity is neglected. Condition (E) guarantees that successive F_w^i do not cross each other. This eventually simplifies the construction of the H-sets. However, it has been shown that condition (E) is not necessary for the existence and uniqueness of the solution , and can therfore be removed as a necessary condition. Finally, condition (C) guarantees that the endpoints of successive F_w^i, are connected by correct slopes given by the shock speeds s_i. Endpoints are here defined as points where oil is at residual saturation $S_o = 0$, or $S_g + S_w = 1$. Note that this condition is always satisfied for *genuinely* triangular systems (i.e, $F_g \equiv F_g(S_g)$), since $F_o = 0$ at residual oil-saturation $S_o = 0$.

4.2. Can physical three-phase transport equations be approximated by a triangular system? Since the fractional-flow of the gas-phase is nearly independent of the water saturation it seems natural to decouple the gas-phase from the other phases by simply plug in a value for the water-saturation, S_w^0, in the gas fractional-flow:

$$F_g(S_w, S_g) \approx F_g(S_w^0, S_g) \stackrel{\text{def}}{=} F_g(S_g). \tag{11}$$

Possibly, we can choose S_w^0 so that the decoupled fractional-flow function is as close as possible to the complete function in some norm over the admissible section of state space. However, S_w^0 is chosen more or less arbitrary to be $S_w^0 = 0$ in the following experiments. In Figures 3 and 4 we compare the solution of the triangular system, with the solution of the full system in two different one dimensional cases. In Case 1 we get excellent agreement between the two solutions. In Case 2 the full solution for the gas-phase consists of two rarefaction waves connected by an intermediate shock wave. This wave structure cannot be captured by the triangular system with a simplified fractional-flow function as given by (11). A possible solution to this problem can be to let S_w^0 be defined locally for each Riemann problem to be solved, which will allow for some feedback. This can be combined with the fact that the total mobility λ_T act as an approximate invariant for the transport. Thus $\lambda_T(S_w, S_g) \approx \lambda_T(S_w^0, S_g)$ may be used to eliminate S_w from F_g in a more accurate way. However, we will not pursue the idea further here.

Another, difficulty that arises from the approximation (11), is that condition (C) is violated since

$$F_w(1 - S_g, S_g) = 1 - F_g(1 - S_g, S_g) \neq 1 - F_g(S_w^0, S_g).$$

If condition (C) is not satisfied the construction of the solution may fail in two ways close to values where $S_o = 0$: Either the construction of appropriate H-sets will fail, or the tracking of admissible waves becomes impossible. Since $S_o \approx 0 \implies S_w \approx 1 - S_g$, we may circumvent the problem by replacing S_w^0 with $1 - S_g$ in (11) when the oil phase is close to residual. Again this require a local definition of S_w^0 which has not been implemented in our codes yet. For the robustness of the method this needs to be done.

5. Numerical experiments

5.1. Two-phase flow: To obtain a comparison between the different methods we first ran a series of experiments with a two-phase flow problem for a quarter-of-a-five-spot problem with noflow conditions at the boundaries. The implementation has been done in two space dimensions. The permeability field in these experiments was generated from a (log-normal?) distribution and is plotted in Figure 1, together with typical flow directions. The permeability contrast in this figure is a factor of 10. To avoid difficulties at the corners the initial saturation is given by:

$$S_0(r) = \begin{cases} 1-(r/0.3)(1-\sqrt[-1]{2}), & r \le 0.3, \\ 0, & r > 0.3, \end{cases} \tag{12}$$

where r denote the radial distance from the lower left corner and $\sqrt[-1]{2}$ is the shock saturation of the Buckley-Leverett profile. In Figure 2, saturation contours are plotted after four time steps for one of the experiments performed. Observe that the results are in excellent agreement. Also note that the three methods allow for very long time steps.

The complexity of the MMOC method is comparable with the SFTM approach, whereas the FMM method gives a much faster advection solver. In Table 1, we have attempted to compare the efficiency of these methods. To do so we fixed a velocity field and did a pure advection of the saturation front up to four time steps. The results in the table indicates that when the number of nodes is increased by a factor 4, the run time increases by a factor 4 for the fast marching method and a factor of 6.5 for the front tracking method. Thus, the fast marching method is optimal in this sense, whereas the front tracking method becomes sub optimal because the number of fronts increase on each local streamline, in addition to the increase of local streamlines.

5.2. Three-phase flow: In figures 3 and 4 we have compared solutions of triangular approximations with the solutions of corresponding full three phase flow problems in one space dimension. Below follows mobilities, fractional flows for the triangular approximations and initial conditions:

Case 1:

$$\lambda_g(S_g) = S_g^2, \qquad \lambda_w(S_w) = S_w^2/10, \qquad \lambda_o(S_o) = S_o^2/10.$$

$$F_g(S_g) = \frac{\lambda_g(S_g)}{\lambda_{\text{tot}}(S_g,0)}, \qquad F_w(S_g,S_w) = \frac{\lambda_{\text{tot}}(1-S_g,0)}{\lambda_{\text{tot}}(S_g,1-S_g)} \cdot \frac{\lambda_w(S_w)}{\lambda_{\text{tot}}(S_w,0)},$$

$$S_g^0 = \begin{cases} 0.6 & \text{if } x < 0.4, \\ 0.2 & \text{if } x > 0.4, \end{cases} \text{ and } S_w^0 = \begin{cases} 0.3 & \text{if } x < 0.4, \\ 0.7 & \text{if } x > 0.4, \end{cases}$$

Case 2:

$$\lambda_g(S_g) = 50S_g, \qquad \lambda_w(S_w) = S_w^3, \qquad \lambda_o(S_o) = 3.14S_o(1-S_w)^2(1-S_g)^2.$$

$$F_g(S_g) = \frac{\lambda_g(S_g)}{\lambda_{\text{tot}}(S_g,0)}, \qquad F_w(S_g,S_w) = \frac{\lambda_w(S_w)}{\lambda_{\text{tot}}(S_g,S_w)}.$$

$$S_g^0 = \begin{cases} 0.25 & \text{if } x < 0.4, \\ 0.1 & \text{if } x > 0.4, \end{cases} \text{ and } S_w^0 = \begin{cases} 0.75 & \text{if } x < 0.4, \\ 0.0 & \text{if } x > 0.4. \end{cases}$$

A standard upwind difference scheme is used to generate the reference solutions for the fully coupled system. In Case 1 the triangular flow functions are carefully constructed

to satisfy the conditions (A-F), see [**15**] for further details. In Case 2, only the flow functions for gas is changed and as a result condition (C) is not satisfied. We are generally not guaranteed that a solution exists, but for the chosen initial value problem it did. We note that the shape of these functions differ significantly from the full three-phase model, especially for Case 1. From these one dimensional examples we conclude that the triangular model may and may not give a good approximation to a fully coupled three-phase flow problem.

To investigate the robustness of the SFTM method we have constructed a two dimensional test example. To make sure that the conditions (A-F) are satisfied the fractional-flows of Case 1 have been used, and we use the same permeability field and boundary conditions as for the two-phase problem of Section 5.1. Initially the saturation for the water is given by Equation (12), and the remaining volume at any point is partitioned into 10% gas and 90% oil. The result is shown in Figure 5 and appears to be much more diffused than the solutions of the two-phase problem. However, as seen from Figure 3, the shape of the solution is really due to the shape of the fractional flows. Again we note long and stable time steps are used.

6. Summary/conclusions

We have presented a front tracking streamline method (SFTM) for multi-phase flow in porous media. The main advantage of this method is to handle the advective part of nonlinear transport using streamline information, and still be able to solve for diffusive/dispersive effects on a regular grid. The method has been compared with a Modified Method of Characteristics (MMOC) and a Fast Marching (FMM) approach for two-phase flow problems. The solutions obtained seems to be equally accurate. The SFTM and the MMOC are comparable in computational efficiency, whereas the FMM has a much faster advection solver. However, compared to the MMOC and the FMM, the SFTM is more flexible and has less restrictions with respect to the complexity of the problems that may be solved.

Using the H-set method the SFTM approach has been extended to solve three-phase flow problems which are triangular. A 2×2 system is triangular if one of the fractional-flows only depend on one of the saturations, whereas the other depends on both saturations. Since most three-phase flow problems are fully coupled in both saturations, triangular systems can only be approximate. However, because the viscosity of the gas is much smaller than the viscosities of the liquid phases the gas-phase is often nearly decoupled from the other phases. We have shown that a *naive* triangular approximation of a fully coupled system may and may not reproduce good approximations to the solutions of the complete system. We have also suggested ways to improve the triangular approximation when this gives poor comparisons with a full three-phase flow system.

Acknowledgments

We are grateful to Knut Andreas Lie and Jostein Natvig for letting us use parts of their computer code.

References

[1] A.K. Ask, H.K Dahle, K.H. Karlsen, and H.F Nordhaug, *A local streamline eulerian-lagrangian method for two-phase flow*, Computational Methods in Water Resources (L.R Bentley, J.F Sykes, C.A Brebbia, W.G gray, and Pinder G.F, eds.), vol. 2, A.A Balkema, 2000, pp. 645–650.

[2] Y. Brenier, *Averaged multivalued solutions for scalar conservation laws*, SIAM J. Numer. Anal. **21** (1984), 1013–1037.

	Grid	Time		Scaling	
		FMM	SFTM	FMM	SFTM
Homogeneous	33×33	0.09	0.14	–	–
	65×65	0.38	0.89	4.22	6.36
	129×129	1.59	5.93	4.18	6.66
	257×257	6.98	43.30	4.39	7.30
Heterogeneous	33×33	0.10	0.17	–	–
	65×65	0.39	1.08	3.90	6.35
	129×129	1.56	7.52	4.00	6.96

TABLE 1. Comparison of run times for the advection solver using the FMM and the SFTM, with $\Delta t = 0.5$ and $T = 2.0$.

[3] G. Chavent and J. Jaffre, *Mathematical models and finite elements for reservoir simulation*, Studies in mathematics and it's applications, vol. 17, North Holland, Amsterdam, 1986.

[4] Z. Chen and R.E. Ewing, *Comparison of various formulations of three-phase flow in porous media*, Journal of Computational Physics **132** (1997), 362–373.

[5] H. K. Dahle, M. S. Espedal, and O. Sævareid. Characteristic, local grid refinement techniques for reservoir flow problems. *International Journal for Numerical Methods in Enginering*, 34:1051–1069, 1992.

[6] M. S. Espedal and K. H. Karlsen. Numerical solution of reservoir flow models based on large time step operator splitting algorithms. In *Filtration in Porous Media and Industrial Applications (Cetraro, Italy, 1998)*, volume 1734 of *Lecture Notes in Mathematics*, pages 9–77. Springer, Berlin, 2000.

[7] K. H. Karlsen, K.-A. Lie, and N. H. Risebro. A fast marching method for reservoir simulation. *Computational Geosciences*, 4(2):185–206, 2000.

[8] A. Datta-Gupta and M. J. King, *A semianalytic approach to tracer flow modeling in heterogeneous permeable media*, Adv. in Wat. Res. **18** (1995), 9–24.

[9] T. Gimse, *A triangular riemann solver.*, Master's thesis, Dept. of Mathematics, Univ. of Oslo, 1988.

[10] Tore Gimse, *A numerical method for a system of equations modelling one-dimensional three-phase flow in a porous medium*, Nonlinear hyperbolic equations—theory, computation methods, and applications (Aachen, 1988), Vieweg, Braunschweig, 1989, pp. 159–168.

[11] L. Holden H. Holden and R. Høegh-Krohn, *A numerical method for first order nonlinear scalar conservation laws in one dimension.*, Comp. Math. Appl. **15** (1988), 595–602.

[12] T. A. Hewett and T. Yamada, *Theory for the semi-analytical calculation of oil recovery and effective relative permeabilities using streamtubes*, Adv. in Wat. Res. **20** (1997), 279–292.

[13] L. Holden and R. Høegh-Krohn, *A class of n nonlinear hyperbolic conservation laws.*, Journal of Differential Equations **84** (1990), 73–99.

[14] M. J. King and A. Datta-Gupta, *Streamline simulation: A current perspective*, In Situ (Special Issue on Reservoir Simulation) **22** (1998), no. 1, 99–139.

[15] J. A. Kok, *Front tracking for three-phase flow in porous media.*, Final report of the postgraduate program Mathematics for industry. Eindhoven, Netherlands, 1994.

[16] J. A. Sethian. *Level set methods*. Cambridge University Press, Cambridge, 1996. Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science.

[17] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, 1995.

DEPT. MATH., UNIV. BERGEN, JOHS. BRUNSGT. 12, N-5008 BERGEN, NORWAY

E-mail address: ingab@mi.uib.no, reshd@mi.uib.no, kennethk@mi.uib.no, hansfn@mi.uib.no

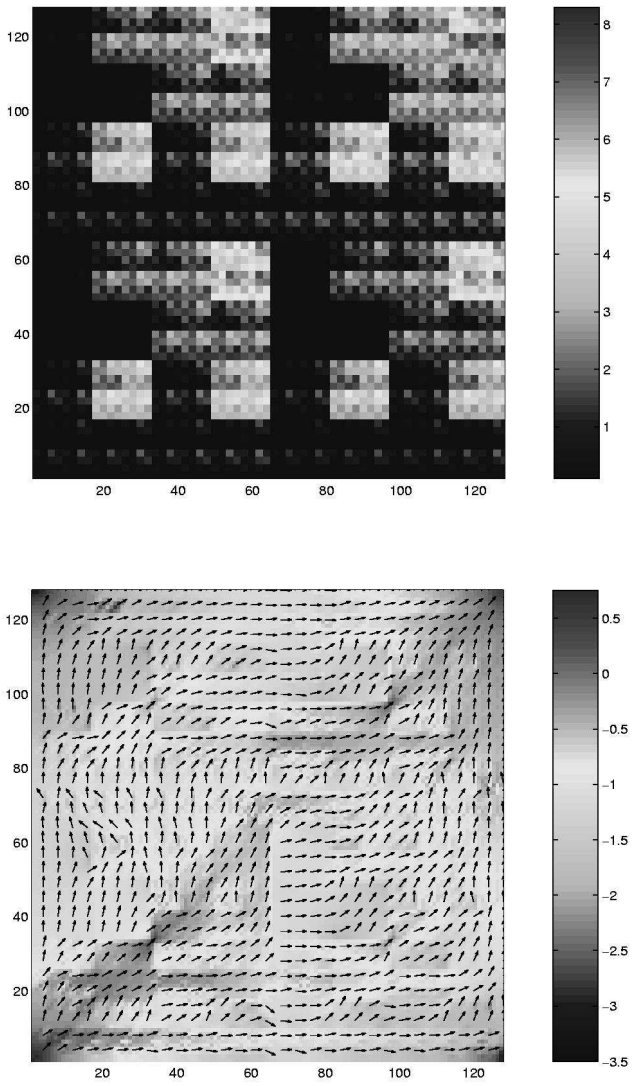


FIGURE 1. Permeability and velocity fields.

