A locally adaptive time-stepping algorithm for petroleum reservoir simulations

by

H. McNamara, G. Bowen, and P.J. Dellar
A locally adaptive time-stepping algorithm for petroleum reservoir simulations

H. McNamara∗,1, G. Bowen2, and P.J. Dellar1

2Schlumberger Oilfield UK Plc., Abingdon Technology Centre
1Oxford Centre for Collaborative Applied Mathematics, Oxford University

February 11, 2011

Abstract

An algorithm for locally adapting the step-size for large scale finite volume simulations of multi-phase flow in petroleum reservoirs is suggested which allows for an “all-in-one” implicit calculation of behaviour over a very large time scale. Some numerical results for simple two-phase flow in one space dimension illustrate the promise of the algorithm, which has also been applied to very simple 3D cases. A description of the algorithm is presented here along with early results. Further development of the technique is hoped to facilitate useful scaling properties.

1 Introduction

The simulation of multi-phase, multi-component fluid flow in oil reservoirs is a tool of key importance for the exploitation of oil resources using enhanced recovery processes. A hierarchy of models of increasing complexity can be employed to describe such processes, including the “classical” Black-Oil model and true compositional models, which may or may not include further effects such as heat transfer and hysteresis in capillary pressures and relative permeabilities. Numerical treatments of such models possess characteristics which make the task particularly complicated and computationally expensive. Fundamental texts regarding such models and their numerics include Aziz and Settari (1979) and Peaceman (1977); considerations of their mathematical form can be found in Trangenstein and Bell (1989b,a) and Chen et al. (2000, 2005).

∗E-mail: mcnamara@maths.ox.ac.uk
In all but the simplest cases, a system of equations describing the transfer of mass (or a proxy such as standard volume), either by phase or by component, is coupled to equations describing the pressure and state relationships of the fluids. Generally the transfer equations for components are predominantly hyperbolic in character (local diffusive effects can be incorporated in more complete descriptions), whereas the pressure equation is parabolic or elliptic. The pressure equation thus couples between geometrically distant regions of the system. Under simplifying assumptions, the pressure variable may be eliminated leaving a hyperbolic system of equations which may admit analytic study. Even in such cases the resulting behaviour is rather complicated, particularly in multi-component models. Examples of the rich dynamics of these problems appear in Juanes and Patzek (2004) and Seto and Orr Jr. (2009).

The numerical implementation of such models to balance the predominantly local transport effects with long-range coupling due to system pressures, which suggests the use of methods with implicit time-stepping. As the equations to be solved are nonlinear, implicit solutions will involve the successive solution of large linear systems in a Newton-based method. If the spatial discretisation of the system results in $N$ computation blocks, and $m$ variables are to be treated implicitly, then these nonlinear solve routines will involve the solution of an $mN \times mN$ linear system a number of times for each time-step taken. The benefit derived from this computational cost is increasing stability as more of the system is treated implicitly.

Some methods which are in common use today include the Implicit Pressure–Explicit Saturations (IMPES) method, in which separate saturation equations are evolved explicitly followed by an implicit update of the pressures and the Fully Implicit Method (FIM), which deals with all variables implicitly. In FIM-type methods the limit on the time-step is related to the convergence of the nonlinear solver, whereas CFL-type constraints provide the limit for more explicit methods such as IMPES. The seminal text by Aziz and Settari (1979) includes details of the FIM and IMPES methods.

The coupling between distant regions to accommodate the pressure behaviour belies the fact that changes in most of the variables are predominantly localised. This localisation results from the development of fluid fronts, as well as effects caused by the presence of wells and by the geometry of the region being simulated. Thus a short time-step is necessitated by behaviours in a small fraction of the simulation, and some form of local adaptivity would be of great benefit to reservoir simulation, allowing the selective application of computational resources to regions where it is of most use.
A number of methods have been proposed, both directly for reservoir simulation and in the related fields of fluid and solid mechanics, to deal with the divergent scales involved. Reservoir simulators can employ the Adaptive Implicit Method (AIM, Forsyth and Sammon, 1986), which uses the computationally cheaper IMPES method in regions where the CFL condition is reasonably large, and the FIM in regions where rapid changes would otherwise require a much smaller time-step. This method is computationally cheaper than the FIM while preserving much of its stability advantage, although the time-steps allowed are still smaller than those for the FIM. In each of these methods a further computational cost is involved in “failed” steps—occasions when the nonlinear (implicit) solver routine fails to converge, leading to a time-step calculation being discarded.

