Oil & Natural Gas Technology

DOE Award No.: DE-FC26-06NT15530

Final Report

Industrial Compositional Streamline Simulation for Efficient and Accurate Prediction of Gas Injection and WAG Processes

Submitted by:
Stanford University
Principal Investigator: Margot G. Gerritsen
Department of Energy Resources Engineering
367 Panama Street
Stanford, CA 94305-2220

Prepared for:
United States Department of Energy
National Energy Technology Laboratory

Award period: October 1, 2004 – October 31, 2008

Office of Fossil Energy

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expresses herein do no necessarily state or reflect those of the United States Government or any agency thereof.

Contact

Margot Gerritsen
Department of Petroleum Engineering
Green Earth Sciences Building
367 Panama Street
Stanford, CA 94305

Phone: (650) 725 2727
Fax: (650) 725 2099
Email: mgerritsen@stanford.edu

Executive Summary

Gas-injection processes are widely and increasingly used for enhanced oil recovery (EOR). In the United States, for example, EOR production by gas injection accounts for approximately 45% of total EOR production and has tripled since 1986. The understanding of the multiphase, multicomponent flow taking place in any displacement process is essential for successful design of gas-injection projects. Due to complex reservoir geometry, reservoir fluid properties and phase behavior, the design of accurate and efficient numerical simulations for the multiphase, multicomponent flow governing these processes is nontrivial. In this work, we developed, implemented and tested a streamline based solver for gas injection processes that is computationally very attractive: as compared to traditional Eulerian solvers in use by industry it computes solutions with a computational speed orders of magnitude higher and a comparable accuracy provided that cross-flow effects do not dominate. We contributed to the development of compositional streamline solvers in three significant ways: improvement of the overall framework allowing improved streamline coverage and partial streamline tracing, amongst others; parallelization of the streamline code, which significantly improves wall clock time; and development of new compositional solvers that can be implemented along streamlines as well as in existing Eulerian codes used by industry.

We designed several novel ideas in the streamline framework. First, we developed an adaptive streamline coverage algorithm. Adding streamlines locally can reduce computational costs by concentrating computational efforts where needed, and reduce mapping errors. Adapting streamline coverage effectively controls mass balance errors that mostly result from the mapping from streamlines to pressure grid. We also introduced the concept of partial streamlines: streamlines that do not necessarily start and/or end at wells. This allows more efficient coverage and avoids the redundant work generally done in the near-well regions. We improved the accuracy of the streamline simulator with a higher order mapping from pressure grid to streamlines that significantly reduces smoothing errors, and a Kriging algorithm is used to map from the streamlines to the background grid. The higher accuracy of the Kriging mapping means that it is not essential for grid blocks to be crossed by one or more streamlines. The higher accuracy comes at the price of increased computational costs, but allows coarser coverage and so does not generally increase the overall costs of the computations. To reduce errors associated with fixing the pressure field between pressure updates, we developed a higher order global time-stepping method that allows the use of larger global time steps. Third-order ENO schemes are suggested to propagate components along streamlines. Both in the two-phase and three-phase experiments these ENO schemes outperform other (higher order) upwind schemes. Application of the third order ENO scheme leads to

overall computational savings because the computational grid used can be coarsened. Grid adaptivity along streamlines is implemented to allow sharp but efficient resolution of solution fronts at reduced computational costs when displacement fronts are sufficiently separated. A correction for Volume Change On Mixing (VCOM) is implemented that is very effective at handling this effect. Finally, a specialized gravity operator splitting method is proposed for use in compositional streamline methods that gives an effective correction of gravity segregation.