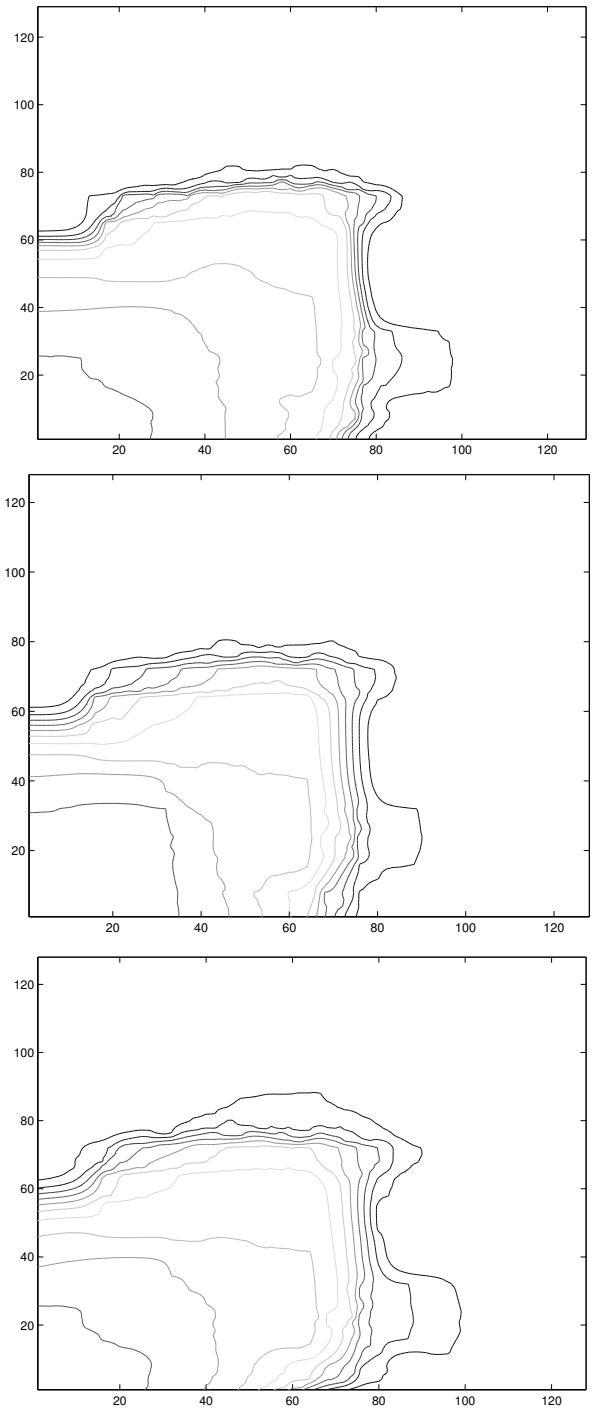


FIGURE 2. Comparisons between the MMOC (upper), the FMM (middle) and the SFTM (lower) on 129 × 129 grid. $\Delta t = 0.5$, $\epsilon = 10^{-2}$ and $T = 2.0$.

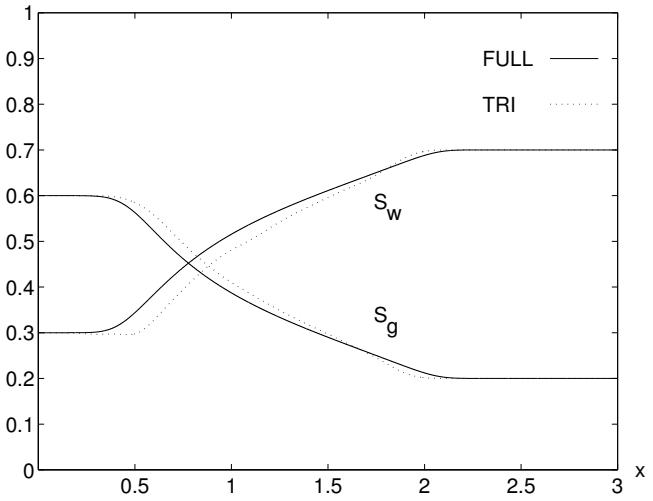


FIGURE 3. Case 1: Comparison between the solutions (gas- and water saturations) of the fully coupled system and the corresponding triangular system. $\Delta t = 0.25$, $T = 0.5$ and $\epsilon = 0.01$. $\Delta x = 0.0005$ for the difference scheme (FULL).

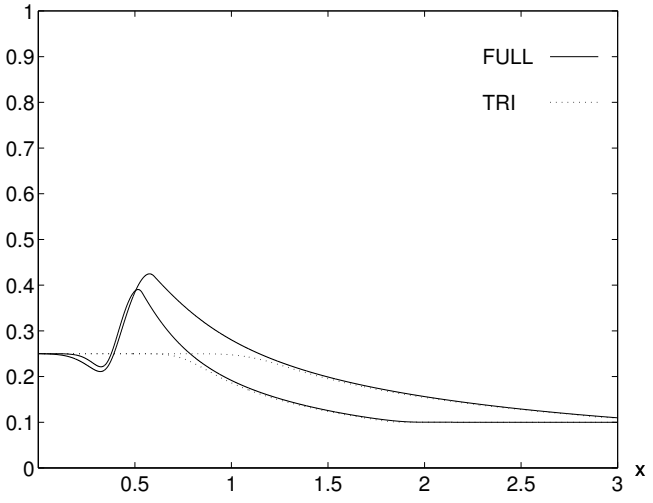


FIGURE 4. Case 2: Comparison between the solutions (only gas-saturations) of the fully coupled system and the corresponding triangular system. saturation, S_w. $\Delta t = 0.25$, $T = 0.5$, 1.0 and $\epsilon = 0.01$. $\Delta x = 0.0005$ for the difference schemes (FULL).

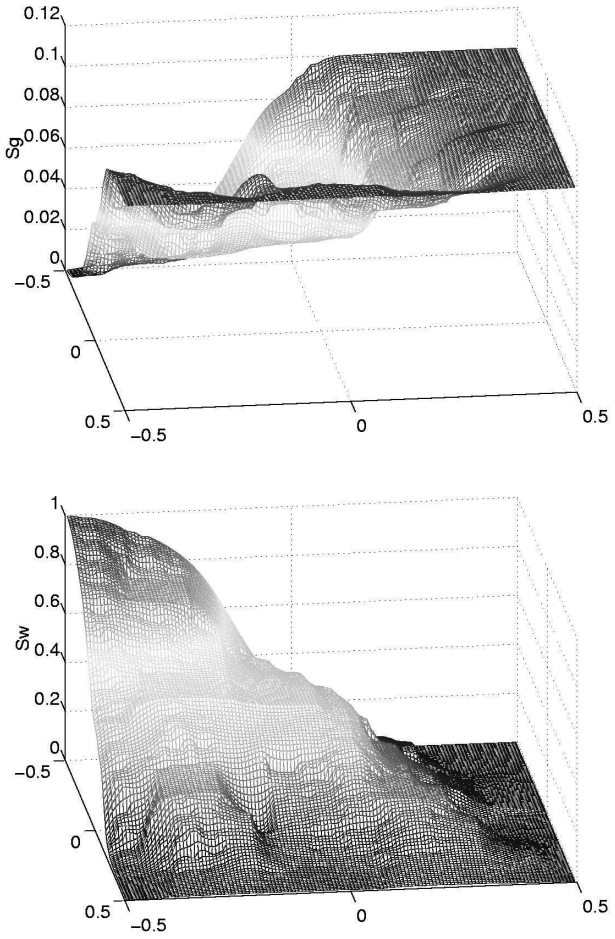


FIGURE 5. Solution of a quarter of a five spot problem for a triangular system, gas saturation (upper) and water saturation (lower) on 129×129 grid. $\Delta t = 0.5$, $\epsilon = 10^{-2}$ and $\boldsymbol{T} = 2.0$.

Paper D

Two phase flow including interfacial area as a variable *

* Published in Computational Methods in Water Resources (L.R Bentley, J.F Sykes, C.A Brebbia, W.G gray, and Pinder G.F, eds.), vol. 1, A.A Balkema, 2000, pp. 231–238.

Two phase flow including interfacial area as a variable

Hans Fredrik Nordhaug, Helge K. Dahle and Magne S. Espedal
Department of Mathematics, University of Bergen, Norway

William G. Gray
Department of Civil Engineering and Geological Sciences, University of Notre Dame, USA

Michael A. Celia
Department of Civil and Environmental Engineering, Princeton University, USA

ABSTRACT: Based on the procedure of (Gray & Hassanizadeh 1998) we state macroscale conservation equations for multi phase flow in porous media including interfacial area as a variable. The phases we consider are a solid phase, a wetting phase and a non-wetting phase. Some modeling work is done to decide the functionality of the coefficients in the equations. In particular we construct constitutive relationships for the interfacial area as a function of saturation and capillary pressure, and the resistance terms. The final set of equations is reduced to a 2×2 system with the saturation and the interfacial area as independent variables. Some preliminary numerical experiments are reported.

1 INTRODUCTION

Multi phase flow has been intensively studied in fields like reservoir mechanics, ground water research and so on. Models in these fields are normally based on generalisations from one phase flow, e.g. Darcy's law is generalised through the introduction of relative permeability. In this work we take another approach. Conservation equations for a complete two phase model including all physical effects are used. These equations are derived through a general and systematic procedure that employs conservations equations at the microscale and thermodynamics analysis, see (Gray & Hassanizadeh 1998; Gray 1999; Gray 2000). This model is very complex and contains a large number of coefficients that must be specified for numerical computation. In this preliminary study, assumptions are made to reduce the number of variables and facilitate the study of the behaviour of the system of equations. In fact, the general set of equations is reduced to a 2×2 system that has saturation and interfacial area as the primary dependent variables.

There are three major reasons for this approach: First, when the complete system is simplified, there are no hidden assumptions, and hence it is easier to see what the limitations of the model are. Secondly, there may be practical reasons for calculating amount of interfacial area. Third, the inclusion of interfacial area may help in reducing apparent hysteresis in the model. Thus, the development of models determining the amount of interfacial are important. Both rely on the reasoning that it is better to simplify a general model than to generalise a simple model. In fact, hysteresis effects, e.g. in $P_c - s^w$ curves, may be a result of taking one phase flow models and generalising them directly to multi phase flow. On the other hand, these effects can be explained and eliminated through a more general approach.

In Section 2 the model and the conservation equations are stated. The basic assumptions are listed and simplifications made. The model is rewritten into a fractional flow formulation in Section 3. This is done for computational and analytical purposes. The resistances introduced in Section 2 is parameterised in Section 4, and capillary pressure is parameterised in Section 5. A numerical experiment is given in Section 6, and some conclusions and final remarks are given in Section 7.

2 CONCEPTUAL MODEL

The following terminology will be used:

a^s	specific area of the solid phase surface.
a^{wn}	specific interfacial area (fluid-fluid).
$a^{\alpha s}$	specific area of the αs-interface (fluid-solid).
$\mathbf{G}^{wn}$	geometric tensor.
J^w_{wn}	average curvature.
$\mathbf{K}$	permeability for the medium.
L_p, L_x	coefficients in linearised equations.
p^α	pressure of α-phase.
$\mathbf{R}^\alpha_{wn}$	resistance for the α-phase due to the wn-interface (fluid-fluid).
$\mathbf{R}^\alpha_{\alpha s}$	resistance for the α-phase due to the αs-interface (fluid-solid).
s^α	saturation of the α-phase.
$\mathbf{u}$	total Darcy velocity.
$\mathbf{u}^\alpha$	Darcy velocity for the α-phase.

$\mathbf{v}^{\alpha}$	velocity of α-phase.
$\mathbf{v}^{wn}$	velocity of the wn-interface.
x_s^{ws}	fraction of the solid surface covered by the wetting phase.
$\gamma^{\alpha\beta}$	surface tension of the $\alpha\beta$-interface.
ε	porosity of the medium.
$\varepsilon^{\alpha} = \varepsilon s^{\alpha}$	volume fraction of the α-phase.
μ^{α}	viscosity for the α-phase.
ρ^{α}	density of α-phase.
Φ^{w}	average contact angle.

2.1 *Conservation equations*

Based on the general theory in (Gray & Hassanizadeh 1998; Gray 2000) and the assumptions below we state a simplified set of equations modeling two phase flow in porous media that include interfacial area as a variable. The basic assumptions are:

- No phase change occurs,
- isothermal system,
- immobile, non-deforming solid,
- common lines are neglected,
- the solid-fluid interface dynamics are negligible (i.e no film flow),
- massless fluid-fluid interface,
- inertial terms are negligible in the momentum equations and
- interfacial tensions are all constant and specified.

Because the solid phase is immobile and non-deforming we have a two phase immiscible flow system where the wetting phase is denoted w and the non-wetting phase n.

In addition the solid is non-deforming so that $\mathbf{v}^s = 0$ and we get

$$\mathbf{v}^{\alpha,s} = \mathbf{v}^{\alpha} - \mathbf{v}^{s} = \mathbf{v}^{\alpha}, \quad \alpha = w, n, wn,$$

and

$$\frac{D^s}{Dt} = \frac{\partial}{\partial t} + \mathbf{v}^s \cdot \nabla = \frac{\partial}{\partial t}.$$

Note that bold capital and lowercase letters denote matrices and vectors, respectively.