So-called “multi-rate” methods (Gear and Wells, 1984; Constantinescu and Sandu, 2007; Wensch et al., 2009) introduced to deal with systems of ordinary differential equations (such as may be obtained through the semi-discretisation of partial differential equations) use a decomposition of the equations themselves into rapidly evolving and slower components. This procedure is superficially similar to the IMPES method used in reservoir simulation, and may also use an implicit-explicit splitting of the computation. As for the IMPES method, no account is made for spatial variations in rates, or for the variation of rates over time.

Methods involving Adaptive Mesh Refinement (AMR), developed for hyperbolic problems (Berger and Oliger, 1984), use a moving refinement of the discretisation of those geometric regions where rapid changes take place, and often also use a shorter time-step on refined regions of the computation grid, (Domingues et al., 2008, 2009; Fern and Löfstedt, 2006). AMR methods have been applied successfully to problems of multi-component flow in porous media (Hornung and Trangenstein, 1997) however to date only incompressible problems have been treated. Further, it is difficult to construct suitable grids from geological data and the problem of matching the solution at different levels of refinement is far from trivial.

A true space-time adaptive meshing procedure has been proposed for dealing with hyperbolic problems. The so-called Tent Pitcher algorithm and its modifications construct a space-time mesh progressively, using elementary building blocks (for example, in a 2D problem, including time would lead to a 3D mesh which may be composed of tetrahedral blocks) and use a discontinuous Galerkin method on each block (Üngör and Sheffer, 2002; Abedi et al., 2004). The time “front” evolves in a non-uniform manner over the spatial grid, with care taken to ensure that quantities are conserved as needed and that stability constraints are observed. The lack of coupling between regions of the spatial domain suggests that this pro-
procedure would need extensive modification to suit problems of reservoir simulation.

The idea of adapting in the time domain is attractive, however. If the time-scale of behaviour in one region is rapid while in another it is much slower, a procedure whereby extra steps are taken in the rapidly evolving region suggests itself. If local time values can be identified, the solution over the entire space-time domain can be approximated using an “all-in-one”, fully coupled (in space and time) nonlinear system of equations. These equations can (hopefully) be solved using Newton’s method or a related method, provided suitable initial guesses for the solution variables can be found.

The objective of this construction is to reduce the computational expense of performing reservoir simulations. The trade-off is between the number of nonlinear (Newton’s method) solve operations and the size of the system being solved. If the long time-step chosen for the “all-in-one” method is 10 times larger than the step required for an FIM implementation, the size of the resulting all-in-one system should be substantially less than 10 times the simpler FIM system.

We present a description of an algorithm which chooses suitable local time-steps, together with numerical results from some simple test cases. Two one-dimensional water/oil systems are compared with analytical (weak) solutions, and an example of the method working on a simple three-dimensional case is also presented.

2 All-in-one solution method

The algorithm introduced in the previous section consists of two distinct components. The first is a method of choosing relevant points in space-time (or local time-steps) for use in the second component, which is the “all-in-one” nonlinear procedure. While the implementation of both is rather involved, the latter stage of the algorithm is more fundamental to the approach. It is conceivable that a number of different time-step choosers could be used to set up the fully coupled nonlinear solver. We will begin by outlining the “all-in-one” space-time solution method, and then discuss methods for picking local time-steps.

The presentation of the algorithm will focus on a simple two-phase problem in one space dimension. We will consider water and oil flowing in a one-dimensional, homogeneous medium. The two phases under consideration are the wetting, aqueous phase, denoted by \( w \), and the non-wetting oil phase, \( o \). The saturation of phase \( \alpha \) is \( S_\alpha \) and the pressure is \( p_\alpha \). Phase velocities are denoted by \( u_\alpha \), densities are \( \rho_\alpha \) and viscosities \( \mu_\alpha \). The
rock is characterised by its porosity (assumed constant, and for this 1D case expressed as fraction of volume per unit length), $\phi$, and the absolute permeability $K$; and the depth as a function of position is $d$. The relative permeabilities of the rock formation for each phase are $k_{ra}$. These are taken to be simple functions of the phase saturations, neglecting hysteresis effects.

