Extension of Streamline-Based Dual Porosity Flow Simulation to Realistic Geology

MARCO R. THIELE, ROD P. BATYCKY
Streamsim Technologies, Inc.

MARTIN IDING
Statoil

MARTIN BLUNT
Imperial College London

Abstract

In this paper we extend the dual porosity streamline-based flow simulation method to complex geological grids containing both single and dual porosity gridblocks with multiple relative permeability regions. We use the streamline dual porosity formalism of *Di Donato et al.*[1] in conjunction with the standard finite-difference simulation capillary pressure driven matrix-fracture transfer model.[8]

Geological heterogeneity and mixed single/dual porosity gridblocks lead to highly irregular cell volumes along each streamline and require extending the *Di Donato et al.* approach in two ways: (1) improve the discretization of the 1D problem along the streamlines by "binning" very small cells with larger ones rather than using a regular discretization; and (2) use an Adaptive-Implicit (AIM) formulation along the streamlines to cope with small cells experiencing large volume throughputs. Regardless of how extreme the binning approach might be, an AIM method is still required because it is not possible to group all small cells to a desired volume tolerance, as single porosity matrix cells cannot be merged with dual porosity cells nor can neighboring flowing cells of different relative permeability regions be merged. Numerical tests show that only a small amount of binning and a modest cutoff for implicitness can have a very favorable impact on 1D solver efficiency and overall solution times, without noticeable loss of detail on the full 3D solutions.

We finish with a large 3D 280,000 gridblock model adapted from the SPE10[2] test case that contains both single porosity and dual porosity regions as well as multiple relative permeability regions. Comparisons with a standard grid-based simulation showed that the streamline-based method, while properly modeling imbibition of water into the matrix, tended to also show more channeling along high permeability fracture regions giving earlier breakthrough times. We attribute this to the reduced amount of smearing in the flood front observed with the streamline simulation as compared with the finite-difference simulation. For this 280,000 cell model the streamline method was 9 times faster than the finite-difference simulation.

Introduction

Streamline-based (SL) simulation is now an accepted technique for flow modeling that complements more standard finite-difference (FD) methods. See Thiele[19] for a thorough overview of SL simulation. Streamlines have been shown to work well on large heterogeneous full-field models where the primary recovery mechanism is fluid displacement via injectors and/or aquifer support[1][14][16]. Results show that for these types of models, SL simulation is one to two orders of magnitude faster than FD methods. SL simulation additionally provides novel information such as flow visualization, well drainage regions, and well allocation factors—all useful in history matching and optimizing field performance. However, SL theory had been confined to single porosity (SP) models thus limiting the technique to non-fractured reservoirs. Yet it is precisely field-scale fracture systems that require significant computer resources at realistic grid resolutions and where SL simulation would be an excellent complement to existing simulation approaches.

Recently *Di Donato et al*.[2] extended the SL theory to fractured media using the dual porosity (DP) approach. The details of streamline simulation and the extension to DP theory are not covered in this paper. In brief, their method accounts for capillary-controlled imbibition via a transfer function between the matrix & fracture. Because they assume incompressible flow, the pressure solve is only a function of the flowing medium properties of each gridblock and is unaffected by the transfer function. A transfer function is included as a source/sink term along the one-dimensional conservation equation for each streamline as part of the solution of the mass transport problem. *Di Donato et al.* implemented a linear transfer function, the *Kazemi et al*[11] transfer function, and the capillary pressure driven model found in conventional grid-based simulation methods.[8] Other dual porosity effects such as gravity-drainage or oil expansion in the matrix due to compressibility were not considered, nor are they addressed here. *Al-Huthali and Datta-Gupta*[6] have also presented a DP and dual permeability using streamlines where gravity-drainage effects were included in the transfer function. The work of *Di Donato et al.* and *Al-Huthali and Datta Gupta* demonstrate that streamlines can be extended to dual porosity systems and can yield similar recovery behavior to grid-based methods[5,6]. However, because runtimes with conventional FD methods scale approximately quadratically with respect to the number of grid cells while the SL method scales approximately linearly, the SL method was shown to be one – two orders of magnitude faster than FD methods at appreciable grid resolution. For the largest models studied by *Di Donato et al*. (1.1 and 4.4 million cells), the dual porosity system was too big to be run in any reasonable time with a FD method on a high-end single CPU computer. In these cases the SL method, although making some assumptions, was the only solution possible.

Di Donato et al.'s work laid the foundation for SL simulation extended to DP systems. The key idea in *Di Donato et al.* is that there is a flowing medium and a non-flowing medium. A streamline always traverses the flowing medium, and interacts with the non-flowing medium through a transfer function. Thus a SP-block can either be a "matrix" block or a "fracture" block and both are simply treated as a flowing medium along the streamline. Conversely, a DP-block will have a flowing medium (the fracture) represented along the streamline and a non-flowing medium (the matrix) coupled to the streamline via a transfer function. See Figure 1 for a graphic representation of this concept.

The geological model studied by *Di Donato et al.* was restricted to a variation of the 10th SPE Comparative Solution Project on Upscaling.[2] The DP variation of SPE10 was assumed to contain only dual porosity cells with constant matrix properties, constant fracture porosity ϕ_f, and

a heterogeneous fracture permeability (k_f) field with a constant transfer function. The intent of this paper is to extend *Di Donato et al.*'s work to more realistic geological systems that contain single and dual porosity cells, and heterogeneous porosity and permeabilities for both the matrix and the fractured regions. As discussed in the next section, the extensions presented in this work are confined to (1) how the 1D streamlines are pre-processed before being passed to the 1D solver; and (2) to modifications in the 1D numerical solution itself. All else is the same as outlined by *Di Donato et al.*[1-3]. We finish by providing DP streamline results on our own variation of SPE10 but with multiple relative permeability regions, SP/DP cells, and a varying matrix-fracture transfer functions multiplier, and compare the results with grid-based simulation using Eclipse.[17]

Impact of Realistic Grids on 1D Streamline Discretization

The key step in streamline simulation is the construction of a 1D problem that is solved numerically along each streamline. As outlined in Thiele[10], the 1D equation to solve along each streamline is in the form

Equation 1:
$$\frac{\partial M_i}{\partial t} + \frac{\partial F_i}{\partial \tau} = 0, \quad i = 1, n_c .$$