From these assumptions, the following equations can be stated:

Mass conservation for the w-phase:

$$\frac{D^w(\varepsilon^w \rho^w)}{Dt} + \varepsilon^w \rho^w \nabla \cdot \mathbf{v}^w = 0. \tag{1}$$

Momentum conservation for the w-phase:

$$-\varepsilon^w \nabla p^w + \varepsilon^w \rho^w \mathbf{g} = (\mathbf{R}^w_{wn} + \mathbf{R}^w_{ws}) \cdot \mathbf{v}^w - \mathbf{R}^w_{wn} \cdot \mathbf{v}^{wn}. \tag{2}$$

Mass conservation for the n-phase:

$$\frac{D^n(\varepsilon^n \rho^n)}{Dt} + \varepsilon^n \rho^n \nabla \cdot \mathbf{v}^n = 0. \tag{3}$$

Momentum conservation for the n-phase:

$$-\varepsilon^n \nabla p^n + \varepsilon^n \rho^n \mathbf{g} = (\mathbf{R}^n_{wn} + \mathbf{R}^n_{ns}) \cdot \mathbf{v}^n - \mathbf{R}^n_{wn} \cdot \mathbf{v}^{wn}. \tag{4}$$

The momentum conservation equation for a massless interface indicates that the interfacial velocity is a weighted sum of the velocities of the adjacent phases. Here, for a massless wn-interface, the following equality is assumed:

$$(\mathbf{R}^w_{wn} + \mathbf{R}^n_{wn}) \cdot \mathbf{v}^{wn} = \mathbf{R}^w_{wn} \cdot \mathbf{v}^w + \mathbf{R}^n_{wn} \cdot \mathbf{v}^n. \tag{5}$$

Geometry equation for interfacial area:

$$\frac{\partial a^{wn}}{\partial t} - \varepsilon J^w_{wn} \frac{\partial s^w}{\partial t} - a^s \cos\Phi^w \frac{\partial x_s^{ws}}{\partial t} = -\nabla \cdot [\mathbf{G}^{wn} a^{wn} \cdot \mathbf{v}^{wn}]. \tag{6}$$

Dynamic capillary pressure equation:

$$\varepsilon \frac{\partial s^w}{\partial t} = -L_p[p^n - p^w + \gamma^{wn} J^w_{wn}]. \tag{7}$$

Linearised constitutive equation for surface area fraction:

$$a^s \frac{\partial x_s^{ws}}{\partial t} = -L_x[\gamma^{ws} - \gamma^{ns} + \gamma^{wn} \cos\Phi^w]. \tag{8}$$

The Darcy velocity for phase α is given by $\mathbf{u}^{\alpha} = \varepsilon^{\alpha}\mathbf{v}^{\alpha} = \varepsilon s^{\alpha}\mathbf{v}^{\alpha}$. We substitute these relationships into Equation (1)-(6). The velocity for the interface between the fluids, $\mathbf{v}^{wn}$, will be kept as it is.

To further simplify the set of equations we additionally assume incompressible flow. Equation (1) and (3) then reduces to

$$\varepsilon \frac{\partial s^w}{\partial t} + \nabla \cdot \mathbf{u}^w = 0, \tag{9}$$

$$\varepsilon \frac{\partial s^n}{\partial t} + \nabla \cdot \mathbf{u}^n = 0. \tag{10}$$

In some simple models, e.g. bundle of capillary tubes, it is reasonable to assume that there will be no production of wn-interfacial area as saturation changes. Therefore, the second and third terms of Equation (6) will cancel. Application of this condition, in general, is a significant assumption that limits the general applicability of the system of equations. However, it will be applied here consistent with the objective of performing a preliminary analysis of a simplified set of equations. This assumption eliminates the need to model the dynamics of the ws-interfacial area, Equation (8), and reduces Equation (6) to:

$$\frac{\partial a^{wn}}{\partial t} + \nabla \cdot [\mathbf{G}^{wn} a^{wn} \cdot \mathbf{v}^{wn}] = 0. \tag{11}$$

The transfer coefficient, L_p , in Equation (7) is also considered large so that Equation (7) reduces to:

$$\gamma^{wn} J^w_{wn} = p^w - p^n = -P_c. \tag{12}$$

Adding Equation (9) and (10) and observing that $s^w + s^n = 1$ gives

$$\nabla \cdot [\mathbf{u}^w + \mathbf{u}^n] = \nabla \cdot \mathbf{u} = 0$$

where

$$\mathbf{u}^w + \mathbf{u}^n = \mathbf{u} \tag{13}$$

is the total Darcy velocity. In 1D $\mathbf{u}$ is a constant.

We are now in the position to rewrite Equation (2), (4), (5), (9)-(12) to a 2×2 system of equations including fractional flow and capillary diffusion terms. This will be done in the following sections.

3 FRACTIONAL FLOW FORMULATION

From Equation (5) we get

$$\mathbf{v}^{wn} = \mathbf{A}^w \cdot \mathbf{u}^w + \mathbf{A}^n \cdot \mathbf{u}^n \tag{14}$$

where

$$\mathbf{A}^\alpha \stackrel{\text{def}}{=} (\mathbf{R}^w_{wn} + \mathbf{R}^n_{wn})^{-1} \cdot \mathbf{R}^\alpha_{wn}/\varepsilon^\alpha, \quad \alpha = w, n.$$

Using this expression for $\mathbf{v}^{wn}$ in Equation (2) and (4) we get

$$-\varepsilon^w \rho^w \nabla \psi^w = \mathbf{B}^w \cdot \mathbf{u}^w - \mathbf{R}^w_{wn} \cdot \mathbf{A}^n \cdot \mathbf{u}^n, \tag{15}$$

$$-\varepsilon^n \rho^n \nabla \psi^n = \mathbf{B}^n \cdot \mathbf{u}^n - \mathbf{R}^n_{wn} \cdot \mathbf{A}^w \cdot \mathbf{u}^w, \tag{16}$$

where $\psi^\alpha = p^\alpha/\rho^\alpha - gz$ and

$$\mathbf{B}^\alpha \stackrel{\text{def}}{=} (\mathbf{R}^\alpha_{wn} \cdot (\mathbf{I} - \mathbf{A}^\alpha) + \mathbf{R}^\alpha_{\alpha s})/\varepsilon^\alpha, \quad \alpha = w, n.$$

Define

$$\mathbf{C}^n \stackrel{\text{def}}{=} \mathbf{R}^w_{wn} \cdot \mathbf{A}^n \cdot (\mathbf{B}^n)^{-1},$$

$$\mathbf{C}^w \stackrel{\text{def}}{=} \mathbf{R}^n_{wn} \cdot \mathbf{A}^w \cdot (\mathbf{B}^w)^{-1}.$$

Add $\mathbf{C}^n$ times Equation (16) to (15), and $\mathbf{C}^w$ times Equation (15) to (16) to get $\mathbf{u}^\alpha$ expressed as a sum of the phase pressures:

$$\mathbf{u}^w = -(\mathbf{M}^w)^{-1} \cdot [\varepsilon^w \rho^w \nabla \psi^w + \varepsilon^n \rho^n \mathbf{C}^n \cdot \nabla \psi^n], \tag{17}$$

$$\mathbf{u}^n = -(\mathbf{M}^n)^{-1} \cdot [\varepsilon^n \rho^n \nabla \psi^n + \varepsilon^w \rho^w \mathbf{C}^w \cdot \nabla \psi^w], \tag{18}$$

where

$$\mathbf{M}^w \stackrel{\text{def}}{=} \mathbf{B}^w - \mathbf{C}^n \cdot \mathbf{R}^n_{wn} \cdot \mathbf{A}^w,$$

$$\mathbf{M}^n \stackrel{\text{def}}{=} \mathbf{B}^n - \mathbf{C}^w \cdot \mathbf{R}^w_{wn} \cdot \mathbf{A}^n.$$

$\mathbf{M}^\alpha$ is the total mobility for the α-phase and Equation (17) and (18) are generalised Darcy's laws. Note that the superficial velocity of each phase now depends on the potential of the other phase, and that resistances also depends on the amount of interfacial area.

Equation (12) gives the following relationship between ψ^n and ψ^w

$$\begin{aligned} \rho^w \psi^w - \rho^n \psi^n &= p^w - p^n + (\rho^n - \rho^w) g z \\ &= (\gamma^{wn} J^w_{wn}) + (\rho^n - \rho^w) g z \\ &= -P_c + (\rho^n - \rho^w) g z \end{aligned} \tag{19}$$

We then have 6 unknowns

$$\mathbf{u}^w, \mathbf{u}^n, \mathbf{u}, \psi^n, \psi^w \text{ and } P_c.$$

Using Equation (13), (17)-(19) we are able to express $\mathbf{u}^w$ and $\mathbf{u}^n$ as a function of the total Darcy velocity $\mathbf{u}$ and the capillary pressure P_c.

$$\mathbf{u}^w = \mathbf{F}_w \cdot \mathbf{u} + \mathbf{D}_w \cdot \rho^{n,w} \mathbf{g} - \mathbf{D}_w \cdot \nabla P_c, \tag{20}$$

$$\mathbf{u}^n = \mathbf{F}_n \cdot \mathbf{u} + \mathbf{D}_n \cdot \rho^{n,w} \mathbf{g} - \mathbf{D}_n \cdot \nabla P_c, \tag{21}$$

where

$$\rho^{n,w}\mathbf{g} = (\rho^n - \rho^w)\mathbf{g},$$

$$\mathbf{F}_w = (\varepsilon^w + \varepsilon^n \mathbf{C}^n) \cdot \mathbf{M}^n \cdot \mathbf{N}^{-1},$$

$$\mathbf{F}_n = (\varepsilon^w \mathbf{C}^w + \varepsilon^n) \cdot \mathbf{M}^w \cdot \mathbf{N}^{-1},$$

$$\mathbf{D}_w = \varepsilon^n \varepsilon^w (1 - \mathbf{C}^w \cdot \mathbf{C}^n) \cdot \mathbf{N}^{-1} = -\mathbf{D}_n,$$

and

$$\mathbf{N} = \varepsilon^n \mathbf{M}^w + \varepsilon^n \mathbf{C}^n \cdot \mathbf{M}^n + \varepsilon^w \mathbf{C}^w \cdot \mathbf{M}^w + \varepsilon^w \mathbf{M}^n.$$

From Equation (14) we get the following expression for the speed of the interface between the fluid phases

$$\begin{aligned} \mathbf{v}^{wn} &= (\mathbf{A}^w \cdot \mathbf{F}_w + \mathbf{A}^n \cdot \mathbf{F}_n) \cdot \mathbf{u} + \\ &\quad (\mathbf{A}^w \cdot \mathbf{D}_w + \mathbf{A}^n \cdot \mathbf{D}_n) \cdot (\rho^{n,w}\mathbf{g} - \nabla P_c) \\ &= \mathbf{F}_{wn} \cdot \mathbf{u} + \mathbf{D}_{wn} \cdot (\rho^{n,w}\mathbf{g} - \nabla P_c) \end{aligned}$$

where $\mathbf{F}_{wn}$ and $\mathbf{D}_{wn}$ is defined implicitly.

3.1 *Resulting 2 × 2 system*

Substituting the above expression for $\mathbf{u}^w$ and $\mathbf{v}^{wn}$ into Equation (9) and (11) we get the following equations:

$$\begin{aligned} \varepsilon \frac{\partial s^w}{\partial t} + \mathbf{u} \cdot \nabla \cdot \mathbf{F}_w + \mathbf{g} \cdot \nabla \cdot \mathbf{D}_w \\ = \nabla \cdot (\mathbf{D}_w \cdot \nabla P_c) \end{aligned}$$

$$\frac{\partial a^{wn}}{\partial t} + \frac{1}{3}\left(\mathbf{u}\cdot\nabla\cdot(\mathbf{F}_{wn}a^{wn}) + \rho^{n,w}\mathbf{g}\cdot\nabla\cdot(\mathbf{D}_{wn}a^{wn})\right)$$

$$= \frac{1}{3}\nabla\cdot(a^{wn}\mathbf{D}_{wn}\cdot\nabla P_c))$$

This constitutes a 2×2 coupled system for the saturation and the interfacial area.

4 PARAMETERISATION OF RESISTANCES

To close the above system we need to specify the resistances. We assume the functional form of $\mathbf{R}^{\alpha}_{\alpha\beta}$ to be

$$\mathbf{R}^{w}_{wn} = \mu^{w} g^{w}_{wn}(a^{wn}) h^{w}_{wn}(s^{w}) \mathbf{K}^{-1} {\varepsilon^{w}}^{2},$$

$$\mathbf{R}^{w}_{ws} = \mu^{w} g^{w}_{ws}(a^{ws}) h_{w}(s^{w}) \mathbf{K}^{-1} {\varepsilon^{w}}^{2},$$

$$\mathbf{R}^{n}_{wn} = \mu^{n} g^{n}_{wn}(a^{wn}) h^{n}_{wn}(s^{n}) \mathbf{K}^{-1} {\varepsilon^{n}}^{2},$$

$$\mathbf{R}^{n}_{ns} = \mu^{n} g^{n}_{ns}(a^{ns}) h_{n}(s^{n}) \mathbf{K}^{-1} {\varepsilon^{n}}^{2}.$$

The motivation for these forms is as follows:
Equation (2) and (4) are generalised forms of Darcy's law. We therefore expect μ^{α} and $\mathbf{K}$ to be terms in the resistances. In addition, to make the dimensions on the left and right side of the equations to match, ${\varepsilon^{\alpha}}^{2}$ must be a factor too. The g's and h's must be non-dimensional (and positive).

To get expressions for the g's and h's we look at the limit cases where we have only one phase. Equation (2) and (4) should then reduce to the usual Darcy's law.