The saturations must satisfy continuity equations

$$\frac{\partial (\phi S_o)}{\partial t} + \frac{\partial u_o}{\partial x} = 0, \quad (1a)$$

$$\frac{\partial (\phi S_w)}{\partial t} + \frac{\partial u_w}{\partial x} = 0. \quad (1b)$$

The phase velocities are given by the usual multi-phase extension of Darcy’s Law where for phase $\alpha$,

$$u_\alpha = -\frac{Kk_{ra}}{\mu_\alpha} \left( \frac{\partial p_\alpha}{\partial x} - \rho_\alpha g \frac{\partial d}{\partial x} \right). \quad (2)$$

The fluids are considered incompressible, and the saturations are expressed as fractions of the pore void space, so that $S_w + S_o = 1$ and $\frac{\partial u_w}{\partial x} + \frac{\partial u_o}{\partial x} = 0$. Thus we can introduce $S = S_o$ and $S_w = 1 - S_o$ and introduce a total velocity $u = u_w + u_o$ such that $\frac{\partial u}{\partial x} = 0$. We further neglect the effects of capillary pressure, so that we can write $p_w = p_o = p$. Defining phase mobilities $\lambda_\alpha = \frac{k_{ra}}{\mu_\alpha}$ and substituting the phase velocities into the continuity equation gives

$$\phi \frac{\partial S}{\partial t} + \frac{\partial}{\partial x} \left( -\lambda_o \left( \frac{\partial p}{\partial x} - \rho_o g \frac{\partial d}{\partial x} \right) \right) = 0, \quad (3a)$$

$$-\phi \frac{\partial S}{\partial t} + \frac{\partial}{\partial x} \left( -\lambda_w \left( \frac{\partial p}{\partial x} - \rho_w g \frac{\partial d}{\partial x} \right) \right) = 0. \quad (3b)$$

This is the form, together with suitable boundary and initial conditions, in which our numerical scheme treats the equations. For the assumptions listed above it is possible to eliminate the pressure from the system entirely leading to the Buckley-Leverett equations with gravity (Buckley and Leverett, 1942). However, to illustrate the local time-step algorithm, which is intended for use in more complicated situations, we will retain the two-equation system. The simplified, single variable system is used to provide exact results for comparison with the numerical results.

### 2.1 Finite-volume formulation

The local time-step algorithm is based on the Finite Volume Method for approximating solutions of the partial differential equations. Considering
the oil equation (3a),
\[ \phi \frac{\partial S}{\partial t} + \frac{\partial}{\partial x} \left( -\lambda_o \left( \frac{\partial p}{\partial x} - \rho_o g \frac{\partial d}{\partial x} \right) \right) = 0, \]
the finite-volume formulation proceeds by using the weak formulation of the equation on a partition of the computational domain. If the region of interest is divided into \( M \) blocks, denoted \( \Omega_i \), for \( i = 0, \ldots, M - 1 \), we integrate the equation over each block to obtain
\[ \int_{\Omega_i} \phi \frac{\partial S}{\partial t} dx + \int_{\Omega_i} \frac{\partial}{\partial x} \left( -\lambda_o \left( \frac{\partial p}{\partial x} - \rho_o g \frac{\partial d}{\partial x} \right) \right) dx = 0. \]
We denote by \( S_i \) the cell averaged value of \( S \) in \( \Omega_i \) and \( \Delta x_i \) the length of \( \Omega_i \). Values of quantities at the left and right boundaries of block \( i \) are labelled with the indices \( i - \frac{1}{2} \) and \( i + \frac{1}{2} \) respectively. We can then write this as
\[ \Delta x_i \phi \frac{d}{dt} S_i + \left( F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}} \right) = 0, \]
where the flux \( F_{i+\frac{1}{2}} \) is the integral \( \int_{\partial \Omega_{i+1}} F(S, p) dx \), the flux of oil through the face of \( \Omega_i \) into \( \Omega_{i+1} \), and \( F_{i-\frac{1}{2}} \) is similarly defined.