For incompressible flow the flux along each streamline, q^{sl}, can be assumed to be a constant and we can relate time-of-flight (TOF), τ, to volume, rewriting the 1D equation as,[10]

Equation 2:
$$\frac{\partial M_i}{\partial t} + q^{sl} \frac{\partial F_i}{\partial V} = 0, \quad i = 1, n_c ,$$

which is similar to the familiar 1D Buckley-Leverett equation. Equation 2 states that once discretized, each streamline will be composed of discrete cells whose volumes vary. This is due to the fact that the $\Delta\tau$ a streamline encounters across each gridblock it traverses as it is traced from an injector to a producer varies due to clipping gridblock corners and velocities being very large near source/sinks and small away from source/sinks. Even homogenous 3D systems can give rise to ΔV's varying significantly along streamlines. For realistic geological models with heterogeneous rock properties the variability in volumes along each streamline is amplified and for models containing areas of both SP and DP cells, and heterogeneous matrix and fracture properties, variability is amplified further.

As illustrated in Figure 1, the SP/DP regions can give rise to both, very small volume and large volume cells, along the 1D streamline which leads to a highly irregular 1D profile. Because very small cells with high throughputs can cause a 1D explicit solver to take very small time steps, small cells need to be removed—either through regularization as done by *Di Donato et al*., or through binning of smaller cells with larger cells. Regularization and binning are possible when all cells are SP or DP and relative permeability regions are a constant, but for realistic grids with mixed SP/DP regions, a streamline will have both SP and DP cells, and relative permeability regions in the flowing medium that are not constant along the 1D path. As discussed next, a combination of binning of small cells and an adaptive-implicit 1D solver are proposed in this paper to efficiently solve the highly irregular solutions along each streamline.

Binning the 1D Irregular Solution

One method of eliminating very small volume cells that occur along a streamline's 1D irregular cell profile is to bin (merge) small cells with a neighboring larger cell and repeat the process until a minimum merged cell volume tolerance is reached. For almost all 3D models studied, the net effect of binning very small cells together creates less of a mixing artifact than that caused by multiple streamlines mixing within each gridblock at each 3D time step. Yet the removal of these small cells can greatly improve the efficiency of the 1D transport solution along each streamline and speed-up the 1D solution with essentially no difference in the final 3D solution. But care must be exercised when setting a binning tolerance, because if the volume tolerance is set too large and too much binning occurs, this has the effect of possibly smearing or advancing injection fronts well ahead of their actual position.

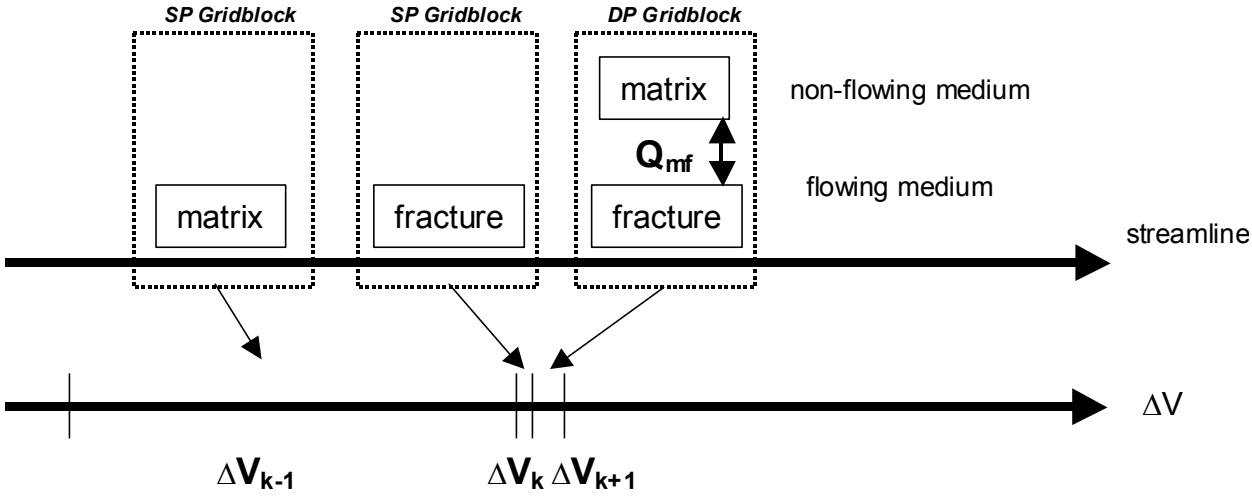


Figure 1: Conceptual drawing of gridblocks intersected by a streamline and the resulting 1D irregular solution.

Models with multiple relative permeability regions introduce an additional complication when binning. A binned cell can only be assigned a single set of relative permeabilities and residual saturations. Additionally, the averaging of saturations for a merged cell can result in saturations that are below the residuals of one of the original cells. Thus, it is preferable to not merge cells together that do not share the same relative permeability curves for either the flowing or non-flowing medium. This is why using an AIM solver is important, as some small cells cannot be merged without introducing non-physical flow effects.

For binning we use the following rules, where cell k is identified as the current cell whose volume is less than some tolerance, the left adjacent cell is k-1, and the right adjacent cell is k+1:

1. Cannot merge a SP matrix cell with a SP fracture cell. This is because the fracture relative permeability curves are rarely if ever the same as matrix relative permeability curves.
2. Cannot merge a SP matrix cell with a DP cell. As in 1 above, the relative permeability curves between the two regions are not the same. Secondly, the matrix-fracture transfer

function applied to the resulting merged cell is no longer defined as the matrix is the flowing medium in one cell, but the non-flowing medium in the adjacent cell.

3. Can merge a SP fracture cell with a DP fracture cell. While this may have the undesired affect of connecting a fracture with the fracture and matrix of a neighboring cell, it does allow for binning of small SP fracture cells bounded on either side by DP cells, which can give improvements in solver performance.
4. When merging SP cells together, and cells k-1, k, & k+1 all share the same relative permeability regions, merge cell k with the smaller volume adjacent cell. Otherwise, first attempt to merge with either the right or left cell that shares the same relative permeability region as cell k. If neither neighbor shares the same relative permeability region, then merge with the neighbor with the smaller volume and assign the relative permeability region of the adjacent neighbor to the newly merged cell (retain the region of the larger volume cell).
5. When merging DP cells together apply the method of item 4 above. However, when merging the matrix portions of the DP cells together, retain the relative permeability region with the highest residual oil saturation.
6. Fluid saturations for a merged cell are based on a volume weighting of the saturations of the irregular cells that define the merged cell.
7. The constant coefficient of the transfer function for a merged cell is based on the non-flowing pore volume weighted average of the irregular cells that define the merged cell.