A significant part of our effort went into the development of a parallelization strategy for streamline solvers on the next generation shared memory machines. We found in this work that the built-in dynamic scheduling strategies of OpenMP lead to parallel efficiencies that are comparable to optimal schedules obtained with customized explicit load balancing strategies as long as the ratio of number of streamlines to number of threads is sufficiently high, which is the case in real-field applications. This is an important result, as it eases the transition of serial to parallel streamline codes. The parallel speedup itself depends on the relative contribution of the tracing and mapping stages as compared to the solution of the transport equations along streamlines. As the physical complexity of the simulated 1D transport process increases, the contribution of the less efficient tracing and mapping stages is reduced and near-linear scalabilities can be obtained. Our work clearly shows that the owner approach, in which threads are assigned whole streamlines, is more attractive than a distributed model, in which streamline segments are assigned to threads, because it allows re-use of existing sequential code for the 1D streamline solves, also for implicit time-stepping algorithms. Any load balance advantage of the distributed model is negligible for real-field applications where the streamline to thread ratio is generally high. Initial concerns about data locality and cash misses never materialized in our work, and little or no performance degradation could be attributed to data locality. Parallel speedup on the new generation Core Multi Processor (CMP) architectures is particularly high: we achieved linear speedup in all tests. We attribute this also to the particularly efficient shared cache design of modern CMP architectures. This work confirms what many researchers have assumed about streamline simulation: that it is trivially parallelizable on shared-memory machines and that near-linear scalability can be achieved. Our parallel strategy has now been implemented in a commercial streamline solver, showing the immediate benefits our work had to industry.

In the third part of this project, we developed two new types of transport solvers specifically for gas injection processes: relaxation methods and iso-diffusive multi-D schemes. The relaxation schemes can be applied to the one-dimensional streamline transport equations. The multi-D schemes are specifically designed for those stages in the gas injection process where Eulerian methods are desirable. For example, in the early stages of injection, gravity segregation is very strong and streamline methods are not capable

of predicting the process with sufficient accuracy. A Eulerian approach can then be used to such time that gravity segregation is more or less established and streamline methods can take over.

The constant sub-characteristic Jin-Xin relaxation scheme, which has been in use for around a decade in the applied mathematics community, is attractive for two-phase multicomponent systems in that it removes the dependency of the numerical solver on the eigenstructure of the system and nonlinear Riemann solutions. But, the necessary restriction on the sub-characteristic speed results in excessive numerical diffusion that can significantly reduce solution accuracy in strongly nonlinear compositional problems. We developed a variant of this scheme, called the constant $\alpha-\beta$ method, which however does not offer sufficient improvement and behaves like the Jin-Xin scheme for eigenvalues with equal magnitude, or the corresponding upwind schemes when small negative eigenvalues are present. We were successful with our new variable extension of Jin-Xin relaxation. This version maintains all the advantages of the standard Jin-Xin scheme, while reducing numerical diffusion considerably by locally imposing the sub-characteristic condition. For ternary gas-oil displacement system, our extensive testing showed that in many cases the second order variable relaxation scheme is competitive with component-wise ENO-RK3 and that it is always as least as good as component-wise TVD-RK2.

To allow hybrid Eulerian-streamline approaches, we developed truly multi-D transport schemes. We used modified equations analysis for determining preferential flow angles for numerical methods on structured grids, which were then used by us to reduce the grid-biasing found in traditional schemes. We developed a general framework for multi-D schemes with local positivity constraints using interaction regions. We showed that grid orientation effects and numerical biasing can be reduced significantly through the use of truly multi-D schemes and the introduction of numerical diffusion corrections. This was demonstrated for the new Flat scheme, which has near constant transverse diffusion, and gave favorable results on both homogeneous and heterogeneous displacements.

Table of Contents

Compositional Streamline Simulation

New compositional solvers: relaxation schemes

New compositional solvers: truly multi-dimensional schemes

Parallel Streamline Simulation

Project conclusions and specific outcomes

1. Introduction

1.1 Challenges in simulating gas injection for Enhanced Oil and Gas Recovery

Gas-injection processes are widely used for enhanced oil recovery (EOR) throughout of the world. In the United States alone, EOR production by gas injection now accounts for approximately 45% of total EOR production and has been steadily increasing (tripled since 1986). The understanding of the multiphase, multicomponent flow taking place in any displacement process is essential for successful design of gas-injection projects. Due to complex reservoir geometry and reservoir fluid properties, the design of accurate and efficient numerical simulations of the flow processes is extremely challenging.