The value of a quantity at the \( n \)th time-step is denoted by a superscript, as in \( S_i^n \). The traditional finite-volume method is obtained by integrating this equation over \( [t^n, t^{n+1}] \) and making an assumption regarding the value used to approximate the flux terms. For example the FIM assumes that \( F_{i+\frac{1}{2}} (t) \) is constant over \( [t^n, t^{n+1}] \) and evaluates it at \( t^{n+1} \), so that the formulation is
\[ \phi S_i^{n+1} - \phi S_i^n + \frac{\Delta t^n}{\Delta x_i} \left( F_{i+\frac{1}{2}}^{n+1} - F_{i-\frac{1}{2}}^{n+1} \right) = 0, \]
where \( \Delta t^n = t^{n+1} - t^n \). A similar procedure approximates the water equation. We note that the flux functions \( F_{i+\frac{1}{2}} \) depend upon both \( S \) and \( p \). The two equations (oil and water) and the two variables, together with suitable initial and boundary conditions, comprise a well-posed system. The FIM formulation used here is implicit in both \( S \) and \( p \), and so an iterative nonlinear solution scheme such as Newton’s method must be employed. Concerns regarding the convergence of this scheme and the quality of the solution limit the time-step used for the calculations.

Subtleties of the approximation arise when considering the evaluation of these flux functions. For example, \( F_{i+\frac{1}{2}} \) depends on the values of \( S \) and \( p \) in both cells \( \Omega_i \) and \( \Omega_{i+1} \). The conventional approach, which we employ, is to up-wind the calculation of the relative permeability. Thus if a phase is flowing from left to right (direction of increasing \( x \)), then \( F_{i+\frac{1}{2}} \) uses \( S_i \) to calculate the relative permeability of that phase.
2.2 All-in-one formulation

For the all-in-one formulation, we suppose that we have obtained suitable local time-steps through some procedure. We are interested in the solution on the time interval $[0, T]$ and on the space interval (for our 1D test system) $[0, 1]$. On the $i$th cell of the interval, we have picked $N_i \geq 1$ time values $0 = t^0_i, \ldots, t^N_i = T$. The cell-averaged value of the solution variables $S, p$ on the $n$th local time interval $(t^n_{i-1}, t^n_i)$, $n = 1, \ldots, N_i$ are denoted $S^n_i$ and $p^n_i$.

We return to the basic finite volume formulation (4), we integrate the expression for the $i$th cell over the temporal interval $(t^n_{i-1}, t^n_i)$

$$\Delta x_i (\phi S_i^{n+1} - \phi S_i^n) + \int_{t^n_{i-1}}^{t^n_i} F_{i+\frac{1}{2}} dt - \int_{t^n_{i-1}}^{t^n_i} F_{i-\frac{1}{2}} dt. \tag{6}$$

In the all-in-one formulation, the flux functions $F_{i+\frac{1}{2}}$ and $F_{i-\frac{1}{2}}$ will vary in time (in a piece-wise constant fashion) over the interval $(t^n_{i-1}, t^n_i)$.

Consider first the flux $F_{i+\frac{1}{2}}$. The $(i+1)$-th cell will have its own local times $t^k_{i+1}$, with $k = 0, \ldots, N_{i+1}$. Introduce $k_- = \max\{k : t^k_{i+1} \leq t^{n-1}_i\}$ and $k_+ = \min\{k : t^k_{i+1} \geq t^n_i\}$. We then have intervals $k_-, \ldots, k_+$ on the $(i+1)$-th cell which overlap in time with interval $n$ on the $i$th cell (note that we may have $k_- = k_+$).

We thus approximate

$$\int_{t^n_{i-1}}^{t^n_i} F_{i+\frac{1}{2}} dt = \sum_{k=k_-}^{k_+} \Delta t^k_{i+\frac{1}{2}} F^k_{i+\frac{1}{2}}$$

where the time intervals are defined as $\Delta t^k_{i+\frac{1}{2}} = \min(t^k_{i+1}, t^n_i) - \max(t^{k-1}_{i+1}, t^n_i)$, and the evaluation of $F^k_{i+\frac{1}{2}}$ is at the upper time level $\min(t^k_{i+1}, t^n_i)$. The procedure for approximating $\int_{t^n_{i-1}}^{t^n_i} F_{i-\frac{1}{2}} dt$ is the same.

Using this procedure, we construct a highly coupled system of nonlinear equations. The coupling accommodates the temporal overlap between a time interval on one cell with multiple intervals on the adjacent cells. Note that the same procedure is used for the water equation, and that this procedure can easily be extended to multiple space dimensions (in 3D, for example, there are six interfaces, each of which can have multiple temporal overlaps).

A computer implementation of the all-in-one algorithm proceeds by “flattening” the system, i.e. writing the solution variables, $S^n_i, p^n_i$, with $i = 0, \ldots, M - 1$ and $n = 0, \ldots, N_i$, as a vector $\mathbf{U}$, with components $U_{ij}$, $j = 0, \ldots, 2 \sum_{i=0}^{M-1} N_i$. The equations can be written formally as $\mathbf{R}(\mathbf{U}) = 0$. 