Case 1: No non-wetting phase

$$s^{w} \to 1, \quad k_{rw} \to 1, \quad a^{wn}/a^{s} \to 0,$$

$$a^{ws}/a^{s} \to 1 \quad \text{and} \quad a^{ns}/a^{s} \to 0.$$

Consider Equation (2). Since the interfacial area a^{wn} is zero, the resistance due to this interface, $\mathbf{R}^{w}_{wn}$, should also be zero and Equation (2) reduces to

$$-\varepsilon^{w}\nabla p^{w} + \varepsilon^{w}\rho^{w}\mathbf{g} = \mathbf{R}^{w}_{ws}\cdot\mathbf{v}^{w}.$$

Hence

$$\mathbf{R}^{w}_{ws} = \mu^{w}\mathbf{K}^{-1}{\varepsilon^{w}}^{2},$$

and we get

$$g^{w}_{wn}(0)h^{w}_{wn}(1) = 0 \text{ and } g^{w}_{ws}(0)h_{w}(1) = 1. \tag{22}$$

Case 2: No wetting phase

$$s^{n} \to 1, \quad k_{rn} \to 1, \quad a^{wn}/a^{s} \to 0,$$

$$a^{ns/a^{s}} \to 1 \quad \text{and} \quad a^{ws}/a^{s} \to 0$$

Consider Equation (4). Since there is no (fluid-fluid) interfacial area, the resistance due to this interface, $\mathbf{R}^{n}_{wn}$, should be zero and Equation (4) reduces to

$$-\varepsilon^{n}\nabla p^{n} + \varepsilon^{n}\rho^{n}\mathbf{g} = \mathbf{R}^{n}_{ns}\cdot\mathbf{v}^{n}.$$

Hence

$$\mathbf{R}^{n}_{ns} = \mu^{n}\mathbf{K}^{-1}{\varepsilon^{n}}^{2},$$

and we get

$$g^{n}_{wn}(0)h^{n}_{wn}(1) = 0 \text{ and } g^{n}_{ws}(0)h_{n}(1) = 1. \tag{23}$$

One set of functions that satisfies the constraints imposed by Equation (22) and (23) are:

$$g^{w}_{wn}(a^{wn}) = \frac{a^{wn}}{a^{s}}, \qquad h^{w}_{wn}(s^{w}) = 1 - s^{w}$$

$$g^{n}_{wn}(a^{wn}) = \frac{a^{wn}}{a^{s}}, \qquad h^{n}_{wn}(s^{n}) = 1 - s^{n}$$

$$g^{w}_{ws}(a^{ws}) = \frac{a^{ws}}{a^{s}}, \qquad h_{w}(s^{w}) = k^{-1}_{rw}(s^{w}) = ({s^{w}}^{2})^{-1}$$

$$g^{n}_{ns}(a^{ns}) = \frac{a^{ns}}{a^{s}}, \qquad h_{n}(s^{n}) = k^{-1}_{rn}(s^{n}) = ({s^{n}}^{2})^{-1}$$

Other constraints arise if we consider the case when the wetting fluid covers the medium ($a^{wn}/a^{s} \to 1$), and the case when the interfacial area is negligible ($a^{wn}/a^{s} \to 0$). One possible set of functions $g^{\alpha}_{\alpha s}$ is:

$$g^{w}_{ws} = 1 + \frac{a^{ns}}{a^{s}}\,\frac{a^{wn}}{a^{s}}$$

$$g^{n}_{ns} = 1 + \frac{a^{ws}}{a^{s}}\,\frac{a^{wn}}{a^{s}}$$

However, this choice appears to be unphysical since the resistance due to the solid-fluid interface depend on the area of the fluid-fluid interface. We will not use this choice in our numerical experiments.

4.1 *Resulting fractional flow and diffusion functions*

In 1D we can write the fractional flow functions $\mathbf{F}_{w}$, $\mathbf{F}_{n}$ and $\mathbf{F}_{wn}$ and the diffusion functions $\mathbf{D}_{w}$, and $\mathbf{D}_{wn}$ as rational polynomials in saturation and interfacial area. The expression are generated using Maple, but omitted because of their length. Using the parameters given in Section 6 we get the surfaces in Figure 1 and 2. Notice that these functions are only weakly dependent on interfacial area.

5 CAPILLARY PRESSURE

In this section a capillary pressure function depending on the (wetting) saturation s and on the interfacial area a^{wn} is constructed. To our knowledge there are almost no available data for $P_c - s^{w} - a^{wn}$ relationships. We therefore use results from network models,

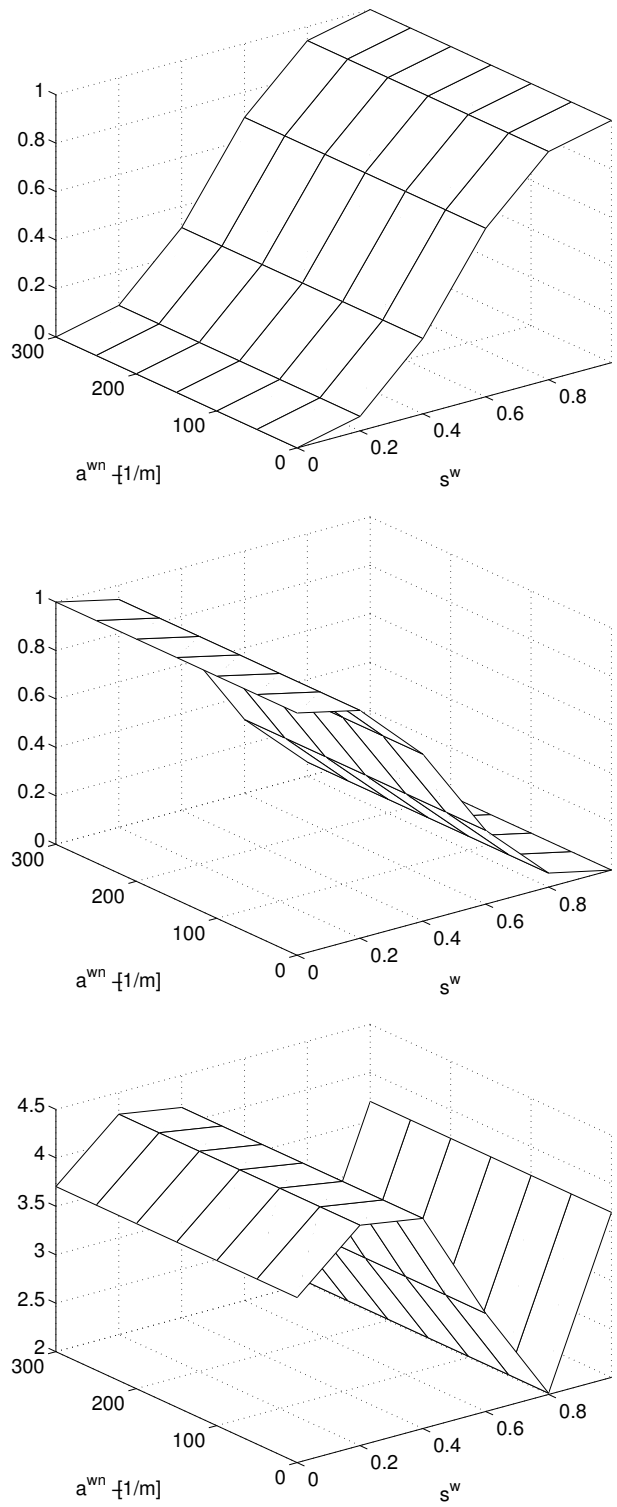


Figure 1: Fractional flow function for the *w*-phase (upper), the *n*-phase (middle) and the *wn*-interface (lower).

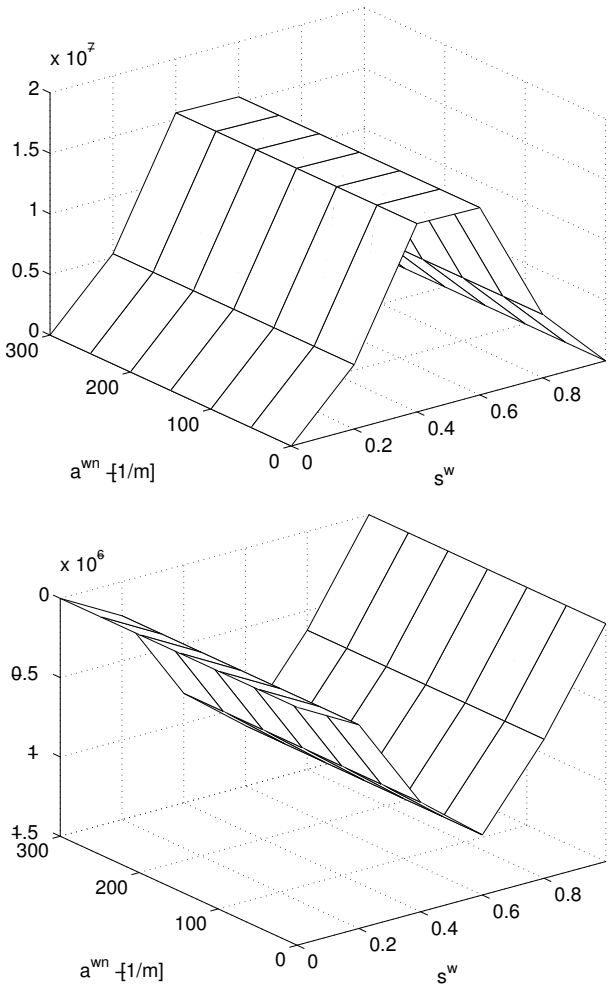


Figure 2: Diffusion function for the *w*-phase (upper) and the *wn*-interface (lower).

see e.g. (Reeves & Celia 1996), and try to capture the qualitative behaviour of surfaces obtained therein.

In (van Genuchten 1980) the following relationship between capillary pressure P_c and saturation s was proposed:

$$P_c = \frac{((s^w)^{-\frac{n}{n-1}} - 1)^{\frac{1}{n}}}{\alpha_G}. \qquad (24)$$

Starting out with this relationship we fix n and vary α_G to get three curves satisfying

$$P_c^1 \leq P_c^2 \leq P_c^3.$$

Here P_c^1 and P_c^3 are approximations to the inner and outer envelopes for the $P_c - s^w$ curves. P_c^2 is chosen midway between the two boundary curves for interpolation purposes. Along these three curves we specify the interfacial area as

$$a_i^{wn}(s^w) = \alpha_i s^w(1 - s^w) + \beta s^w, \quad i = 1, 2, 3, \qquad (25)$$

where α_i is chosen such that $a_1^{wn} \leq a_2^{wn} \leq a_3^{wn}$. In Figure 3 and 4 plots of capillary pressure and interfacial area are shown for a set of parameters.

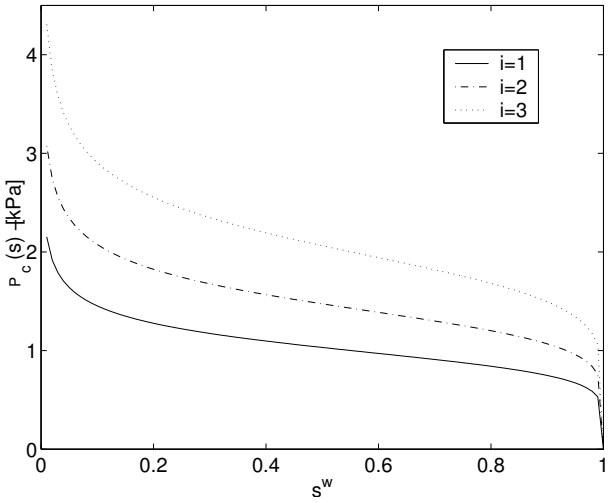


Figure 3: Capillary pressure P_c^i (as a function of saturation) for $n = 7$ and $\alpha = 0.10, 0.07, 0.05$.

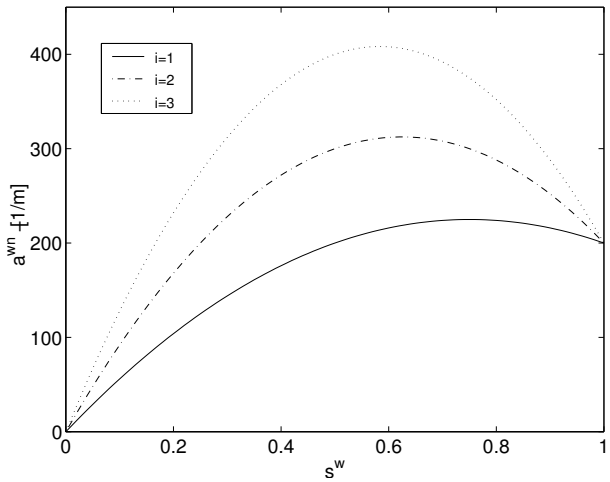


Figure 4: Interfacial area a_i^{wn} on P_c^i, $i = 1, 2, 3$.

Interpolating (quadratically) from these three curves defines the function. For a given P_c we specify $a^{wn}(s, P_c)$ as

$$a^{wn}(s, P_c) = c_1 {s^w}^2 + c_2 s^w + c_3. \tag{26}$$

Let $s_i^w(P_c) = (P_c^i)^{-1}(P_c)$, where $(P_c^i)^{-1}$ is given by Equation (24) with $\alpha_G = \alpha_G^i$. It follows that $a_i^{wn}(s_i^w) = a_i^{wn}(s_i^w(P_c)) = a_i^{wn}(P_c), i = 1, 2, 3$. Using Lagragian interpolating we get the following expression for the interfacial area as a function of capillary pressure and saturation:

$$\begin{aligned} a^{wn}(P_c, s^w) = \\ a_1^{wn}(P_c)\frac{(s^w - s_2^w(P_c))(s^w - s_3^w(P_c))}{(s_1^w(P_c) - s_2^w(P_c))(s_1^w(P_c) - s_3^w(P_c))} \\ + a_2^{wn}(P_c)\frac{(s^w - s_1^w(P_c))(s^w - s_3^w(P_c))}{(s_2^w(P_c) - s_1^w(P_c))(s_2^w(P_c) - s_3^w(P_c))} \\ + a_3^{wn}(P_c)\frac{(s^w - s_1^w(P_c))(s^w - s_2^w(P_c))}{(s_3^w(P_c) - s_1^w(P_c))(s_3^w(P_c) - s_2^w(P_c))} \end{aligned}$$

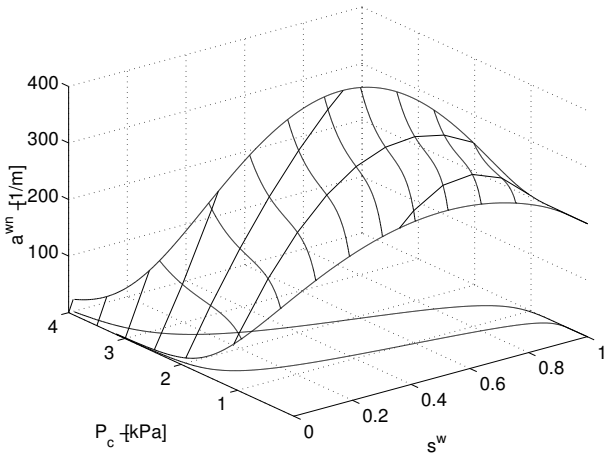


Figure 5: Interfacial area as a function of wetting saturation and capillary pressure given by the three curves above.