When gas displaces oil at a sufficiently high pressure, the local displacement efficiency can be high. Miscibility is said to develop when optimal, piston-like local displacement efficiency is achieved. This is often possible when CO_2 is used as injection gas. Due to reservoir heterogeneity and low gas viscosity, however, the injected gas may contact only a small portion of the reservoir as it finds the high permeability flow paths. That is, the global sweep efficiency of a gas flood may not be high. The process performance of gas injection schemes depends on this balance between local displacement efficiency and global sweep efficiency, and both need to be captured accurately by a performance prediction tool.

This presents three substantial challenges for reservoir simulators, particularly when a compositional model is used to describe the reservoir fluids.

- Because the global sweep depends foremost on the underlying heterogeneity and gravity, realistic high-resolution reservoir models are required for accurate predictions. Upscaling to coarse simulation grids may result in loss of the important permeability contrasts, which in turn may result in inaccurate predictions of breakthrough times or global sweep efficiency. A compositional fluid description leads to high computational cost per grid cell due to the potential large number of unknowns in the system and the added expense of phase equilibrium calculations.
- In modeling the complicated multi-phase multi-component flows it is paramount to have a good

physical understanding of the interactions between components in the flowing phases. The number and type of (pseudo) components used strongly affect the accuracy of the simulation. Moreover, physical models, such as relative permeabilities, can have a large effect on the simulation model results. This is especially true in three-phase flows, such as in WAG processes or CO_2 sequestration in depleted reservoirs, where predicted recovery can range widely depending on the selection of relative permeability models alone (Guzman, 1994).

- Phase behavior and relative permeability functions introduce a strong nonlinear coupling between the component transport equations. It also introduces a strong nonlinear coupling between the flow and transport equations due to the total mobility, compressibility and volume change on mixing. Consequently, simulations are very sensitive to modeling errors and numerical errors, such as numerical diffusion. Cost-effective but strongly diffusive numerical methods, such as the first order upwind schemes commonly used in simulators, are unsuitable for these compositional problems, as they tend to exaggerate the global sweep and reduce the local displacement efficiency (see, e.g., Jessen *et al.* (2002)).

1.2 Existing solvers

For the reasons outlined above, compositional simulation has been the subject of research for many years, but a unified compositional solution framework does not yet exist. Most methods for compositional simulation are based on Eulerian grids (Aziz and Settari (1979)). The well-known IMPEC (IMplicit Pressure, Explicit Composition) methods, where the pressure equation (also referred to as the flow equation) is solved implicitly and the component concentrations are obtained through an explicit solve of the transport equations, are limited by severe global stability restrictions on the time step size. On the other hand, the stable fully implicit methods (FIM) are computationally very expensive and therefore limited foremost by the number of unknowns. They may also introduce more numerical diffusion that, because of the strong nonlinearities present, may seriously affect performance prediction. Required variable substitution because of phase changes introduces further complications. Specialized solvers are required, see, e.g., Lacroiz *et al.* (2003). An intermediate approach is given by the IMPSAT methods where the pressure and saturation equations are solved implicitly and the remaining equations explicitly. Recent work on IMPSAT methods is provided in Haukås *et al*. (2004), and Cao (2002). Another alternative is to use an Adaptive implicit method (AIM), formulated to reduce the number of implicit unknowns required by FIM and to alleviate the time step restrictions associated with moving compositions explicitly throughout an entire reservoir (Thomas and Thurnau (1983)). The computation time required for simulation can of course also be reduced through parallelization (Abate *et al.* (2001) and Wang *et al.* (1999)), or by including adaptive mesh refinement (AMR) as in Sammon (2003). AMR focuses computational effort in regions near displacement fronts to accurately capture the local displacement efficiency. Although much progress has been made in recent years, conventional simulators that are based on finite difference or finite element discretization of the three-dimensional governing system of equations can generally not provide solutions for realistic reservoirs in realistic time frames, even after code parallelization and addition of AMR capability.