7
Starting from an initial guess $U_0$, Newton’s method iterates to give a sequence of approximations $U_r$. The iterative procedure terminates when $||R(U_r)||$ is less than some convergence criterion, typically using the uniform norm.

As is normally the case with Newton’s method, convergence is only assured if the initial guess, $U_0$ is sufficiently close to the desired solution. Whether or not a given initial guess $U_0$ will converge to a solution in a reasonable number of iterations cannot reliably be determined a priori. Thus any procedure for choosing local time-steps will also need to provide a reasonable initial guess for the solution variables.

### 2.3 Local time chooser

The local time chooser is key to the success of the all-in-one method, and there are a number of possible approaches. The technique described here has proved to work successfully for a number of test cases; however, this is an area of ongoing research and alternatives may supercede this method.

The overall scheme involves three phases. In the first phase, the system is linearised over the large time interval, and a “pressure landscape” is found. The second phase uses the pressure thus found to order the computational cells according to the direction of flow (a topological sorting procedure). Finally, each cell is visited, in this “flow order”, and a crudely applied backwards Euler scheme is used to choose times and estimate solution variables. This procedure is repeated (usually at least once for each phase present in the simulation) until the number of local times chosen remains unchanged. The simulation then passes the local times and solution variables to the all-in-one solver.

We describe the first version of the local time chooser algorithm, which was used to produce the numerical results in the next section. We possess the initial solution variables, $S^0_i, p^0_i$. We seek local times $t^n_i$ such that the saturation change, $\Delta S^n_i = S^n_i - S^{n-1}_i$ is less than a specified value, $\delta S$, supplied to the algorithm. As described above, three phases make up the calculation.

#### First phase—linear step

The first phase is essentially a single Newton method iteration on the time interval $(0, T)$. We write the solution variables as the vector $U = (S_0, p_0, S_1, p_1, \ldots)$. Starting with the initial profile $U^0$, we seek a rough guess for $U^T$, which we will write as $\hat{U}^T$. 
The residual of the fim formulation is defined on each cell as
\[ R_{i,1}(S, p) = \phi S_{i+1}^n - \phi S_i^n + \frac{1}{\Delta x_i} \left( \Delta t^n_i \frac{F_{i+1/2,1}^n - F_{i-1/2,1}^n}{2} \right), \]
\[ R_{i,2}(S, p) = -\phi S_{i+1}^n + \phi S_i^n + \frac{1}{\Delta x_i} \left( \Delta t^n_i \frac{F_{i+1/2,2}^n - F_{i-1/2,2}^n}{2} \right), \]
where subscripts 1 and 2 denote the components, and suitable fluxes \( F_{i+1/2} \) have been defined for the aqueous phase. Each flux function depends upon the solution variables in two cells: for example, \( F_{i+1/2} \) terms depend upon values in cells \( i \) and \( i + 1 \). A single fim step consists of finding the solution of the nonlinear system of equations \( R(U) = 0 \).

Analytic expressions for the components of the Jacobian \( J(U) \) of the function \( R \) are used to take a single step of Newton’s method by solving the linear system of equations
\[ J(U^0)(\hat{U}^T - U^0) = -R(U^0) \]
for the vector \( \hat{U}^T \). While this vector is not a solution to the system of equations, it can act as an starting point for the remainder of the time-choosing procedure. Indeed, its main purpose is to provide a reasonable pressure on which to carry out the subsequent phases of the calculations.

**Second phase—topological ordering**

The pressure “landscape” found in phase 1 is used to impose a topological ordering on the computational cells. One of the fluid phases present in the system is chosen, and the modified pressure (taking into account, for example, gravitational potential) between pairs of cells is computed using the approximate solution values found in phase 1. The sorting considers the computational cells as comprising the nodes of a directed graph, with edges formed by the connections between them. The edge connecting cell \( i \) and cell \( j \) has direction given by the sign of the modified pressure difference. If the higher pressure is in cell \( i \), the edge connecting them is directed from cell \( i \) to cell \( j \).

An implementation of the repeated deletion of sources algorithm (as described in Knuth, 1997) constructs such a topological ordering. In this way we construct an ordering of the cells such that if cell \( n \) is “upstream” of cell \( n \) then it appears earlier in the ordering. The time-choosing algorithm in phase 3 will visit the cells in this order.