Since the interfacial area is non-monotone in capillary pressure, see Figure 5, it is generally impossible to find a unique expression for P_c as a function of saturation and interfacial area. In the equations we will need the derivative of capillary pressure with respect to saturation and interfacial area. If we assume uniqueness of P_c locally around a point (s^w, a^{wn}) we can employ implicit differentiation of function to evaluate

$$\left(\frac{\partial P_c}{\partial s^w}\right)_{a^{wn}} \quad \text{and} \quad \left(\frac{\partial P_c}{\partial a^{wn}}\right)_{s^w}.$$

6 EXAMPLES

We have done a simple test case in 1D. Data for parameters are taken from (Dahle & Celia 1999) and from a table over typical values of permeability and porosity for sand stones. The specific area of the solid phase surface is calculated using the Carman-Kozeny equation. The following values are used:

$\mu^w = 10^{-3}\ [Ns/m^2]$
$\mu^n = 10^{-2}\ [Ns/m^2]$
$\epsilon = 0.27$
$K = 1.3\times10^{-9}\ [m^2]$
$u = 1.3\times10^{-3}\ [m/s]$
$a^s = 7.6\times10^{4}\ [1/m]$
$a^{ns} = 0.7\times10^{4}\ [1/m]$
$a^{ws} = 6.9\times10^{4}\ [1/m]$
$n = 7$
$\alpha_G^i = 10^{-3}, 0.7\times10^{-3}, 0.5\times10^{-3}$
$\alpha_i = 400, 800, 1200$
$\beta = 0$

Since we are working in 1D, gravity terms are neglected. To avoid difficulties with boundaries and discontinuities, a smooth ramp is initially specified for the saturation. An increased amount of interfacial area is evenly distributed around this ramp.The equations are solved by sequential time stepping using a straight forward upstream/central difference scheme.

The initial data and solutions for interfacial area and saturation are shown in Figure 6. This problem appears to be advection dominated. It follows that since the advective flux is nearly insensitive to interfacial area, the dynamics of the saturation is nearly independent of the changes in interfacial area.

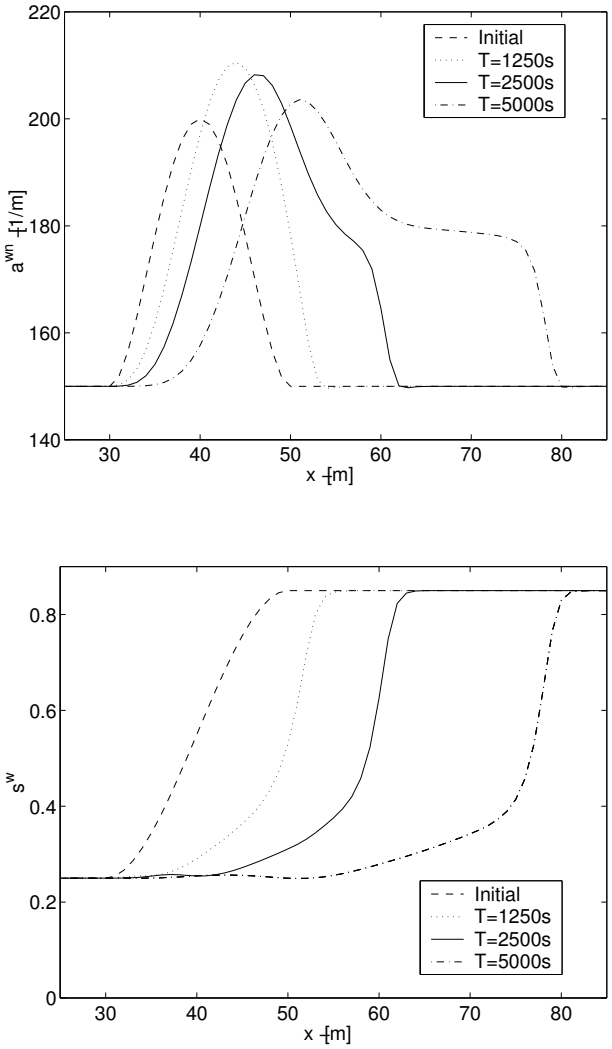


Figure 6: Profiles for interfacial area (upper) and wetting saturation (lower) at $t = 0, 1300, 2600$ and 5200s .

7 CONCLUSIONS

In this work, we have presented initial numerical solutions for an enhanced model of two phase flow which includes fluid-fluid interfacial area as a primary variable. The solutions are for a simplified test problem under a specific set of assumptions, and with a particular choice of functional forms for the nonlinear coefficients. For this particular case, the saturation profiles show little dependence on interfacial area. This result is consistent with the forms of the fractional flow functions, which show minimal sensitivity to interfacial area. This in turn is a reflection of the functional forms chosen to parameterise the various nonlinear resistance terms that arise in the governing equations. Because the correct functional forms for these resistance terms are not known, more general sensitivity studies are required to assess the practical importance of the interfacial area equations to flow modeling. In addition, numerical solutions need to be obtained for cases where the simplifying assumptions applied to the governing equations are systematically examined, so that the sensitivity of the solutions to these assumptions can be assessed. The results presented herein represent a starting point for this more general analysis.

While we cannot reach any general conclusions based on our initial calculations, we can make a few general observations about the overall system of equations. First, inclusion of the interfacial area equations allows for direct calculation of the amount of interfacial area in the system, as a function of space and time. Equations written for the individual phases also incorporate effects of the interfaces through functional dependences in the appropriate nonlinear coefficients. In addition, inclusion of a unique functional relationship between P_c, s^w, and a^{wn}, and the associated transport equation for interfacial area, allow hysteresis within the $P_c - s^w$ plane to be incorporated into the algorithm with no additional effort. This means that arbitrary drainage and imbibition cycles can be simulated without regard for redefinition of the constitutive curves. This appears to be a major advantage of this approach to multi phase flow modeling. The cost of this additional generality and flexibility is the requirement to identify and quantify additional parameters, as well as their functional dependences on interfacial area. Additional equations also need to be solved, as compared to the traditional equations for multi phase flow; and these additional equations remain to be analysed mathematically, as do the associated numerical algorithms implemented for their solution.

The overall significance of interfacial areas in the mathematical description of multi phase flow in porous media remains to be determined. To make this determination, numerical simulators are required to allow a wide range of possible functional forms for the nonlinear coefficients to be explored. The degree to which the governing equations can be simplified must also be explored numerically. This requires a systematic and comprehensive numerical approach. The results presented herein represent a first contribution to the overall numerical study.

ACKNOWLEDGEMENTS

This work was supported in part by the U.S. Department of Energy under grants DE-FG07-96ER14701 and DE-FG07-96ER14703, and in part by the Norwegian Research Council (NFR) under grants 116153/431 and 127460/410.

REFERENCES

Dahle, H. & Celia, M. (1999). A dynamic network model for two-phase immiscible flow. *Computational Geosciences 3*, 1–22.

Gray, W. (1999). Thermodynamics and constitutive theory for multiphase porous-media flow considering internal geometric constraints. *Advances in Water Resources 22*(5), 521–547.

Gray, W. (2000). Macroscale equilibrium conditions for two-phase flow in porous media. *International Journal of Multiphase Flow 26*(3), 467–501.

Gray, W. & Hassanizadeh, S. (1998). Macroscale continuum mechanics for multiphase porous-media flow including phases, interfaces, common lines and common points. *Advances in Water Resources 21*(4), 261–281.

Reeves, P. & Celia, M. (1996). A functional relationship between capillary pressure, saturation, and interfacial area as revealed by a pore-scale network model. *Water Resources Research 32*(8), 2345–2358.

van Genuchten, M. (1980). A closed form equation for predicting the hydraulic conductivity of unsatured soils. *Soil Science Society of America Journal 44*, 892–899.

Paper E

A pore network model for calculation of interfacial velocities*

* Submitted to Advances in Water Resources, fall 2001.

A PORE NETWORK MODEL FOR CALCULATION OF INTERFACIAL VELOCITIES

H.F. Nordhaug [a], M. Celia [b], H.K. Dahle [a]

[a] *Department of Mathematics, University of Bergen, Johs. Brunsgt. 12, N-5008 Bergen, Norway*

[b] *Department of Civil and Environmental Engineering, Princeton University, Princeton, NJ 08544, USA*

Abstract

Two-phase flow in porous media is characterized by fluid-fluid interfaces that separate the fluid phases at the pore scale. These interfaces support pressure differences between phases, and their dynamics allow for saturation changes within the porous medium. Dynamic pore-scale network models allow interface dynamics to be modeled explicitly, such that each fluid-fluid interface within a network of pores is tracked explicitly in space and time. Because these models produce a detailed description of both phase and interface dynamics, results from these models can be averaged to provide values for many upscaled variables. These include nontraditional variables such as amounts of interfacial area, and volume-averaged interfacial velocities. While these upscaled variables provide insights into the underlying dynamics of two-phase flow systems, they also allow new theories involving interfacial area to be tested directly. Results from a test of one such theory reveals that proposed constitutive equations involving interfacial velocities fail to match results from the network model for most cases considered. Therefore the macroscopic equations require additional development before they can be used for macro-scale simulations.

Key words: Two-phase flow, dynamic pore-scale network models, interfacial area, upscaling

1 Introduction

Multi-phase porous media systems are characterised by fluid-fluid interfaces that exist at the pore scale. These interfaces define the spatial boundaries of each phase

Email addresses: hansfn@mi.uib.no (H.F. Nordhaug), celia@princeton.edu (M. Celia), reshd@mi.uib.no (H.K. Dahle).

at any given instant in time. Interfaces also have properties such as interfacial tension, which allows each fluid to maintain a different pressure. The resulting difference between individual phase pressures is usually called capillary pressure. Fluid-fluid interfaces also provide the surfaces across which mass is transferred from one fluid phase to another, in problems such as dissolution of non aqueous phase liquid (NAPL) into flowing ground waters. Fluid-solid interfaces provide similar surfaces for mass exchanges between fluid and solid phases. For these reasons, an understanding of interfacial behaviour at the pore scale, and subsequent scaling of that behaviour to the more practical continuum scale, is important to proper descriptions of porous media flow systems.

The view that interfacial area plays an important role in porous media flows is supported by the recent theoretical developments reported by Gray and Hassanizadeh [1]. In their remarkable work, a thermodynamic approach is used to show that the relationship between capillary pressure and saturation is incomplete, and that specific interfacial area (defined as amount of interfacial area per unit volume of porous medium) should enter into the relationship. Therefore one should seek a relationship between capillary pressure, saturation, and interfacial area(s). Such a relationship has been derived, using a pore-scale network model, by Reeves and Celia [2]. Other simulation studies have explored a variety of additional aspects of the new theories of Gray and Hassanizadeh, see for example Held and Celia [3]. One of the intriguing aspects of the theoretical developments of Gray and Hassanizadeh is the inclusion of equations based on interface dynamics. That is, in order to properly account for interfacial effects in the governing equations, new equations must be written to take account of interfacial dynamics. These are based on conservation laws written for the interfaces, with associated constitutive relations.

One of the difficulties associated with tests of theories involving interfacial areas is the lack of experimental procedures to measure these areas. While partitioning tracers offer hope for inferring amounts of area (for example Heonki et al. [4]), and novel laboratory procedures and equipment offer hope for more direct measurement of interfacial areas (Montemagno and Gray [5]), experimental determination of interfacial area is still a daunting task. And experimental determination of interfacial dynamics, which tends to occur on very short time scales, is even more difficult. Therefore, testing of these theories must rely on use of specific kinds of computational models, including so-called pore-scale network models. Pore-scale network models typically represent the pore space of the medium using simplified geometries, and within this geometric representation solve equations to track explicitly the location of all fluid-fluid interfaces within the network. Typical geometric representations include regular lattice structures, such as a cubic lattice, with pore bodies corresponding to the vertices of the lattice, and pore throats connecting the pore bodies. Pore bodies are often assigned spherical shapes, although cubes or other shapes with corners are sometimes used to account for wedge and corner flows. Similarly, pore throats may be cylindrical with circular cross section, or cylindrical with rectangular or perhaps triangular cross sections. For all of these choices, the

geometry is kept sufficiently simple that interface configurations can be calculated analytically, for a given set of fluid pressures. These models are often run to mimic typical laboratory experiments, such as pressure cell tests to determine the relationship between capillary pressure and relative fluid saturation. This is accomplished by use of lattices that are sufficiently large to define meaningful continuum-scale measures, such as fluid saturation. Results of such simulations show all of the major features of experimental relationships, including finite entry pressures, residual saturations, and hysteresis. Because the simulators define interface geometries explicitly, additional geometric information may be extracted from network models. For example, fluid-fluid interfacial area can be calculated easily, given knowledge about the location and shape of each fluid-fluid interface.

Two general types of pore-scale network models may be identified: quasi-static models and dynamic models. In a quasi-static models, the location of any fluid-fluid interface is governed by equilibrium considerations only. Equilibrium states are determined from the Young-Laplace equation, which relates the capillary pressure to the interfacial tension and the interface curvature, viz. (Dullien [6])

$$P_c = \frac{\sigma}{R} \tag{1}$$

In this equation, σ is interfacial tension and R is the radius of curvature. Depending on the shapes chosen for the pore elements, the Young-Laplace equation is modified to include contact angles and pore geometry to develop rules about whether or not an interface is stable at a given location, given an imposed capillary pressure. If the interface is unstable, it is moved through the network until a stable position is found, or until it exits the network. No time dependence is included in the calculation; the interface is simply moved from one equilibrium position to another. This type of model is consistent with an algebraic relationship between capillary pressure and saturation, where changes in capillary pressure are translated instantaneously to changes in saturation. Examples of these kinds of models include those described by Dullien [6], Ferrand and Celia [7], and Reeves [8], among many others.