The Adaptive Implicit 1D Solver

The literature on adaptive implicit methods (AIM) is well documented and the reader is referred to an extensive literature for details[7],[20]. The AIM 1D-solver used here is significantly simplified by assuming a) an incompressible system and no dispersion thus making the transport problem hyperbolic; b) the conservation equations are written in terms of total mass; c) single-point upstream weighting is used for the discretization; and d) the problem is strictly 1D along streamlines and the upstream direction is always known.

For any SP-cell k we can discretize Equation 2 using a single-point upstream weighting scheme and implicit mass fluxes as

Equation 3:
$$M_{i,k}^{n+1} = M_{i,k}^{n} + \frac{q\Delta t}{\Delta V_k}\left(F_{i,k-1}^{n+1} - F_{i,k}^{n+1}\right)$$

where n+1 is the new time level and n is the current time level of the 1D equation, and

Equation 4:
$$M_{i,k} = \left(\sum_{j=1}^{n_p} \rho_j x_{i,j} s_j\right)_k$$

Equation 5:
$$F_{i,k} = \left(\sum_{j=1}^{n_p} \rho_j x_{i,j} f_j\right)_k$$

We can expand the downstream flux at the current iteration level (m+1) as,

Equation 6:

$$F_{i,k}^{m+1} = F_{i,k}^{m} + \sum_{l=1}^{n_c} \left(\frac{\partial F_{i,k}}{\partial M_{l,k}} \right)^{m} \Delta M_{l,k},$$
$$\Delta M_{l,k} = M_{l,k}^{m+1} - M_{l,k}^{m}$$

which results in a $n_c \times n_c$ system of linear equations in terms of $\Delta M_{i,k}, i = 1, n_c$. Newton's method[13] is then used to solve for the flux at the new time level. Because of the hyperbolic nature of the conservation equations and the fact that the equations are solved in 1D, the solution progresses from upstream to downstream sequentially with the upstream flux always being known at the new time level. If a block is tagged as explicit, then the flux of the cell is trivially calculated using $M_{i,k}^{n}$. If the cell is tagged as implicit then Newton's method is used to solve for $M_{i,k}^{n+1}, F_{i,k}^{n+1}$ simultaneously.

Matrix-Fracture Source Term

In a DP-cell, the transport is assumed to occur in the fracture with a source term accounting for the transfer of oil/water between the matrix and the fracture. Thus, the discretized conservation equation for the fracture block (subscript *f*) is re-written as

Equation 7:

$$M_{i,k}^{n+1}\Big|_f = \left[M_{i,k}^{n} + \frac{q\Delta t}{\Delta V_k} \left(F_{i,k-1}^{n+1} - F_{i,k}^{n+1} \right) \right]_f + \Delta t T_{i,k},$$

where as for the matrix block (subscript *m*) we have that that

Equation 8:

$$M_{i,k}^{n+1}\Big|_m = M_{i,k}^{n}\Big|_m - \Delta t T_{i,k}.$$

Ideally, the matrix-fracture source term should be solved implicitly. On the other hand, the different time scales between the flow in the fracture and the exchange of oil/water between matrix and fracture allow a looser coupling. We therefore use an operator splitting approach in that we first solve

Equation 9:

$$M_{i,k}^{n+}\Big|_f = \left[M_{i,k}^{n} + \frac{\Delta t}{\Delta \tau} \left(F_{i,k-1}^{n+1} - F_{i,k} \right) \right]_f,$$

either explicitly or implicitly for the intermediate fracture masses $M_{i,k}^{n+}$, then solve

Equation 10:

$$M_{i,k}^{n+1}\Big|_f = M_{i,k}^{n+}\Big|_f + \Delta t T_{i,k}^{n+} \;;\; M_{i,k}^{n+1}\Big|_m = M_{i,k}^{n}\Big|_m - \Delta t T_{i,k}^{n+},$$

where the transfer function is now evaluated at the intermediate fracture masses $M_{i,k}^{n+}$. The transfer function used in this work is driven by a capillary pressure difference between matrix and fracture of the form

Equation 11:

$$T_i = \rho_i \beta \frac{\lambda_{w,f} \lambda_{o,m}}{\lambda_{w,f} + \lambda_{o,m}} \left(P_{c,f} - P_{c,m} \right),$$

where λ is the mobility of the oil/water (subscript *o,w*) in the fracture/matrix (subscript *f,m*) and P_c is the capillary pressure in the fracture/matrix (subscript *f,m*).

Level of Implicitness & Time Step Selection Along Streamlines

A number of methods have been proposed to decide if a cell is implicit or explicit[9],[15]. Here we use a very simple throughput tolerance defined by

Equation 12: $$\Delta V_k < V_{ai}^{tol} q_j^{sl} \Delta T$$

where $q_j^{sl}\Delta T$ represents the total volume throughput for streamline j at the 3D global time step size ΔT, ΔV_k is the pore volume of the cell, and ΔV_{ai}^{tol} the volume tolerance for the AIM—an input parameter. Thus, if the pore volume is smaller than the volume throughput times a tolerance, the cell is flagged as being implicit. This check is done only once per streamline per 3D time step. The time step size Δt used along a streamline to solve the transport problem over the 3D time step ΔT (streamlines are updated every ΔT) is simply the time step size dictated by the smallest explicit cell along the streamline.

Practical Implementations in the AIM-Solver

Compared to the explicit formulation, every implicit block requires a significant amount of extra work with the additional risk on non-convergence of Newton's method. This must be weighted against the potential of taking a much larger time step across a cell that might otherwise severely restrict the time step if treated explicitly. A smooth implementation of the AIM solver must take care to balance these two criteria. In this work, we have implemented the following practical features to ensure robustness and efficiency of the algorithm:

1. The initial guess for the start of the Newton loop is an explicit step with a reduced time step size.
2. The Jacobian is evaluated every third Newton iteration.
3. A line search follows each Newton iteration to ensure global convergence and a steady reduction of the residual.