**Third phase—choosing times**

The time choosing algorithm has simple and clear aims. On each cell, we seek a sequence of times \( \{t_1, t_2, \ldots \} \) such that an initial “guess” of the
saturation variable at these times satisfies
\[ \Delta S^n_i = S^n_i - S^{n-1}_i \leq \delta S, \]
which is a parameter of the algorithm.

The chooser proceeds as follows. An initial local time is chosen (for these initial investigations the high-level time \( T \) was used here) and denoted \( t^1_i \). The local-time finite volume formulation in (6) is used on the \( i^{th} \) cell, while keeping the solution variables in all adjacent cells constant. This is an implicit definition for the solution variables \( S^1_i \) etc. at the time level \( t^1_i \). A decision is then made whether to accept or reject this proposed time level.

There are two mechanisms by which a time level may be rejected. First, the nonlinear solution of (6) may not converge in a reasonable number of steps. Second, the change in the saturation variable may exceed the parameter \( \delta S \). If it is rejected, the time interval in question is halved and the process repeated on the new interval.

If the time is accepted, then both this value and the solution variables at this time level are stored for use later, and a new time is chosen. If the accepted step is from \( t^{n-1}_i \) to \( t^n_i \), an interval of \( \Delta t^n_i \), then the new time tested will be \( t^n_i + s\Delta t^n_i \), where \( s \) is a parameter of the implementation, usually between 1 and 2 (in our experiments the value chosen was 1.5).

### Iteration

The three phases of the algorithm are repeated until no new times are stored by the chooser phase. The topological sort in phase 2 depends upon the choice of phase. On each iteration the phase used for the sorting is alternated between those present in the system. This is key to the success of the technique on problems involving flows in different directions, such as the gravitational inversion case presented in the next section.

Once the total number of time points in the domain has settled down, the fully-coupled system is taken to be complete. The all-in-one solver can then be used to converge on the desired solution.

### 3 Numerical experiments & results

Numerical experiments were carried out on two test cases of 1D incompressible two-phase flow. These cases are: a waterflood case consisting of a uniform flat reservoir completely saturated with oil subjected to the injection of water from one end at a constant rate; and a gravitational inversion case, in which a vertical column of water initially sits atop a column of oil with no imposed external flow. The first of these cases is the classic Buckley-Leverett model; the second is somewhat artificial, but we
consider it a good test of the algorithm.

The 1D system of equations (3) admits a simplification to a single hyperbolic equation in one variable, of the form

\[ \frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0 \]

with different forms of \( f(u) \) in each case (using the so-called fractional flow formulation as outlined in LeVeque, 2002, chapter 16). Similarity solutions to the Riemann problems were used to construct the solution. Taking \( u \) to represent the water saturation (thus \( u = 1 - S \)), the waterflood case has initial conditions

\[ u(x, 0) = \begin{cases} 
1 & \text{for } x < 0, \\
0 & \text{for } x > 0.
\end{cases} \]  

(7)

The boundary conditions for this case is a constant flux of water through the left \((x = 0)\) boundary, and a constant pressure at \(x = 1\) (the boundary conditions are only used in the derivation of \( f(u) \) in the single equation form, but they are more important for the numerical calculations). For the waterflood problem the depth, \( d \), in (3) is constant.

The gravity inversion case has initial conditions given by

\[ u(x, 0) = \begin{cases} 
1 & \text{for } x < \frac{1}{2}, \\
0 & \text{for } x > \frac{1}{2}.
\end{cases} \]  

(8)

Boundary conditions consist of a constant pressure at \( x = 1 \), as in the waterflood case, and a zero flux condition at \( x = 0 \). The depth in this case is taken to be \( d = x \), so that a constant gravitational force is felt in the positive \( x \)-direction.

Calculations were carried out with the fim and the “all-in-one” system described in the previous section. Estimates of the \( L_2 \)-errors were calculated using cell-averaged exact values. In figures 1 and 2, the approximate solution for the respective methods are plotted along with the exact solution for the first problem. In each image the same meshes are used, with refinement of the mesh producing more accurate approximations. Note that the “step-size” in figures 1–4 are constant within each figure. Similar images for the gravity-inversion case are in figures 3 and 4.