As an alternative to the Eulerian methods mentioned above, Euler-Lagrange type methods can be used, see, for example Wang *et al.* (2005). The Euler-Lagrange localized adjoint methods (ELLAM) take advantage of the stability of Lagrangian methods over long time steps and do not suffer from the mass conservation errors often observed with Lagrangian methods (Celia *et al.* (1990), Russell and Celia (2002)). Several investigations are being conducted to ascertain the applicability of ELLAM methods to displacement problems that have strong sensitivities to phase behavior. The streamline methods are another class of Euler-Lagrange type methods based on the physical observation that in heterogeneous reservoirs the time scale at which fluids flow along streamlines is often much faster than the time scale at which the streamline locations change significantly. This allows decoupling of the transport problem into a sum of 1D problems along streamlines. Previous studies have shown that streamline methods can predict the global sweep of water floods in heterogeneous reservoirs effectively (see, for example, Thiele and Batycky (2006)). A preliminary extension of streamline methods to compositional simulation was discussed in Thiele *et al*. (1997). Commercial streamline simulators are currently being extended to handle compositional processes, but are not yet at the stage where they can reliably predict performance of (near-) miscible gas injection processes.

1.3 Motivation for using streamline simulation

Streamline simulation has rather attractive properties. When using streamlines, the costly three-dimensional equations governing transport of compositions, discussed in the next chapter, are decoupled into a set of one-dimensional advection problems along the streamlines. After gravity segregation has been established, gas injection processes are generally strongly advection-dominated, which makes the use of flow based grids for transport, such as streamlines, natural and attractive. Furthermore, flow paths, and therefore streamline trajectories, are likely controlled by reservoir heterogeneity because of the high mobility ratios in gas injection processes. So, it can be expected that streamlines do not move significantly in time periods between well updates.

De-coupling the three-dimensional transport equations into a set of one-dimensional solves has several computational advantages. First, the decoupled transport systems are naturally parallelizable. Since they contain the bulk of the computational work (due to the computationally intense phase equilibrium calculations), streamline simulation is especially attractive. Second, AMR is easily implemented on the one-dimensional streamline grids, which further reduces computational costs. Third, in screening, optimization or history matching studies, the one-dimensional transport equations along streamlines may be solved approximately using the Method of Characteristics, which provides semi-analytical solutions for constant initial and injection states (Jessen and Orr (2002)). This leads to highly efficient proxies. When finite difference methods must be used to solve the one-dimensional transport equations, the de-coupling of the transport problems allows each streamline solve to have its own time step. Thus, severe stability constraints that may lead to very small time steps in one part of the computational domain, may not affect time steps away from this area. Finally, in the decoupled transport system, numerical diffusion introduced by the transport solvers is limited to the longitudinal direction, and can be effectively controlled using higher order methods. The gains in speed allows for a higher grid density to be used on both pressure and streamline grids so that important heterogeneity features are better represented, and fronts are resolved more sharply. Also, an adequate number of components can be used to increase the accuracy of the performance prediction.

General shortcomings of streamline methods include negligence of cross flow between pressure updates which must be corrected for, potentially large impacts of smoothing and mass balance errors introduced by the mappings between streamlines and pressure grid, and the challenge of generating economic streamline coverage. Most of these shortcomings of streamline methods are well-known, and solution strategies have been proposed. For example, operator splitting techniques have been developed to account for gravity as in Bratvedt *et al.* (1996), and for capillary flow as in Berenblyum *et al.* (2003) and high streamline densities can in cases alleviate mapping errors. In compositional simulation, however, numerical solutions are much more sensitive to errors because of the strong nonlinear couplings introduced by the phase behavior. Therefore, greater care must be taken when mapping between the pressure grid and streamlines, and when solving the transport equations along streamlines. Also, gas injection processes require accurate representation of gravity effects. Attention must be paid to the quality of the pressure grid and upscaling strategies on this grid as important high permeability flow paths should be represented well.