1D DP Examples

To test the AIM solver, we used four 50x1x1 DP systems in which the only difference was the block volume of the individual cells. All systems had the same total pore volume. The first system used a uniform grid and the remaining 3 used randomly distributed grids with progressively increasing differences between the volume of the smallest and the largest cell. Statistics of the cells volumes are given in Table 1.

	Uniform (U)	Random1 (R1)	Random2 (R2)	Random3 (R3)
Average Volume	**10**	**10**	**10**	**10**
Min Volume	**10**	**0.64**	**0.0071**	**0.00037**
Max Volume	**10**	**50.2**	**149.00**	**280.00**

Table 1: Pore volumes for the 4 test systems used. The histograms for R1, R2, and R3 are shown in Figure 2.

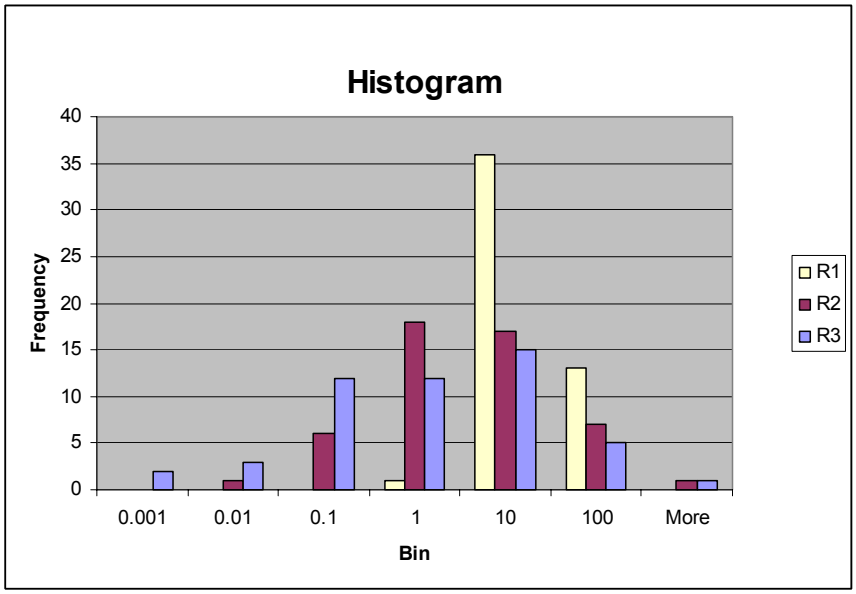


Figure 2: Histogram for the block pore volumes associated with the three random systems used for testing the AIM-solver.

To see the impact on run times we used different throughput tolerances (V_{ai}^{tol}) equal to 0.0, 0.001, 0.01, and 0.1. The number of explicit cells remaining after applying the volume tolerances are given in
Table 2. Run times in minutes associated with each 1D system are given in Table 3 below.

Number of Explicit Cells as a Function of Throughput Tolerances				
V_{ai}^{tol}	Uniform (U)	Random (R1)	Random (R2)	Random (R3)
0.000	**50**	**50**	**50**	**50**
0.001	**50**	**50**	**42**	**25**
0.01	**50**	**43**	**18**	**11**
0.1	**50**	**6**	**7**	**4**

Table 2: Number of explicit cells after applying the throughput tolerance V_{ai}^{tol}.

Run Times as a Function of Throughput Tolerances (in min)				
V_{ai}^{tol}	Uniform (U)	Random (R1)	Random (R2)	Random (R3)
0.000	**0.06**	**0.22**	**15.42**	**********
0.001	**0.06**	**0.22**	**0.49**	**0.52**
0.01	**0.06**	**0.07**	**0.12**	**0.11**
0.1	**0.06**	**0.05**	**0.05**	**0.05**

Table 3: Run times of each system as a function of the throughput tolerance V_{ai}^{tol}.

The importance of the results in Table 3 is best illustrated by R3 ($V_{ai}^{tol} = 0$), which cannot be solved without an AIM formulation. Keeping all blocks explicit for R2 leads to an unacceptable run time of 15.42 minutes for this small 50x1x1 system, while even a small tolerance of 0.001

allows the systems to be solved more efficiently (a speed-up by a factor of 31 for R2—from 15.42 mins to 0.49 mins). At a tolerance of 0.1 there is little difference in run times between the various cases, and all examples have an equivalent water breakthrough profile as shown in Figure 3.

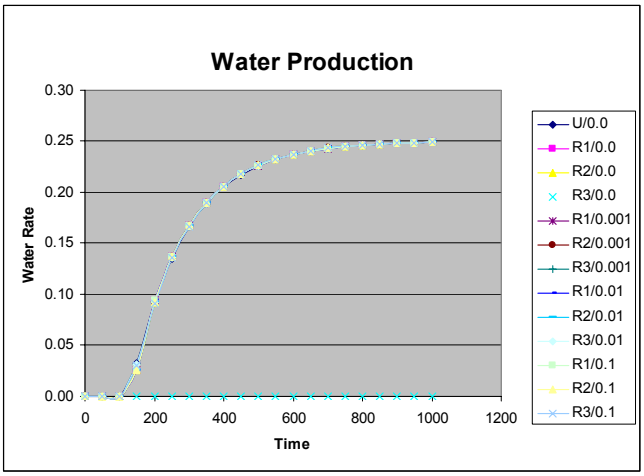


Figure 3: Water production profiles for all the runs of Table 2 & Table 3.

A 3D DP Synthetic Example

Here we will illustrate the effects of binning and the level of implicitness in the 1D solver on the overall solution time and production profile of a 3D system[18]. For comparison purposes we studied the 20x55x17 10th SPE Comparative Solution Project model used by *Di Donato et al*. Although this model does not have SP/DP regions, it is 3D, has a heterogeneous fracture permeability distribution, and a divergent/convergent flow field due to the source/sinks. Thus this model does give variable volume cells along each streamline traced at each timestep, although the variability will not be as extreme as in a SP/DP system. We choose this test model because there are no restrictions on binning cells due to varying DP/SP or relative permeability regions, which could obscure the effect of varying the binning tolerance, V_{bin}^{tol}. Approximately 2500 streamlines were tracer per timestep per model, and the ratio of largest to smallest cell volume for each streamline ranged from approximately 1 to 7 orders of magnitude.