Error estimates for both problems are illustrated in figures 5 and 6. As the overall time-step, \( T \), decreases in the all-in-one method, we expect that fewer and fewer local time points will be chosen, and so the method effectively reverts to fim-like behaviour. This is confirmed by the results shown in the insets of these two figures. The main figures show the scaling
behaviour for these 1D problems. The discrepancy in figure 5 is likely due to the choice of $T$ for the all-in-one calculations.

To illustrate the action of the time-chooser algorithm, the time points chosen for particular simulation runs are shown in figures 7 and 8. The coloured lines on these figures show the characteristic line of the shock(s) which develop in the exact solutions. That the time-chooser algorithm has selected points close to these lines is an indication that it can identify rapid changes in the solution and commit extra computation to such regions.

A further illustration of how the algorithm chooses locally suitable time levels is given in figures 9 and 10. The solution changes between successive time points in the same spatial position is consistently small. It is noted that the algorithm is not (as yet) completely optimal, in that it appears occasionally to select extraneous time points — most clearly illustrated by the second “ray” of points at the bottom of figure 8. At present there is no mechanism by which time points may be discarded — once added to the system they remain in the calculation — so that
Figure 3 – Detail of water solution profile for gravity inversion problem using the FIM.

Figure 4 – Gravity inversion problem using all-in-one method.

erroneous points identified early in the calculation will be carried through to its conclusion.

In order to investigate the accumulation of error over the length of the simulation, the gravity inversion problem was computed on a grid of 200 cells using different sized steps (values of \( T \)). The total time was split into between 1 and 16 steps and the all-in-one method used on each step. The errors when compared to the exact solution, over time, are shown in figure 11. While there is a degree of ordering of the errors (with some exceptions, more steps implies less error), the overall behaviour is clustered quite tightly as demonstrated by the inset plot.

3.1 3D example

To demonstrate that the all-in-one method can be used in more complicated situations, a 3D example is presented. Using a \( 10 \times 10 \times 3 \) grid we computed a miscible gas injection problem. Two phases and two compo-
nents are present in the simulation, and there is mass transfer between the phases. Gas is injected at a constant rate at a corner on the top layer, while a production well with a constant pressure is placed at the opposing corner of the same layer.

Results of the calculation are presented in figures 12 and 13, compared with the output of a widely used commercial simulator. Figure 12 shows the gas phase saturation at a large time from commencing injection. Figure 13 shows the gas saturations for two particular cells over the time of the simulations, again comparing with the existing simulator. A very close match is observed.

Figure 5 – Convergence results and comparison with the \( \text{fim} \) for the Buckley-Leverett waterflood problem. The \( \text{fim} \) errors are in green, the all-in-one errors in blue. The inset shows variation of error with changes in time-step, while the main figure shows variation with changes in mesh size.

Figure 6 – Convergence results and comparison with the \( \text{fim} \) for the second case. In both this and figure 5 the red points in the main figure and inset correspond, as do the purple horizontal lines. There are two lines on the main plot (blue and green as before) which almost coincide.

14
Figure 7 – Time points chosen for a run of the all-in-one method on the Buckley-Leverett waterflood problem. The travelling shock as water is injected from the left is clearly found by the time-chooser algorithm.

Figure 8 – Time points chosen for a run of the all-in-one method for the gravity inversion problem. Two shocks travelling in each direction, feature in the solution.

Figure 9 – Solution in space and time for waterflood problem.
Figure 10 – Solution in space and time for gravity inversion problem.

Figure 11 – Total error against number of steps used for all-in-one method on case (ii). The inset shows the accumulation of error over the respective steps.
Figure 12 – Gas phase saturation during a gas injection simulation with miscible gas. An existing simulation code compares well with the all-in-one result (the simulation times in each case are comparable, but not identical).

Figure 13 – Gas phase saturation against time for two particular cells chosen from the 3D simulation case. Results from the commercial simulator match very closely with results from the all-in-one algorithm.
4 Conclusions & future work

We have demonstrated the all-in-one method for simulating complex flows in porous media such as petroleum reservoirs. The method shows promise, with good convergence properties, and can be used for more complicated situations such as the oil injection case in the previous section.

A number of open questions remain. The implementation of the method is currently at the proof-of-concept stage, and is not tuned for performance. In particular, the time-chooser phase of the algorithm is quite wasteful of compute time as it discards numerous candidate steps. Efforts are being made towards improving the algorithm in this regard by using an alternative method to choose the first time-step on cells at the start of the time-choosing phase. The time-chooser phase of the algorithm could be particularly amenable to small amounts of parallelisation, at least on a shared memory architecture such as current multi-core processors. The topological sort which is carried out could enable time-choosing on different branches of the “cell graph” to be carried out independently. The current implementation is built on parallel aware code, and a future objective would be to implement a fully parallel version of the algorithm.