A second type of pore network model involves computation of transient behaviour associated with interface movement. That is, unstable interfaces are tracked through the network until a stable position is reached, but the transient nature of the movement from one position to another is explicitly described and modeled. While these transient models are more computationally complex, they allow the underlying transients associated with interface movement to be incorporated and analysed explicitly. While most pore-scale network models reported in the literature are quasi-static, there have been several dynamic models that have been developed. These include the model of Blunt and King [9]; a series of models by Payatakes and coworkers (see, for example, Valavanides and Payatakes [10]), and more recent models by Mogensen and Stenby [11], Aker et al. [12], Hassanizadeh (Dijkstra et al. [13]) and by Dahle and Celia [14]. The models of Payatakes are the most

comprehensive, including a focus on mobilization of trapped fluids and so-called drop traffic flows.

In this paper, we use a model based on the original work of Blunt and King [9] and calculate dynamic interfacial behaviour, with focus on volume-averaged interfacial velocity. After describing the salient features of the model, and providing a definition of average interfacial velocity, specific calculations are performed to demonstrate how continuum-scale measures of both interface velocity and phase velocity can be quantified. The model is then used to test a specific theoretical conjecture regarding interfacial velocity and its relationship to average phase velocities. We show fundamental differences between stable and unstable displacements, and discuss how these differences impact the comparisons between theory and model results.

2 The dynamic network model

The pore-scale network model used herein is an extension of the model of Blunt and King [9]. The pore network is a rectangular lattice, with spherical pore bodies and cylindrical pore throats, with pore-size distributions defined for the bodies and throats, see Figure 1(a). Following Blunt and King [9], the model is simplified by the following assumptions: (1) local capillary pressure in the pore throats is assumed to be negligible, so that only one pressure exists within a pore body, independent of the local saturation of that pore body; (2) while the radius of a pore throat, r_{ij}, serves to define its hydraulic conductance, the volume contributed by the pore throat is assumed to be small relative to volumes of pore bodies, therefore movement of an interface through a pore throat is assumed to occur instantaneously; (3) flow within pore throats is assumed to be laminar and given by Poiseuille's law; (4) both fluids are assumed to be incompressible. With these assumptions, the set of governing equations is relatively simple. Each equation must obey volume conservation within each pore body, such that

$$V_i \frac{dS_i^\alpha}{dt} + \sum_{j \in N_i} Q_{ij}^\alpha = 0 \tag{2}$$

where V_i represents the volume of pore body i, S^α represents local saturation (percent of V_i filled with fluid α), Q_{ij}^α is the volumetric flux from pore body i to it's neighbour j, and N_i is a list of all neighbour pore bodies for pore body i. This equation is written for both fluid phases, wetting ($\alpha = w$) and non-wetting ($\alpha = n$). The volumetric flux is related to pressures at the pore bodies by Poiseuille's law,

$$Q_{ij}^\alpha = G_{ij}^\alpha (p_i^\alpha - p_j^\alpha) \tag{3}$$

where p_l^α represents pressures, and G_{ij}^α represents hydraulic conductance in the pore throat connecting pore bodies i and j. Because the pore throats are cylindrical, and interface movement through them is instantaneous, only one fluid can occupy a given pore throat, at a given time. Therefore the fluid occupying the pore throat has conductance

$$G_{ij}^\alpha = \frac{\pi r_{ij}^4}{8\mu^\alpha l_{ij}}, \tag{4}$$

while the non-occupying fluid has zero conductance. Summation of Equation (2) over the two phases gives the equation

$$\sum_{j \in N_i} (Q_{ij}^w + Q_{ij}^n) = 0 \tag{5}$$

Substitution of Equation (3) for each of the phase fluxes Q_{ij}^α provides a set of algebraic equations with the pore-body pressures as unknowns. These can be solved using standard matrix solution methods.

In the simulations reported herein, we only consider drainage, so that the resident fluid is the wetting fluid, and the invading fluid is non-wetting fluid. Time steps are chosen so that during any time step, only one pore body reaches full non-wetting phase saturation. That pore body then generates additional interfaces, located at all connecting pore throats that are filled with wetting fluid. Those interfaces are then tested for capillary stability, using the most recent pore body pressures. If they are stable, they remain in place, and the connecting pore throat is marked as trapped, with conductances set to zero for both phases. If the interface is unstable, it passes through the pore throat, non-wetting fluid occupies the pore throat, and non-wetting fluid can then begin to invade the adjoining pore body.

Overall, the algorithm proceeds as an Implicit Pressure Explicit Saturation (IMPES) routine. The major unknowns are the pressure and saturation of each pore body. For a given distribution of fluids, phase conductances are calculated and put into Equation (5), which is solved for a new pressure field. That pressure field is then used in Equation (3) to compute fluxes through the pore throats. These fluxes are then used, in conjunction with knowledge of the current saturations in each pore body, to determine the minimum filling time for each of the pores, and this is set as the time step size. Then Equation (2) is used to update the saturations in each pore body. Newly created interfaces are tested for stability, conductances are updated, and the procedure is repeated. In the matrix solution for pressures, regions of wetting fluid that become completely surrounded by non-wetting fluid, and are therefore hydraulically trapped, are removed from the matrix equations to avoid singular matrices. This algorithm provides a transient response for both pressure and saturation.

This overall leads naturally to investigations of relationships between these variables, as well as others, such as interfacial area. In order to calculate these variables,

averages need to be defined over a representative volume. The volume might be chosen to correspond to the entire volume of the network, or it might be defined as essentially two-dimensional slices through the network, in the direction perpendicular to the macroscopic direction of displacement. In either case, volume-averaged saturation is defined as

$$S^{\alpha} = \frac{\sum_{i \in N_{vol}} V_i S_i^{\alpha}}{\sum_{i \in N_{vol}} V_i} = \frac{1}{V} \sum_{i \in N_{vol}} V_i S_i^{\alpha} \tag{6}$$

where V is the volume of the chosen averaging region, and N_{vol} denotes the set of pore bodies within the chosen averaging volume. Capillary pressure is defined as the difference between volume-averaged phase pressures, such that

$$P_c = p_n - p_w = \frac{1}{V}\Big(\sum_{i \in N_{vol}} V_i S_i^n p_i - \sum_{i \in N_{vol}} V_i S_i^w p_i \Big) \tag{7}$$

where p_i corresponds to the pressure in pore body i. Notice that while no local capillary pressure exists (by assumption in the model), a macroscopic capillary pressure is still well defined based on the average phase pressures.

Average properties associated with interfaces require somewhat more care in their definitions. Because no capillary pressure is associated with individual pore bodies, the shape (especially the curvature) of a particular interface is not specified. The only information for active (non-trapped) interfaces is that they reside in a specific pore body. To assign a measure of interfacial area, an interface within a filling pore body is assigned an area equal to the diameter of the pore body. Those interfaces trapped at the entrance to pore throats are assigned the area associated with the diameter of the pore throat. Therefore, the specific interfacial area, defined as the amount of interfacial area per unit volume of porous medium, is defined as

$$a^{wn} = \frac{1}{V}\Big(\sum_{i \in N_{vol}} a_i^{wn} + \sum_{j \in M_{vol}} a_j^{wn} \Big) = \frac{A^{wn}}{V} \tag{8}$$

where M_{vol} denotes the set of pore throats that are contained within the averaging volume V.

The averaged variable that is the most difficult to define is the volume-averaged interfacial velocity. Because interfaces that reside within pore bodies are not resolved, their velocities need to be inferred from fluxes into and out of the pore body, changes in saturation within the pore body, and lengths of travel associated with the pore body and its connected pore throats. Consider a fluid-fluid interface that is spawned by complete filling of a given pore body. It first comes into contact with the connected pore throat, and a capillary stability test is performed. If the interface experiences sufficient capillary pressure to drive it through the pore throat, it will move through the throat with infinite velocity, then will fill the connected

pore body at some finite rate, given by the subsequent computed changes in saturation for that pore body. The infinite speed is a consequence of the simplifying assumption that the pore throats are volumeless. To define a finite speed that preserves proper global velocities, the pore filling associated with saturation changes is extended in length to cover the combined pore body – pore throat combination, and the 'locally averaged' velocity for a specific interface is defined as

$$\|\mathbf{v}_i^{wn}\| = l_{i,j}\frac{\Delta S_i^n}{\Delta t} \tag{9}$$

where the double brackets signify magnitude of the interface velocity vector, the length $l_{i,j}$ denotes the length of the pore throat through which the entering interface travels plus the diameter of the pore body i, and ΔS_i^n denotes the change in saturation over the time interval Δt. The direction assigned to the interfacial velocity vector is the average of the total inflow vector and total outflow vector. Finally, the volume-averaged velocity vector is given by the sum of each interfacial velocity weighted by the area of the interface,

$$\mathbf{v}^{wn} = \frac{1}{A^{wn}}\sum_{i\in N_{vol}} \mathbf{v}_i^{wn} a_i^{wn} \tag{10}$$

Volume-averaged phase velocities may be defined analogously to average interfacial velocities with volume replacing areas as the appropriate weights, such that

$$\mathbf{v}^{\alpha} = \frac{1}{V S^{\alpha}}\sum_{i\in N_{vol}} \mathbf{v}_i^{\alpha} V_i S_i^{\alpha} \tag{11}$$

where $\mathbf{v}_i^{\alpha}$ is given by an equation similar to Equation (9), with appropriate modification for the case of $S = 1$ or $S = 0$.

3 Theoretical Conjectures

One of the more intriguing developments in the past decade has been proposal of a new set of governing equations for two-phase flow in porous media. These equations include volume-averaged, specific interfacial area as a key variable. Among other things, capillary pressure and saturation must have an extended relationship that includes interfacial area; conservation equations are written for interfaces, and these equations include interfacial area as a dependent variable; and new constitutive relationships proposed to close this expanded equation set include interfacial area as a key component. Examples of these equations include the following:

$$\frac{da^{wn}}{dt} + \nabla\cdot(\mathbf{G}^{wn} a^{wn} \mathbf{v}^{wn}) = F^{wn} \tag{12}$$

where $\mathbf{G}^{wn}$ is a geometric tensor and F^{wn} refers to a general production term for interfacial area (see [15]). Also, as a proposed constitutive relationship for interfacial velocities, Nordhaug et al. [15] have proposed the following relationship between volume-averaged interface velocity and volume-averaged phase velocities:

$$(\mathbf{R}^{w}_{wn} + \mathbf{R}^{n}_{wn}) \cdot \mathbf{v}^{wn} = \mathbf{R}^{w}_{wn} \cdot \mathbf{v}^{w} + \mathbf{R}^{n}_{wn}\mathbf{v}^{n} \tag{13}$$

In this equation, the coefficients $\mathbf{R}^{\alpha}_{wn}$ represent resistances to flow, and could be functions of phase saturation(s) or other variables. The general idea of the equation is that the interfacial velocity should be a weighted sum of the phase velocities. While the resistance coefficients may be unknown, the general concept can be tested via the network model. Similarly, the terms in the earlier equation (Equation (12)) can be determined from the network model, and therefore the equation may be tested for validity. Because these variables cannot be measured experimentally at this time, the only tool available to test these proposed new equations is the network simulator. This is how we use the network model in the next section, as continuum-scale variables are computed and the equations proposed above are evaluated and tested.

3.1 Modeling of resistance terms $\mathbf{R}^{\alpha}_{wn}$

In [15] a complete model for two phase flow including interfacial area was presented. We provide a few of the key equations to illustrate the ideas. The governing equations includes Equation (12) and (13) which are geometric constraint and momentum balance equations, respectively, written for a massless interface. In addition, phase equations for mass and momentum written as:

$$\frac{\partial(\varepsilon^{\alpha}\rho^{\alpha})}{\partial t} + \varepsilon^{\alpha}\rho^{\alpha}\nabla \cdot \mathbf{v}^{\alpha} = 0. \tag{14}$$

and

$$-\varepsilon^{\alpha}\nabla p^{\alpha} + \varepsilon^{\alpha}\rho^{\alpha}\mathbf{v} = (\mathbf{R}^{\alpha}_{wn} + \mathbf{R}^{\alpha}_{\alpha s}) \cdot \mathbf{v}^{\alpha} - \mathbf{R}^{\alpha}_{wn} \cdot \mathbf{v}^{wn}. \tag{15}$$

The functionality of the resistance terms $\mathbf{R}^{\alpha}_{wn}$ and $\mathbf{R}^{\alpha}_{\alpha s}$ in Equation (13) and (15) were assumed to be as follows:

$$\begin{aligned}
\mathbf{R}^{w}_{wn} &= \mu^{w} g^{w}_{wn}(a^{wn}) h^{w}_{wn}(s^{w}) \mathbf{K}^{-1} {\varepsilon^{w}}^{2}, \\
\mathbf{R}^{w}_{ws} &= \mu^{w} g^{w}_{ws}(a^{ws}) h_{w}(s^{w}) \mathbf{K}^{-1} {\varepsilon^{w}}^{2}, \\
\mathbf{R}^{n}_{wn} &= \mu^{n} g^{n}_{wn}(a^{wn}) h^{n}_{wn}(s^{n}) \mathbf{K}^{-1} {\varepsilon^{n}}^{2}, \\
\mathbf{R}^{n}_{ns} &= \mu^{n} g^{n}_{ns}(a^{ns}) h_{n}(s^{n}) \mathbf{K}^{-1} {\varepsilon^{n}}^{2}.
\end{aligned}$$

These functional forms were chosen according to the following observations in [15]. Equation (15) is a generalised form of Darcy's law. When the saturation goes

to zero or one for one of the phases Equation (15) should reduce to the normal Darcy's law for two phase flow. The viscosity and permeability terms, μ^α and $\mathbf{K}$, are included to make the dimensions match on the left and right hand sides. The g's and h's must be non-dimensional (and positive). The volume fraction terms, ${\epsilon^\alpha}^2$, are added to be able to go from average phase velocities to Darcy velocities.