To illustrate the sensitivity to the binning volume tolerance, we forced the 1D solver to remain explicit ($V_{ai}^{tol} = 0$), and varied V_{bin}^{tol} from $1\text{x}10^{-4}$ to 0.1. To illustrate the sensitivity on level of implicitness, we fixed $V_{bin}^{tol} = 1\text{x}10^{-4}$, and varied V_{ai}^{tol} from 0 to 0.10. Field water production profiles are shown in Figure 4, and timing results, using a P4-2.2Ghz PC, are shown in Figure 5.

Figure 4 shows that binning irregular cells can affect the breakthrough profile as the tolerance is increased beyond a certain threshold. In Figure 4, this happens as the tolerance changes from 0.01 to 0.1 and is expected—binning very small cells will do little to smear the waterfront along each streamline but binning larger cells will. Thus excessive binning will cause mixing and smearing that can change the production profile. The smearing effect due to binning might still be small however compared to mixing that occurs at the 3D level between streamlines for each global timestep when all streamline saturations are mapped back to the 3D grid. Also note that while the binning tolerance has little impact on the solution as it is varied from $1\text{x}10^{-4}$ to 0.1, it has a large impact on run times as shown in Figure 5. Run times were reduced from about 95

minutes to 5 minutes (19x faster) just by removing cells smaller than 0.01 of the total streamline volume.

Figure 4 shows that the overall field breakthrough profile is fairly insensitive to the tolerance used to toggle cells from explicit to implicit. This is expected for two reasons: (1) the 1D results in Figure 3 showed little difference in production profiles, as the level of implicitness was varied; and (2) the slight increase in numerical diffusion in the 1D solver due to more cells handled implicitly is masked by the mixing effect between streamlines at the 3D level that occurs at each global timestep. Figure 5 shows that if very small cells are not binned and allowed to pass to the 1D solver then the AIM tolerance can have a great impact on run time. At a fixed $V_{bin}^{tol} = 1\text{x}10^{-4}$ CPU time was reduced from about 95 minutes to 9 minutes (10.5x faster) as V_{ai}^{tol} changed from 0 to 0.10. Similar dependence on V_{ai}^{tol} was seen in the previous section also. For $V_{bin}^{tol} > 0.01$ on the other hand, the level of adaptive implicitness had almost no effect on run time because small cells are already removed by the binning algorithm and few cells are treated implicitly.

One might conclude from this example that binning alone is enough to improve runtime without affecting results and that an AIM solver is not necessarily needed. Recall that this model does not contain SP/DP regions or varying relative permeability regions, so there are no cells ever encountered that cannot be binned to meet a desired tolerance. For grids with SP/DP regions, it is quite possible that very small cells cannot be binned with a neighboring cell and thus need to be passed to the 1D solver as is. Because any one very small cell when compared to throughput can force the 1D solver to take very small time steps and thereby slow the entire simulation—in certain cases radically—it is important to handle these cells via the AIM method.

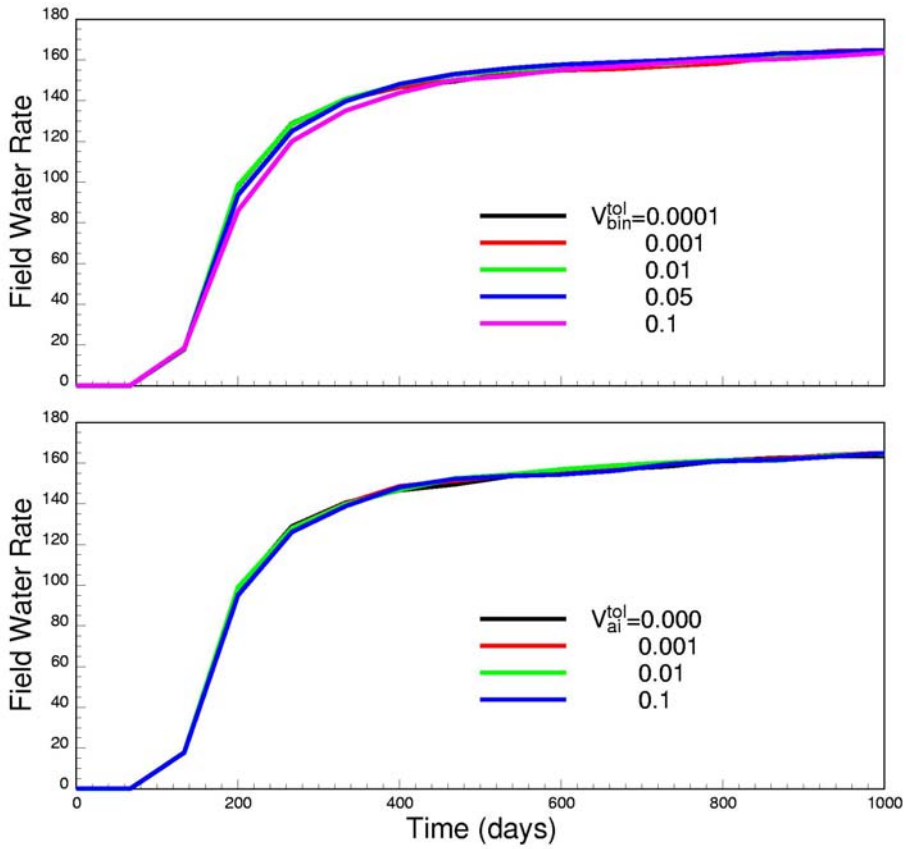


Figure 4: 3DSL field water production rate sensitivity to binning tolerance and adaptive implicit tolerance.