1.4 Main project achievements

The main achievements of this project are in the following three main areas:

1. Development of specialized tools to allow the extension of streamline simulators to gas injection processes, discussed in chapter 2 of this final report.

   These tools include improved mappings from streamlines to pressure grid and vice versa to reduce the undesirable smoothing and the potentially large mass balance errors that may be introduced by traditional mappings, an adaptive streamline coverage algorithm that adds streamlines to a (coarse) base streamline grid in regions where coarser coverage may lead to large mapping errors, a higher order pressure solve, and a new operator splitting method for gravity. Along streamlines we use higher order upwind schemes to reduce numerical diffusion, which was demonstrated to lead to incorrect predictions of front speeds and strong under-estimation of displacement efficiency in one of our previous studies (Mallison *et al.* (2005a)). We extended this study to three-phase flows and included an adaptive meshing strategy along streamlines that is used to further reduce computational expense.

2. Development of specialized transport and pressure solvers for multi-phase, multi-component systems, discussed in chapters 3 and 4.

   To alleviate some of the problems encountered with using traditional upwind methods, we formulated a new relaxation approach for compositional problems. This is discussed in detail in chapter 3. Relaxation methods can be applied on multi-dimensional Eulerian grids, as well as alongs one-dimensional streamlines.

   We contributed not only in the design of methods for the one-dimensional transport solves needed in streamline simulation, but also in the design of general transport solvers for compositional simulation. This is motivated by the fact that in industrial applications it may be necessary to resort to a hybrid method: when cross flow is strong (for example at the start of a gas injection process when gravity segregation is critical), Eulerian methods may be used and whenever appropriate a switch can be made to streamline simulation, and vice versa. We focused specifically on the design of truly multi-dimensional positive transport solvers to reduce the strong sensitivity felt to the computational grid, referred to as grid orientation effects, inherent to adverse mobility ratio flows. Chapter 4 describes our work in this area.

3. Parallelization of compositional streamline solvers and inclusion of parallel strategies in commercial streamline simulators. The parallel strategies are outlined in chapter 5.

We developed a parallelization strategy for streamline simulators on shared memory systems with particular emphasis on the next generation of shared memory machines: the core multi-processors (CMP). We conducted this part of the project in very close collaboration with a commercial streamline simulator developer who has taken the work and developed its own commercial version. This is a direct impact that our project has had on industrial codes.

Regarding the contents of this final report, there are two important notes:

- The development of the adaptive pressure solve and associated specialized transmissibility upscaling method, which is an integral part of the compositional streamline framework, is discussed extensively in the final report of grant DE-FC26-03NT15405. The numerical development work of this project overlapped with the 15405 project in this area.
- Experimental work on multiphase flow, with a special emphasis on three-phase relative permeability has been performed in collaboration with Prof. Sally Benson. This experimental program was sponsored at Stanford by the Global Climate and Energy Project (GCEP). Funds from this grant were therefore not needed.