By considering the results from the network model we should be able to investigate whether Equation (13) is reasonable. However, to be able to say something quantitative about the resistance terms $\mathbf{R}^\alpha_{wn}$ we must use the definitions above. When $\mathbf{K}$ is a scalar we get the following expression for the interface velocity [15]:

$$\mathbf{v}^{wn} = F^w\mathbf{v}^w + F^n\mathbf{v}^n = \frac{f^w}{f^w + f^n}\mathbf{v}^w + \frac{f^n}{f^w + f^n}\mathbf{v}^n \tag{16}$$

where we have chosen

$$f^\alpha = (1 - s^\alpha)^3(s^\alpha)^2. \tag{17}$$

4 Equation testing and numerical results

In the following we use a three-dimensional pore network to compute averaged interfacial and phase velocities. We use these computed velocities to test the theoretical conjecture given in Equation (13). We also compute other averaged variables, such as specific interfacial area, to demonstrate certain behavior patterns in the two-phase flow system. For the averaged quantities, we sometimes use a single averaging volume, spanning the entire network, and we sometimes use a 'sliding average' that involves subsets of the total network volume. The latter averaging allows us to compute spatially varying averaged quantities.

4.1 Experimental setup

All simulations are performed on a three-dimensional network of size $10 \times 10 \times 50$ pore bodies. When sliding averages are calculated, the size of the sub-volumes is $10 \times 10 \times 10$, and the averages as calculated along the largest direction (that is, the direction with 50 pore bodies, assumed to be the vertical direction).

Boundary and initial conditions are set up to correspond to primary drainage. The top boundary is set to have a pressure of 4000 Pa, and fixed saturation of $S^w = 0$, while the bottom boundary is set with pressure of 0 Pa and saturation $S^w = 1$. No-flow conditions are imposed along the lateral sides of the domain. The initial condition is taken as $S^w = 1$, with a uniform pressure of 2000 Pa.

Fluid viscosities are chosen so that both stable and unstable displacements are simulated. Table 1 lists the viscosities used for the wetting and nonwetting fluids. While we have performed simulations over a range of viscosity values, we will use three representative values of viscosities so that the viscosity ratio between nonwetting and wetting fluids is 10, 1, and 0.1. In all simulations, the fluids are assigned an interfacial tension of 0.72 [N/m], and the contact angle is set at zero. We ignore density differences between the two fluids.

Table 1
Wetting and non-wetting fluid viscosities

Set	μ_w	μ_{nw}	
1	0.001	0.010	[Ns/m^2]
2	0.001	0.001	[Ns/m^2]
3	0.001	0.0001	[Ns/m^2]

The radii of the pore bodies and pore throats are generated using a cut-off log-normal distribution. The same standard deviation (or the parent distribution) is used for both the bodies and the throats, but different means and upper and lower cut-off values are used. Appropriate information is provided in Table 2.

Table 2
Distribution parameters for pore and throat radius.

	Set 1	
Standard deviation	0.25	
Pore body mean	$1.2 \cdot 10^{-3}$	[m]
Pore body lower cut-off	$0.6 \cdot 10^{-3}$	[m]
Pore body upper cut-off	$1.8 \cdot 10^{-3}$	[m]
Pore throat mean	$4.0 \cdot 10^{-4}$	[m]
Pore throat lower cut-off	$2.0 \cdot 10^{-4}$	[m]
Pore throat upper cut-off	$6.0 \cdot 10^{-4}$	[m]

4.2 Calculation of Local and Averaged Quantities

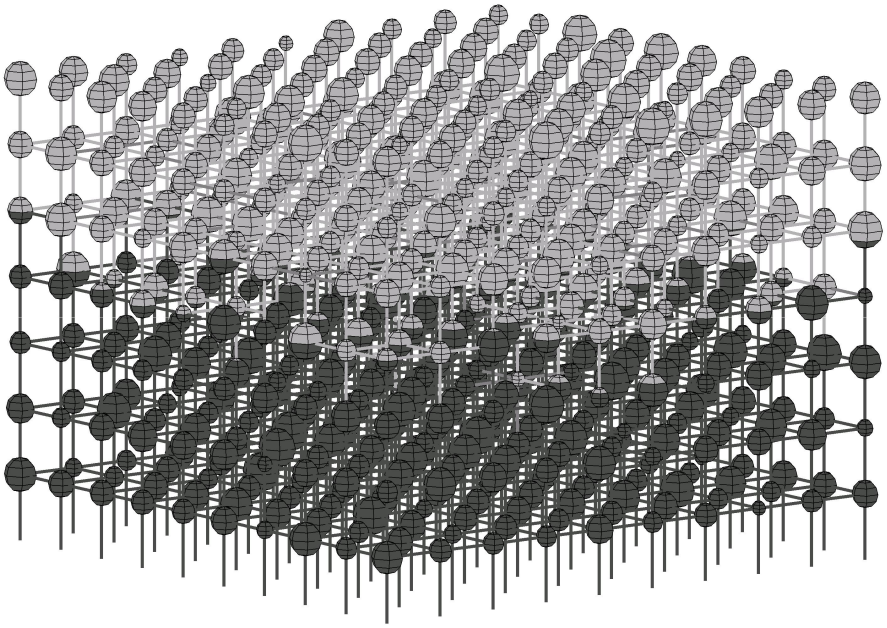
Numerical solution of the network equations provides values for fluid pressures and associated saturations in each pore body, for each time step. From these values, and the geometry of both the network and the chosen averaging volumes, we compute averaged quantities as post-processing calculations. Because we have information about location and state of all fluid-fluid interfaces, we can distinguish between trapped and moving interfaces. Therefore, when defining averaged quantities, we

may choose to distinguish between averages that include all interfaces, and averages that only involve active interfaces. As the default, we include all interfaces in the calculations, consistent with definitions such as that given in Equation (8). However, there are times when it is useful to isolate only those interfaces that are moving. When averages are taken over active interfaces only, then we focus the calculations on pore bodies only, and we neglect pore throats, including pore throat volumes. The reason is that interfaces that are trapped will reside at the entrance to pore throats, so for consistency between interfacial discrimination and volume calculations, we decided to eliminate pore throat volumes and areas when we ignore trapped interfaces.

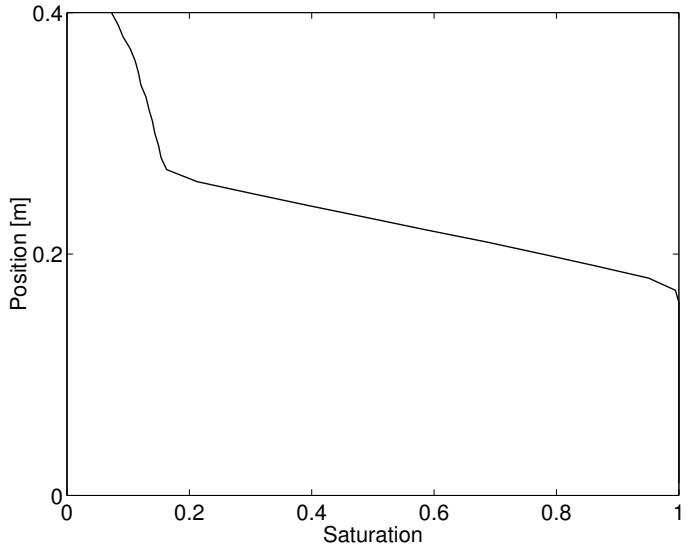
4.3 Stable displacement

Several simulations were run with viscosity rations greater than one. To illustrate the case of stable displacement, a viscosity ratio of 10 is used, with the boundary and initial conditions presented above. Network occupancy of fluids is shown in Figure 1(a), for an intermediate time corresponding to movement through the upper portion of the network. A flat, piston-like front can be seen moving through the network. Very little wetting fluid is left behind, with virtually all trapped wetting fluid existing as trapped singlets in pore throats. In Figure 1(b), average saturation, based on the sliding average concept, is plotted as a function of depth. The macroscopic frontal behavior of the displacement is clear in this figure.

Average phase and interfaces velocities may be calculated for this case. In Figure 2, the averaged interfacial velocity is compared to that predicted by Equation (13). The results in Figure 2(a) show that the average interfacial velocity is consistently below the predicted values based on the average phase velocities (Equation (13)). A relatively poor prediction is also seen in the sliding averages shown in Figure 2(b). However, if we exclude the trapped interfaces, and only use active interfaces in the calculations, we find that the theory matches the numerical values quite well, see Figure 3. This is because elimination of trapped interfaces leaves only pores that are actively filling, and the macroscopically flat front means that interfaces are constrained to move with the same velocity of the invading and defending fluids, just as would happen in a single tube or pipe. So viscous stability, which produces flat front macroscopically, lead to active interface velocities that are essentially identical to the two phase velocities, which themselves are equal. In this case, the theoretical equation appears to hold. However, inclusion of all interfaces leads to disparity between the values, due to the presence of trapped interfaces.

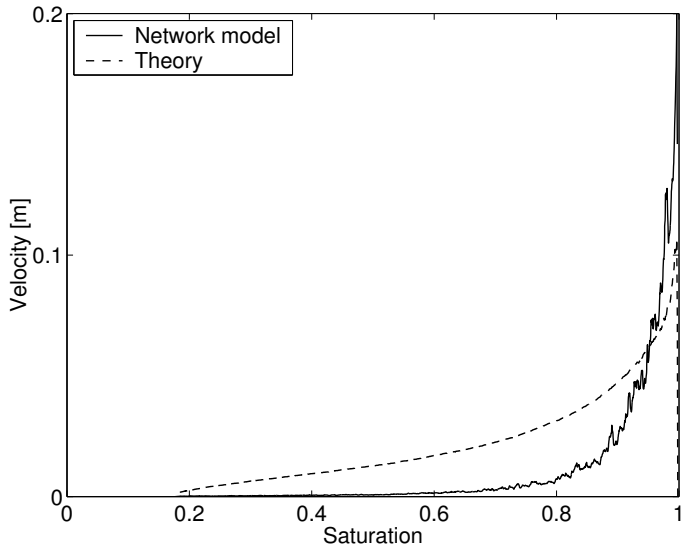


(a) microscale

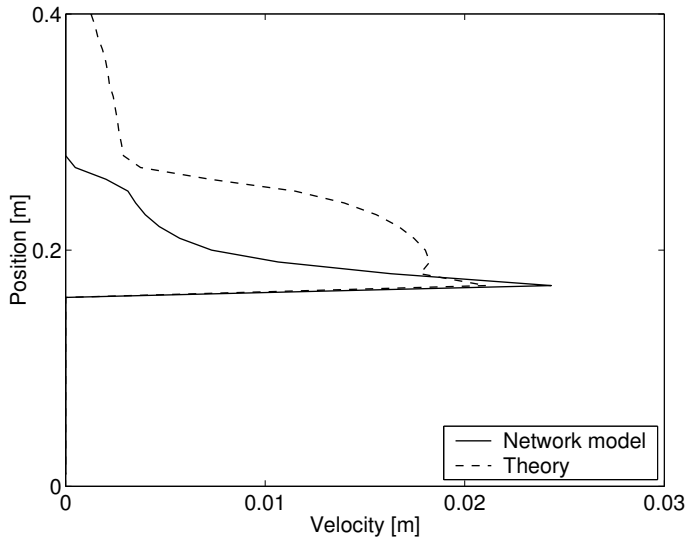


(b) macroscale

Fig. 1. Stable displacement – $M_\mu = 10.0$. Snapshot of the saturation front.

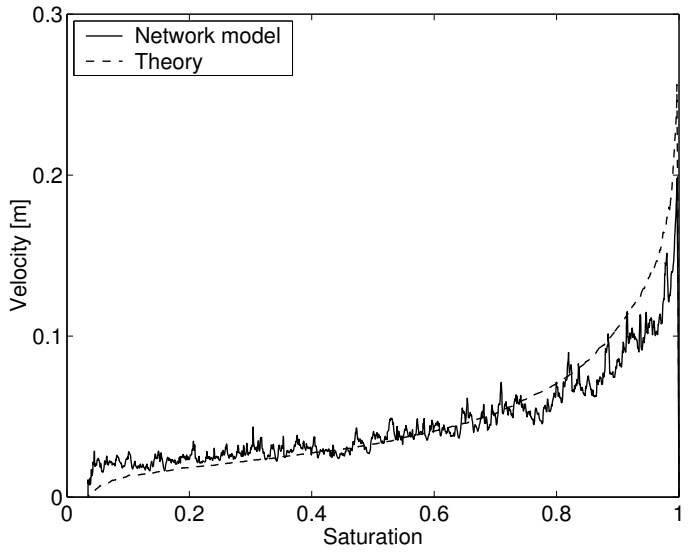


(a)

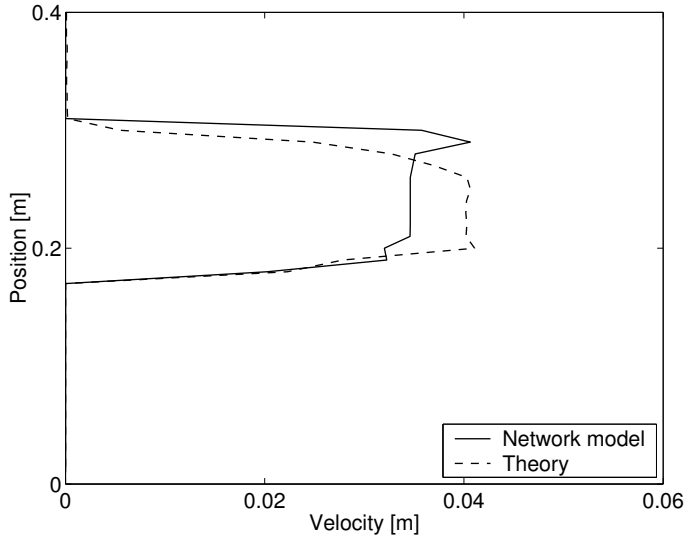


(b)

Fig. 2. Stable displacement – $M_\mu = 10.0$. Comparison of interface velocity in the network model and computed using phase velocities for one REV (a) and sliding REVs (b).



(a)



(b)

Fig. 3. Stable displacement – $M_\mu = 10.0$. Comparison of interface velocity in the network model and computed using phase velocities for one REV (a) and sliding REVs (b). Throat volume and trapped interfaces are not acounted for.