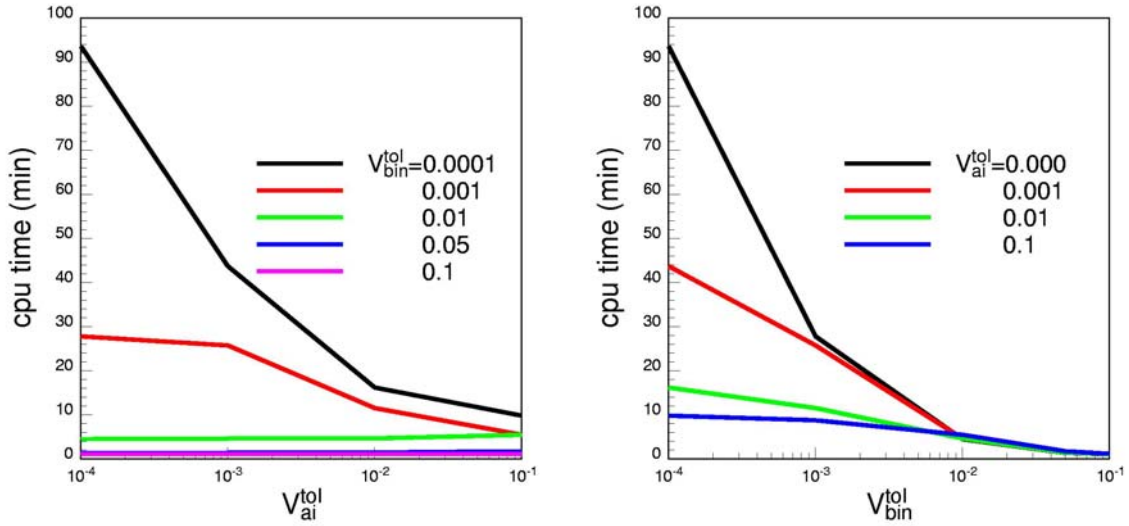


Figure 5: CPU time sensitivity to binning tolerance and adaptive implicit tolerance.

A 3D SP/DP Example

We conclude with an example containing more realistic geology, constructed by modifying the SPE10 data set. This example is based on a 60x220x17 grid with a permeability field upscaled from the original 60x220x85 model. The upscaled permeabilities were used to construct both a matrix and fracture permeability field, and multiple relative permeability regions (Figure 6). Specifically, the matrix permeability field was computed as being $1/10^{th}$ of the upscaled permeabilities and the fracture permeability field was 10 times the upscaled permeabilities. Matrix was assumed to be present throughout the entire model but in areas where the upscaled permeability was <10 mD, fractures were assumed to not be present (SP gridblocks). In total there were 126,000 DP gridblocks and 98,000 SP gridblocks. Relative permeability regions were created as follows: the fracture system represented one rock region (straight line fractional flows), matrix blocks in the DP region were divided into 3 regions based on permeability cutoffs, and SP gridblocks were given their own relative permeability curves.

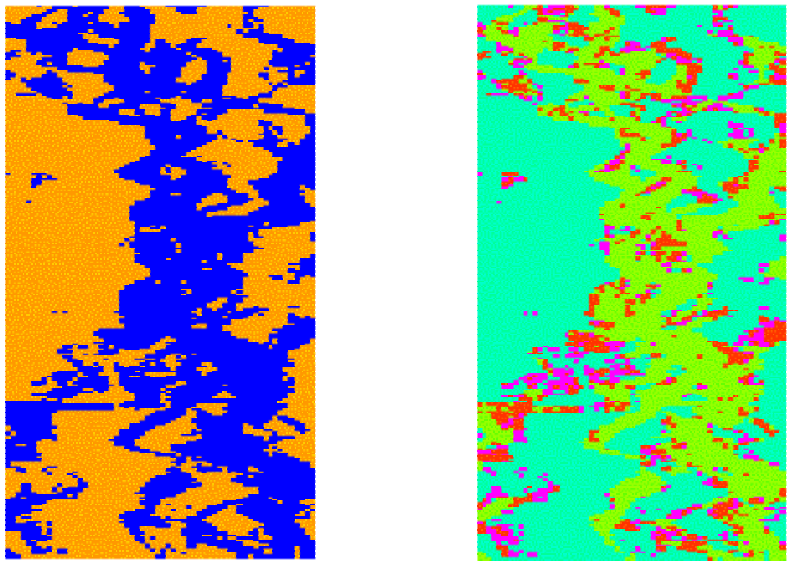


Figure 6: Left image is SP/DP mask for layer 13. Orange indicates matrix only gridblocks while blue indicates matrix/fracture gridblocks. Right image shows relative permeability regions for layer 13.

As the streamline method does not include capillary pressure in the flowing medium transport equations, care was taken to remove capillary pressure from the fracture and the SP matrix relative permeability curves for validation with ECLIPSE. As in the original SPE10 problem, an injector was located in the center of the model, while a producer was located at each corner. Where a well's path intersected DP gridblocks it was completed in the fracture medium of a DP gridblock, otherwise it was completed in the matrix of SP gridblocks. Producers were on total liquid rate control with rates set such that producers in higher permeability regions had a higher liquid rate compared with those in lower permeability regions. The injector was on pressure control. The volume replacement ratio (VRR) was exactly one.

Figure 7 is a comparison of oil rates for each producer for the streamline method and Eclipse. Although the oil production trends between 3DSL and Eclipse are similar for producer 4, 3DSL predicts earlier water breakthrough. We tested different timestep sizes and volume cutoff tolerances yet 3DSL results were unchanged still showing earlier water breakthrough. As shown in Figure 8, water breakthrough is earlier because 3DSL shows a sharper thinner water front in the fracture system than Eclipse does for the layer where breakthrough to producer 4 first occurs. This is typical of highly heterogeneous models where streamlines do a better job of capturing high permeability streaks than FD methods do. The remaining well profiles agree better because there are less continuous streaky high perm zones between the injector and these wells.