The transition, on a computation cell level, from a phase being mobile to being immobile, or vice-versa, causes particular challenges to reservoir simulator computations. The sharp and significant change in regime can lead Newton-type methods very far from the desired solution, without guarantee of return. As refinement of the all-in-one algorithm continues, the time-chooser phase could focus particular attention on such events. Placing one or more time points very close to the mobile/immobile transition could improve stability and performance, particularly in the final, all-in-one iterative procedure.

References


Trangenstein, J.A. and Bell, J.B. [1989a] Mathematical structure of com-


RECENT REPORTS

52/10  Order parameters in the Landau-de Gennes theory - the static and dynamic scenarios
       Majumdar

53/10  Liquid Crystal Theory and Modelling Discussion Meeting
       Majumdar
       Mottram

54/10  Modeling the growth of multicellular cancer spheroids in a bioengineered 3D microenvironment and their treatment with an anti-cancer drug
       Loessner
       Flegg
       Byrne
       Hall
       Moroney
       Clements
       Hutmacher
       McElwain

55/10  Scalar Z, ZK, KZK, and KP equations for shear waves in incompressible solids
       Destrade
       Goriely
       Saccomandi

56/10  The Influence of Bioreactor Geometry and the Mechanical Environment on Engineered Tissues
       Osborne
       ODea
       Whiteley
       Byrne
       Waters

57/10  A numerical guide to the solution of the bidomain equations of cardiac electrophysiology
       Pathmanathan
       Bernabeu
       Bordas
       Cooper
       Garny
       Pitt-Francis
       Whiteley
       Gavaghan

58/10  Particle-scale structure in frozen colloidal suspensions from small angle X-ray scattering
       Spannuth
       Mochrie
       Peppin
       Wetlaufer

59/10  Spin coating of an evaporating polymer solution
       Munch
       Please
       Wagner

60/10  Stochastic synchronization of neuronal populations with intrinsic and extrinsic noise
       Bressloff
       Lai

61/10  Metastable states and quasicycles in a stochastic Wilson-Cowan model of neuronal population dynamics
       Bressloff

62/10  Adsorption and desorption dynamics of citric acid anions in soil
       Oburger
       Leitner
       Jones
       Zygalakis
       Schnepf
       Roose

63/10  A dual porosity model of nutrient uptake by root hairs soil
       Zygalakis
       Kirk
       Jones
<table>
<thead>
<tr>
<th>Date</th>
<th>Title</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/11</td>
<td>Improving the efficiency of optical coherence tomography by using the non-ideal behaviour of a polarising beam splitter</td>
<td>Lippok Nielsen Vanholsbeeck</td>
</tr>
<tr>
<td>02/11</td>
<td>Self-diffusion in remodelling and growth</td>
<td>Epstein Goriely</td>
</tr>
<tr>
<td>03/11</td>
<td>Spontaneous rotational inversion in Phycomyces</td>
<td>Goriely Tabor</td>
</tr>
<tr>
<td>04/11</td>
<td>From individual to collective behaviour of coupled velocity jump processes: a locust example</td>
<td>Erban Haskovec</td>
</tr>
<tr>
<td>05/11</td>
<td>Solving Eigenvalue problems on curved surfaces using the closest point method</td>
<td>MacDonald Brandman Ruuth</td>
</tr>
<tr>
<td>06/11</td>
<td>A numerical methodology for the Painleve equations</td>
<td>Fornberg Weideman</td>
</tr>
<tr>
<td>07/11</td>
<td>Strong stability preserving two-step Runke-Kutta methods</td>
<td>Ketcheson Gotlieb MacDonald</td>
</tr>
<tr>
<td>08/11</td>
<td>Hysteresis and Post Walrasian Economics</td>
<td>Cross McNamara Kalachev Pokrovskii</td>
</tr>
</tbody>
</table>

Copies of these, and any other OCCAM reports can be obtained from:

Oxford Centre for Collaborative Applied Mathematics
Mathematical Institute
24 - 29 St Giles’
Oxford
OX1 3LB
England
www.maths.ox.ac.uk/occam