4.4 Unstable displacement

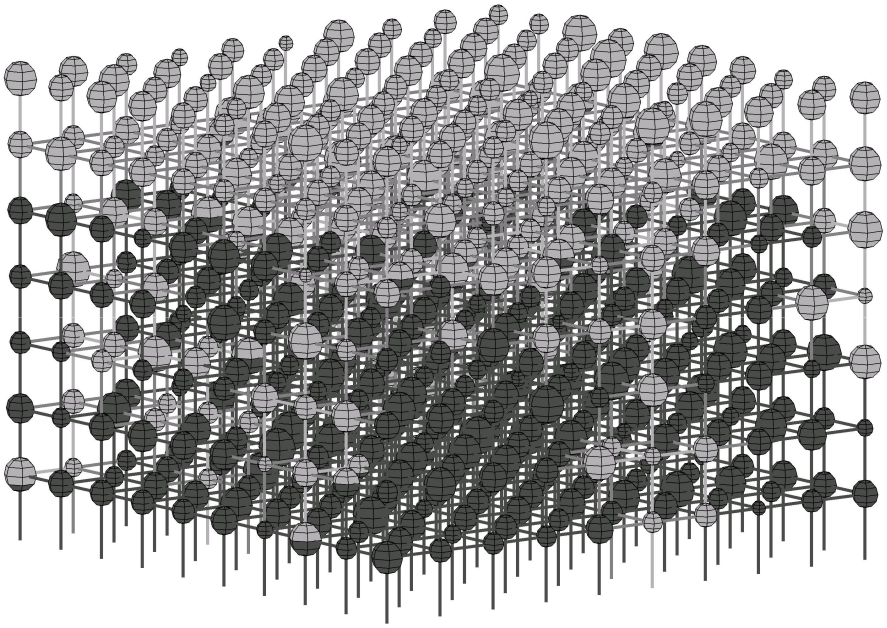
To illustrate the model behavior in the case of unstable displacements, we use a viscosity ratio of 0.1, with the same boundary and initial conditions described above. In this case, we expect a much more irregular front, with possible viscous fingering, and this is essentially what we observe. In Figure 4(a), an irregular and fingered pattern is clear from the fluid occupancy plot of the network. The associated sliding average plot showing average saturation as a function of depth (Figure 4(b)) shows a much more dispersed saturation distribution. Now when the average velocities are calculated, there is a very large disparity between average interfacial velocities and average phase velocities. This results in a poor match between the interfacial velocities predicted by Equation (13) and those calculated in the network model. We observe that in the unstable case, local horizontal flows are more prevalent than in the stable case, due to the highly irregular shape of the invading front. However, the dominant flow direction remains the vertical.

To illustrate some additional difference between the stable and unstable displacements, we have plotted amount of interfacial area as a function of saturation for the viscosity ratios of 10, 1, and 0.1. Figure 6(a) shows the amount of trapped interfacial area as a function of saturation, with the most irregular fronts (viscosity ratio of 0.1) producing the largest amount of trapped interfacial area, and the most stable flat front (viscosity ratio of 10) producing the least amount of trapping. The clear trend is for increasing amounts of trapped interfacial area with decreasing viscosity ratio. In Figure 6(b), the amount of active interfacial area is plotted as a function of saturation, again for the three different viscosity ratios. Here we see a more complex relationship, but the unstable displacements clearly produce the largest amount of active interfacial area, but also produces the highest residual wetting phase saturation. The trend moves consistently to lower amounts of active interfacial area, and lower residual saturations, as viscosity ratio increases.

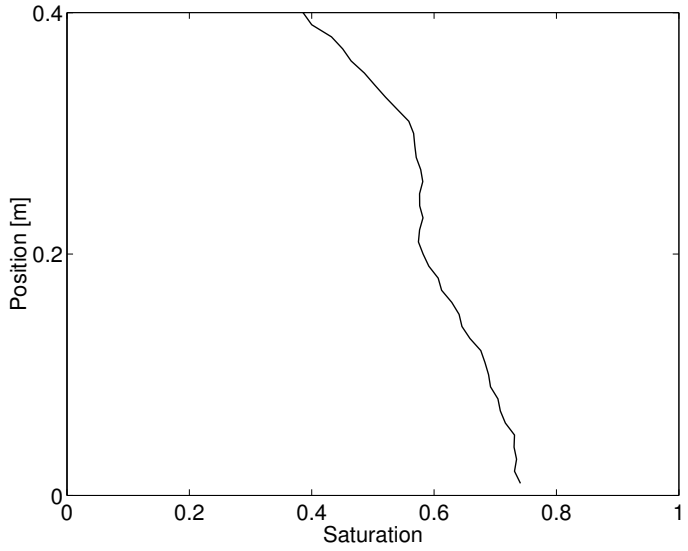
5 Summary and conclusions

The dynamic network model presented herein is relatively simple to implement, and leads to numerical calculations that are straight-forward to implement. The calculation of averages allows many continuum-scale variables to be calculated, and thereby facilitates testing of new theories that involve non-traditional variables. In this work, we have focused on interfacial dynamics, and calculation of average interfacial velocities. We have used these results to test a specific conjecture that relates average interfacial velocity to average phase velocities.

Overall, we conclude that the proposed equation that relates interfacial velocity to phase velocities only hold under very specific conditions, namely piston-like sta-

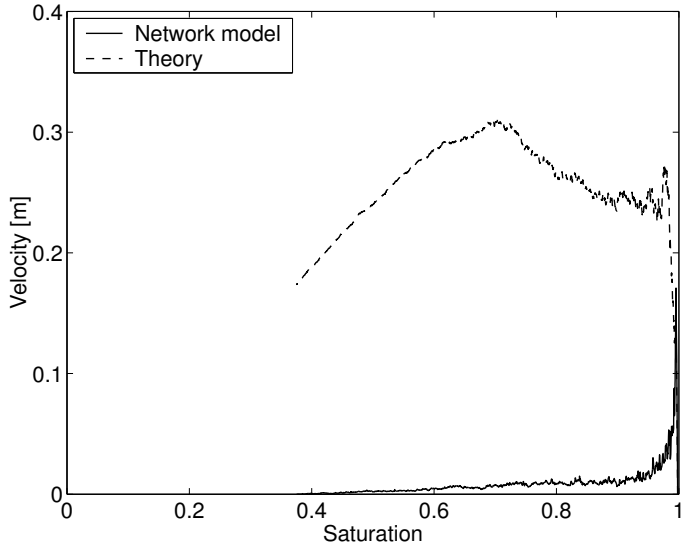


(a) microscale

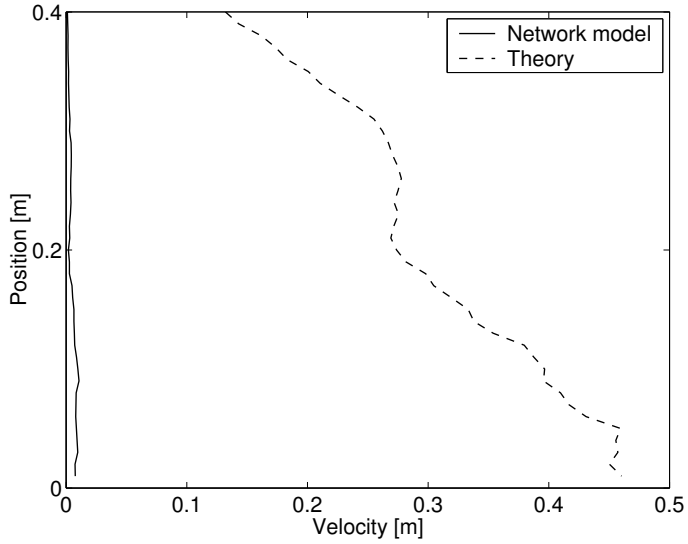


(b) macroscale

Fig. 4. Unstable displacement – $M_\mu = 0.1$. Snapshot of the saturation front.

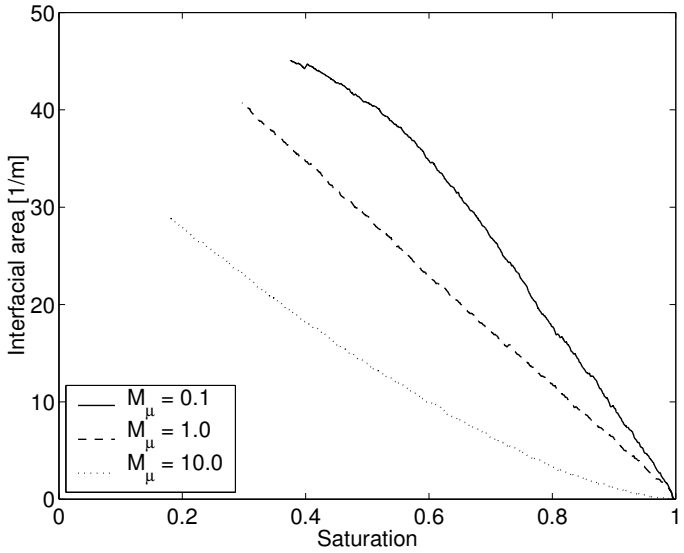


(a)

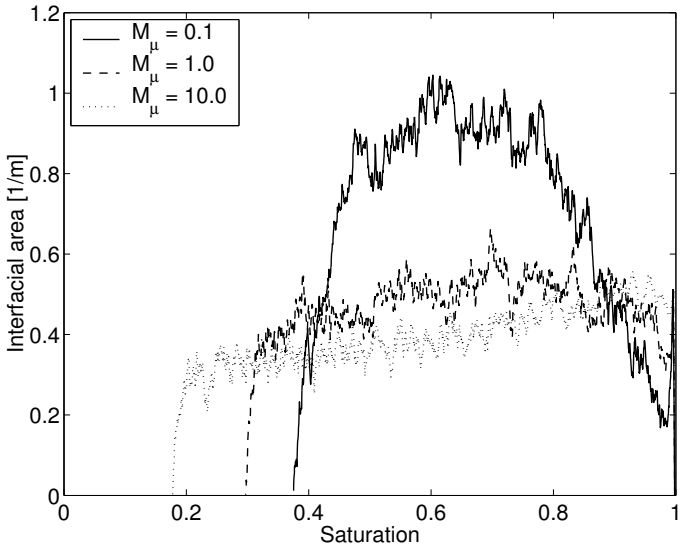


(b)

Fig. 5. Unstable displacement – $M_\mu = 0.1$. Comparison of interface velocity in the network model and computed using phase velocities for one REV (a) and sliding REVs (b).



(a) trapped



(b) moving

Fig. 6. Amount of interfacial area for different viscosity ratios M_μ.

ble displacements in which trapped interfaces are neglected. Under more general conditions, where the invading front is more irregular and piston displacement does not occur at the macroscopic scale, the equation provides a poor prediction. Therefore new constitutive equations to close the extended set of equations for two-phase flow, which include specific interfacial area as a primary variable, need to be developed.

While this work focused on a specific constitutive equation, the general approach is illustrative of the kinds of problems that can be solved with dynamic pore-scale network models. Such models allow local-scale information to be used directly to define averaged, or upscaled, variables. Because the local-scale information is highly detailed, many upscaled variables can be calculated, including nontraditional ones like interfacial area and average interfacial velocity. This greatly facilitates testing of new theories, and has the potential to provide significantly improved insights into fundamental behaviors in two-phase porous media flows.

Acknowledgements

The authors gratefully acknowledge the contributions of W.G. Gray and S.M. Hassanizadeh to this work. Partial support for this work was provided to M.Celia by the National Science Foundation under Grant EAR-9805376 and to H.F. Nordhaug by the Norwegian Research Council (NFR) under grant 116153/431.

References

[1] W. Gray, S. Hassanizadeh, Macroscale continuum mechanics for multiphase porous-media flow including phases, interfaces, common lines and common points, Advances in Water Resources 21 (4) (1998) 261–281.

[2] P. Reeves, M. Celia, A functional relationship between capillary pressure, saturation, and interfacial area as revealed by a pore-scale network model, Water Resources Research 32 (8) (1996) 2345–2358.

[3] R. Held, M. Celia, Modeling support of functional relationships between capillary pressure, saturation, interfacial area, and common lines, Advances in Water Resources 24 (3-4) (2001) 325–343.

[4] K. Heonki, P. Rao, M. Annable, Determination of effective air-water interfacial area in partially saturated porous media using surfactant adsorption, Water Resources Research 33 (12) (1997) 2705–2711.

[5] C. Montemagno, W. Gray, "photoluminescent volumetric imaging: A technique for the exploration of multiphase flow and transport in porous media, Geophysical Research Letters 22 (4) (1995) 425–428.

[6] F. Dullien, Porous Media: Fluid Transport and Pore Structure, 2nd Edition, Academic Press, 1992.

[7] L. Ferrand, M. Celia, The effect of heterogeneity on the drainage capillary pressure - saturation relation, Water Resources Research 28 (3) (1992) 859–870.

[8] P. Reeves, The development of pore-scale network models for the simulation of capillary pressure - saturation - interfacial area - relative permeability relationships in multi-fluid porous media, Ph.D. thesis, Department of Civil Engineering and Operations Research, Princeton University (1997).

[9] M. Blunt, P. King, Relative permeabilities from 2-dimensional and 3-dimensional pore-scale network modeling, Transport in porous media 6 (4) (1991) 407–433.

[10] M. Valavanides, A. Payatakes, True-to-mechanism model of steady-state two-phase flow in porous media, using decomposition into prototype flows, Advances in Water Resources 24 (3-4) (2001) 385–407.

[11] K. Mogensen, E. Stenby, A dynamic two-phase pore-scale model for imbibition, Transport in Porous Media 32 (1998) 299–327.

[12] E. Aker, K. Maloy, A. Hansen, G. Batrouni, A two-dimensional network simulator for two-phase flow in porous media, Transport in Porous Media 32 (1998) 163–186.

[13] T. Dijkstra, G. Bartelds, J. Bruining, S. Hassanizadeh, Dynamic pore-scale network for two-phase flow, in: van Genuchten et al. (Ed.), Characterization and Measurement of the Hydraulic Properties of Unsaturated Soils, 1999, pp. 63–69.

[14] H. Dahle, M. Celia, A dynamic network model for two-phase immiscible flow, Computational Geosciences 3 (1999) 1–22.

[15] H. Nordhaug, H. Dahle, M. Espedal, W. Gray, M. Celia, Two phase flow including interfacial area as a variable, in: L. Bentley, J. Sykes, C. Brebbia, W. gray, P. G.F (Eds.), Computational Methods in Water Resources, Vol. 1, A.A Balkema, 2000, pp. 231–238.