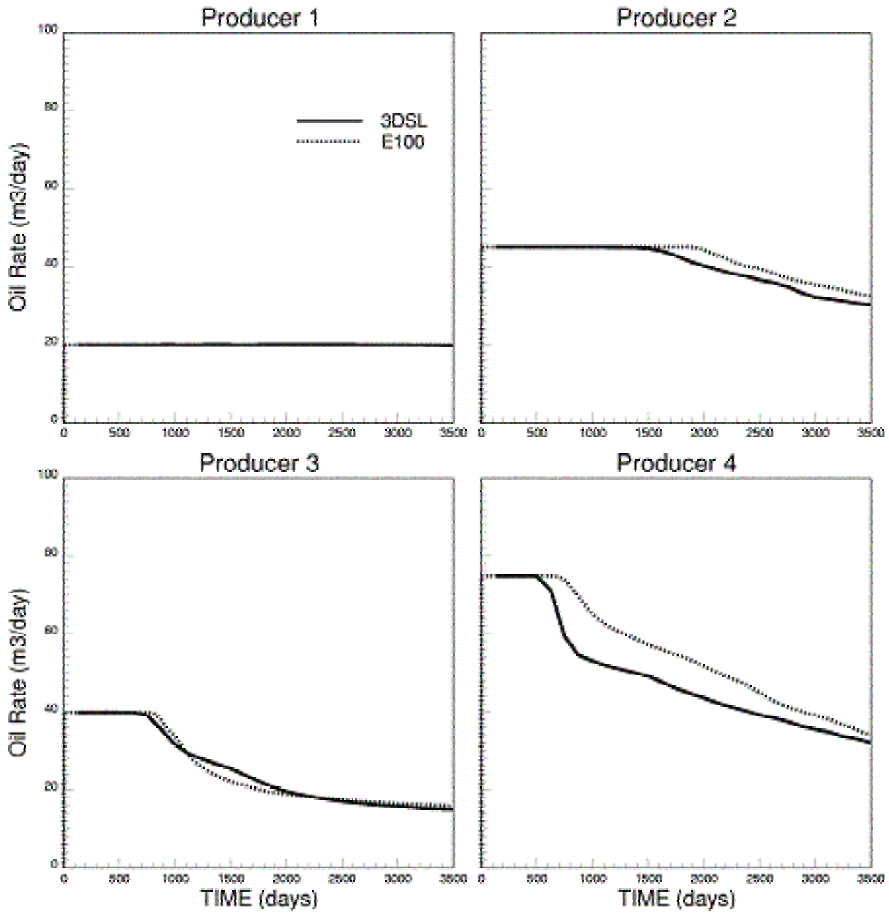
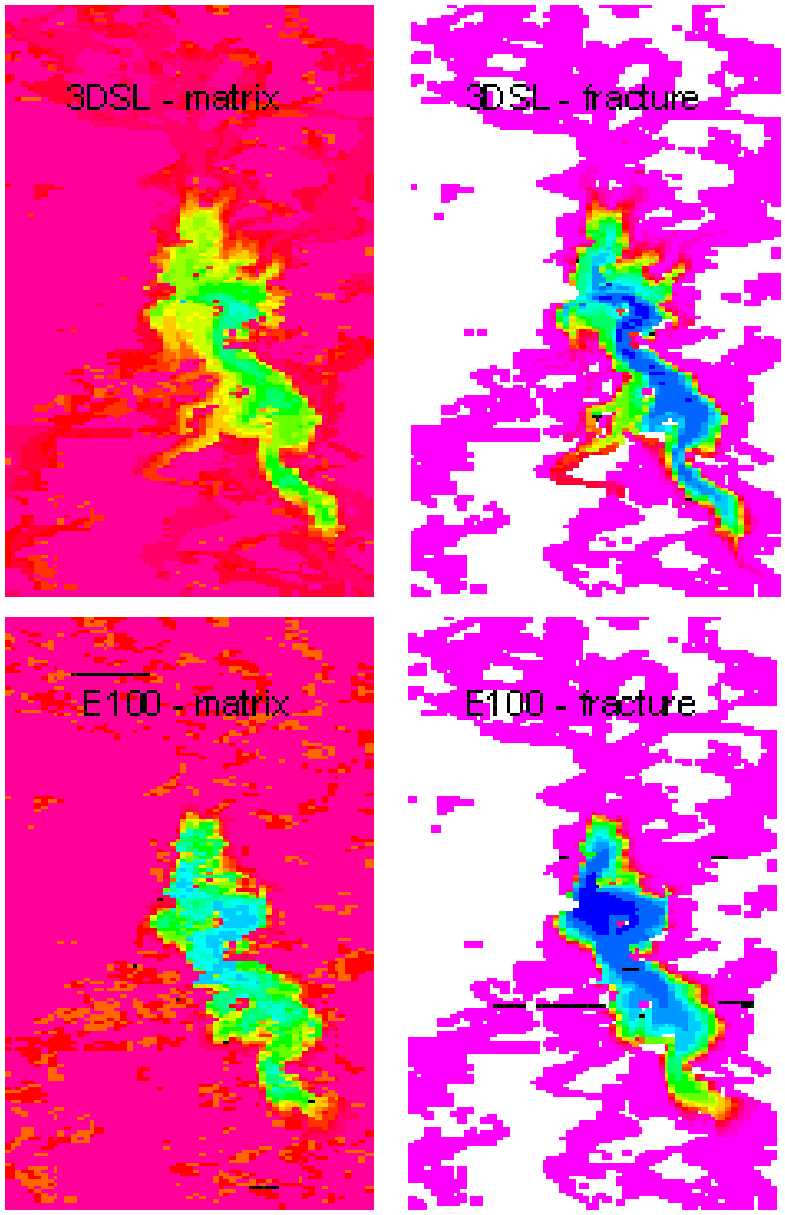


Figure 7: Comparison of oil rate between 3DSL and Eclipse (E100) for each producer for the 60x220x17 SP/DP model.



Both simulation models were run to 3500 days, with the streamline solution requiring about 28 timesteps (243 min on a P4-2.2Ghz Windows PC) and Eclipse requiring about 670 timesteps (1415 min on a Xeon-2.8Ghz Windows PC). Although 3DSL maintains near linear scaling of runtime with model size for both SP and DP systems, this speedup factor is not nearly as high as that quoted by *DiDonato et al.* for a similar sized DP model. The difference in speedup factors between our results and *DiDonato et al.*'s is likely due to the necessary overhead required of a commercial code for widespread applications versus more specialized research code, but also possibly due to the regular discretization along streamlines in *DiDonato et al.* work vs. the irregular discretization used here here.

Figure 8: Water saturation for layer 13 in both the fracture and matrix prior to breakthrough at producer 4 (lower right corner).