2 Compositional Streamline Simulation

2.1 Governing equations

We consider a gas injection process containing N_p phases and N_c components. Mass balance for the N_c components present in the system is expressed through the system of transport, or mass balance, equations in the form

$$\phi \frac{\partial C_i}{\partial t} + \nabla \cdot \underline{U}_i = q_i, \qquad i = 1,\ldots,N_c. \tag{1}$$

Here, ϕ is the porosity of the porous medium, t is time, q_i is the molar flow rate of component i due to wells, and N_c is the number of components. The composition (moles per unit volume) of a mixture is given by

$$C_i = \sum\nolimits_{j=1}^{N_p} x_{ij} \rho_j S_{j,} \quad i = 1,\ldots,N_c, \tag{2}$$

where N_p is the number of phases, x_{ij} is the mole fraction of component i in phase j, ρ_j is the phase molar density and S_j the volumetric phase saturation. Similarly, the overall molar flux is given by

$$\underline{U}_i = \sum\nolimits_{j=1}^{N_p} x_{ij} \rho_j \underline{u}_j, \quad i = 1,\ldots,N_c. \tag{3}$$

Phase mole fractions and densities are determined by phase equilibrium (flash) calculations using the approach of Michelsen (1998) and the Peng-Robinson equation of state (Peng and Robinson (1976)). The phase velocities u_j are given by the standard multiphase extension of Darcy's law,

$$\underline{u}_j = -\mathbf{k}\lambda_j (\nabla p + \rho_{mj} \underline{g}), \quad j = 1,\ldots,N_p, \tag{4}$$

with $\mathbf{k}$ the permeability tensor of the porous medium and ρ_{mj} the phase mass density. The phase mobilities are given by $\lambda_j = k_{rj}/\mu_j$ where μ_j are phase viscosities determined by the Lorenz-Bray-Clark correlation (Lohrenz *et al.* (1964)). For two-phase flows, we use Corey type relative permeability curves for k_{rj}. For three-phase flows the standard methods of Stone (I and II) and

Baker are implemented. Assuming incompressibility of the rock, a balance on the total fluid volume, V_t, leads to the pressure equation

$$c_t \frac{\partial p}{\partial t} + \sum\nolimits_{i=1}^{N_c} \frac{\partial V_t}{\partial n_i} \nabla \cdot \underline{U}_i = \sum\nolimits_{i=1}^{N_c} \frac{\partial V_t}{\partial n_i} q_i. \tag{5}$$

Here n_i are the mole numbers for each component and c_t is the total fluid compressibility.

In streamline simulation, the pressure, and therefore the flow through Darcy's law, is calculated on a three-dimensional Eulerian grid. But, rather than moving compositions between cells on the pressure grid, the streamline method constructs a Lagrangian grid at each global time step in order to solve the transport equations (1). The streamlines making up this grid follow the total velocity $\underline{u}_t$, given by

$$\underline{u}_t = \sum\nolimits_{j=1}^{N_p} \underline{u}_j. \tag{6}$$

In the absence of gravity, the phase velocities are aligned with the total velocity. However, with gravity there may be a discrepancy between the directions of the phase velocities due to density differences. In this case, the overall molar flux (3) can be split into a component in the direction of the total velocity field and a vertical component that accounts for gravity segregation as

$$\underline{U}_i = \underline{u}_t F_i + \underline{g} G_i. \tag{7}$$

The overall fractional flow of each component is given by

$$F_i = \frac{1}{\lambda_t} \sum\nolimits_{j=1}^{N_p} x_{ij} \rho_j \lambda_j \,, \quad i = 1, ..., N_c. \tag{8}$$

The total fluid mobility λ_t is defined as the sum of the phase mobilities. G_i represents the overall gravity fractional flow of component *i*. If we assume that the flow is incompressible then, ignoring source terms, the splitting given in (7) can be used to express the transport equation (1) as

$$\varphi \frac{\partial C_i}{\partial t} + |\underline{u}_t| \frac{\partial (u_d F_i)}{\partial \xi} + |\underline{g}| \frac{\partial G_i}{\partial z} = 0, \qquad i = 1, ..., N_c. \tag{9}$$