Conclusions & Discussion

The dual porosity formulation as outlined by *DiDonato et al.* for streamline-base flow simulation has been extended to more realistic geological models containing both single and dual porosity regions and multiple relative permeability regions. Mixed SP/DP grids give rise to a very large distribution in cell volumes along a 1D profile for each streamline with relatively small cells potentially forcing a 1D explicit solver into unrealistically small time steps. To solve this problem, we propose to merge small volume cells with their neighbors as one way to eliminate the small cells from a streamline's 1D profile. However, because not all small cells can be merged with their neighbors—for example, SP cells cannot be merged with DP cells—the 1D solver method was modified to be adaptive implicit so that any remaining small cells that fall below a set threshold are treated implicitly. Results showed that binning and implicitness have a large and favorable impact on runtimes with little adverse affect on solution profiles.

We concluded with results for a large 280,000 cell model that contained both SP & DP cells as well as multiple relative permeability regions. Streamline results compared well with the grid-based method, but showed more channeling and slightly earlier breakthrough in high permeability zones, most likely due to reduced numerical smearing of the displacement front. Speed-up factors were not as high as those quoted by *DiDonato et al.*, primarily due to the increased overhead in the commercial streamline simulator versus *DiDonato's* research code and regular vs. irregular discretization along streamlines.

While the final test case was realistic in terms of grid size and properties, the primary extension to this work would be applying the streamline method to a typical dual porosity field case. These cases typically have a couple of hundred wells with years of history, yet only a couple of 100,000 cells meaning only a few grid cells between wells. Increasing the number of cells between wells should significantly improve the predictive power of such models, despite some of the simplifying assumptions introduced into the streamline DP-approach. Streamline-based results of such systems on conventional PC's would therefore be expected to be of substantial aid reservoir engineers attempting to understand the dynamic behavior of fractured reservoirs undergoing water injection.

Acknowledgements

The authors would like to thank the members of the ITF project and Imperial College for their financial support. Thanks to Dr. Michael Edwards for his discussions on the AIM method and Dr. Darryl Fenwick for his assistance with the flow simulations.

References

[1] Baker, R.O., Kuppe, S., Chugh, S., Bora, R., Stojanovic, S., and Batycky, R.; "Full-Field Modeling Using Streamline-Based Simulation: Four Case Studies," SPE Reservoir Evaluation & Engineering, April 2002, Vol. 5, No. 2.

[2] Christie, M.A. and Blunt, M.J.: "Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques," SPE Reservoir Evaluation & Engineering, August 2001, 308-317.

[3] Di Donato, G., Huang W., and Blunt, M.: "Streamline-Based Dual Porosity Simulation of Fractured Reservoirs," SPE 84036, proceedings of 2003 SPE ATCE, Denver, Co.

[4] Di Donato, G. and Blunt, M.J.: "Streamline-based dual-porosity simulation of reactive transport and flow in fractured reservoirs," Water Resources Research, **40**, W04203, doi:10.1029/2003WR002772 (2004).

[5] Huang, W., Di Donato , G. and Blunt, M. J.: "Comparison of streamline-based and grid-based dual porosity simulation," Journal of Petroleum Science and Engineering, **43**, 129-137 (2004).

[6] Al-Huthali, A. H. and Datta-Gupta, A.: "Streamline simulation of water injection in naturally fractured reservoirs," SPE 89443, proceedings of the SPE/DOE Symposium on Improved Oil Recovery, Tulsa, OK, 17-21 April (2004).

[7] Forsyth, P.A. and Sammon, P.H.: "Practical Considerations for Adaptive Implicit Methods in Reservoir Simulation," Journal of Computational Physics, February 1986, Vol. 62, No. 2, 265-281.

[8] Gilman J.R. and Kazemi, H.: “Improvement in Simulation of Naturally Fractured Reservoirs,” SPEJ, September 1983, Vol. 23, 695-707.

[9] Grabenstetter, J., Li, Y.-K., Collins, D.A., and Nghiem, L,X.:” Stability-Based Switching Criterion for Adaptive-Implicit Compositional Reservoir Simulation, paper SPE 21225 proceedings of the 11th SPE Symposium on Reservoir Simulation, Anaheim, CA, Feb 17th-20th, 1991.

[10] Hewett, T.A. and Yamada, T.: “Theory of the Semi-Analytical Calculation of Oil Recovery and Effective Permeabilities Using Streamlines,” Adv. Water Res., 1997, Vol 20, No. 5-6, 279.

[11] Huang, W., Di Donato, G., and Blunt, M.J.: “Comparison of streamline-based and grid-based dual porosity simulation,” Journal of Petroleum Science and Engineering, **43**, 129-137 (2004).

[12] Kazemi, H., Gilman, J.R., and Eisharkawy, A.A.: “Analytical and Numerical Solution of Oil Recovery From Fractured Reservoirs With Empirical Transfer Functions,” SPE Reservoir Engineering, May 1992, Vol. 6, 219-227.

[13] Kelly, C.T.: “Solving Nonlinear Equations with Newton’s Method,” SIAM, 2003.

[14] Lolomari, T., Bratvedt, K., Crane, M., and Milliken, W.: “The Use of Streamline Simulation in Reservoir Management: Methodology and Case Studies,” paper SPE63157 in proceedings of the 2000 ATCE, Dallas, TX (October).

[15] Russell, T.F.: “Stability Analysis and Switching Criteria for Adaptive Implicit Methods Based on the CFL Condition,” paper SPE 18416 prepared for the SPE Symposium on Reservoir Simulation, Houston, TX, Feb 6th-8th, 1989.

[16] Samier, P., Quettier, L. and Thiele, M.R.: “Applications of Streamline Simulations to Reservoir Studies,” SPE Reservoir Evaluation & Engineering, August 2002, Vol 5. No. 2.

[17] Schlumberger: ECLIPSE Reference Manual 2003A (ECLIPSE is a Mark of Schlumberger).

[18] Streamsim Technologies, Inc.: “3DSL User Manual V2.2”, June 2004. 3DSL is a registered trademark of Streamsim Technologies, Inc.

[19] Thiele, M.R.: "Streamline Simulation," keynote address presented at the 2003 Forum on Reservoir Simulation, Baden-Baden, Germany.

[20] Thomas, G.W. and Thurnau, D.H.: “Reservoir Simulation Using an Adaptive Implicit Method,” SPEJ, 23 (1983) 759-768.