Here ξ is arc length measured along a streamline, u_d is the dimensionless velocity, which is the velocity scaled by the injection velocity, and $z = z(\xi)$ is depth. Operator splitting is used to solve (9) by first moving fluids along streamlines and then segregating the phases according to density differences. The introduction of the time-of-flight τ and the operator identity

$$\varphi \frac{\partial}{\partial \tau} = |\underline{u}_t| \frac{\partial}{\partial \xi}, \tag{10}$$

leads to a dimensionless expression

$$\frac{\partial C_i}{\partial t} + \frac{\partial (u_d F_i)}{\partial \tau} = 0, \qquad i = 1, ..., N_c, \tag{11}$$

for flow along streamlines, which are everywhere tangential to the total velocity field. Equations (11) are solved along streamlines in most streamline methods along with the flash calculations. The equations can also be expressed as a function of arc-length rather than time-of-flight. The overall performance of the streamline method can depend on this coordinate choice. The time of flight formulation has the advantage that the resulting 1D transport equations are dimensionless as given in (11). The disadvantage is that the spacing of the cells along the streamlines in the physical space becomes distorted so that fewer cells are located in high flow regions (e.g. near wells). The arc length formulation gives better control over the density of the streamline grid and hence generally leads to smaller mapping errors.

In the above, we assumed the flow was incompressible. Streamline equations can also be formulated for the compressible case, as outlined in Cheng *et al.* (2005). Compressibility effects generate source and sink terms along the streamlines which account for fluid expansion and compression, respectively. In compositional simulation of depletion processes, as well as at early times in compositional gas injection processes, compressibility effects are expected to be important. However, we believe that during displacement the impact of these terms is relatively weak as compared to the effects of volume change on mixing (VCOM) in many gas injection processes. Therefore we have focused on the latter in this work, and introduced a VCOM correction along streamlines.

Apart from the full compositional equations, we use a simplified miscible flow model to help demonstrate our new methods. In this model, the flow equation is given by

$$\nabla \cdot (k\lambda \nabla p) = q_t, \tag{12}$$

and the transport equation by

$$\phi \frac{\partial S}{\partial t} + u \cdot \nabla S = 0, \tag{13}$$

where S is the volumetric saturation of the injected fluid. We take the viscosity of the fluid to depend on the saturation of the injected fluid, so $\mu = \mu(S)$, with the end-point mobility ratio $M = \frac{\mu(1)}{\mu(0)}$. For saturations $0 < S < 1$, we compute the mobility using a quarter-power mixing rule as $\lambda(S) = \left[(1-S) + M^{1/4}S\right]^4$.

2.2 The general framework

The general streamline simulation process consists of the 5-step cycle, illustrated in Figure 1, which is repeated until the final simulation time is reached. The steps are as follows:

1. Given an initial solution on the pressure grid, a pressure solution is computed (using explicit coefficients).
2. Using Darcy's law, given by equation (4), the total velocity field is computed from the pressure field.

3. Streamlines are traced, streamline grids created, and compositions are mapped from the pressure grid onto the streamlines. Streamline tracing can be done with the well-known Pollock's tracing method (Pollock (1988)), or higher order streamline tracing as outlined in, for example, Matringe & Gerritsen (2004).
4. The transport equations (11) are solved along the streamlines using an appropriate numerical method for some time until the next pressure update. During this time, the pressure field, and hence the velocity field and streamlines are held fixed.
5. When necessary, the new compositions are mapped back to the pressure grid. Gravity cross flow is accounted for by an operator splitting method, and the process is restarted from step 1.

The streamline method is similar to IMPEC, but here, the transport equations are solved on the specialized transport grid formed by the streamlines. Also, there are two distinct time step sizes in this process. The first is the time step between pressure updates (the global time step) and the second the time-step used in the numerical transport solve along streamlines to move compositions between pressure updates (the local time step). The local time steps are determined by stability criteria when explicit schemes are used, and by the desired temporal accuracy in case of implicit methods. The time steps may vary from streamline to streamline, which is especially attractive in explicit methods where the stability restrictions can be locally severe. Appropriate selection of the global time-step is non-trivial.

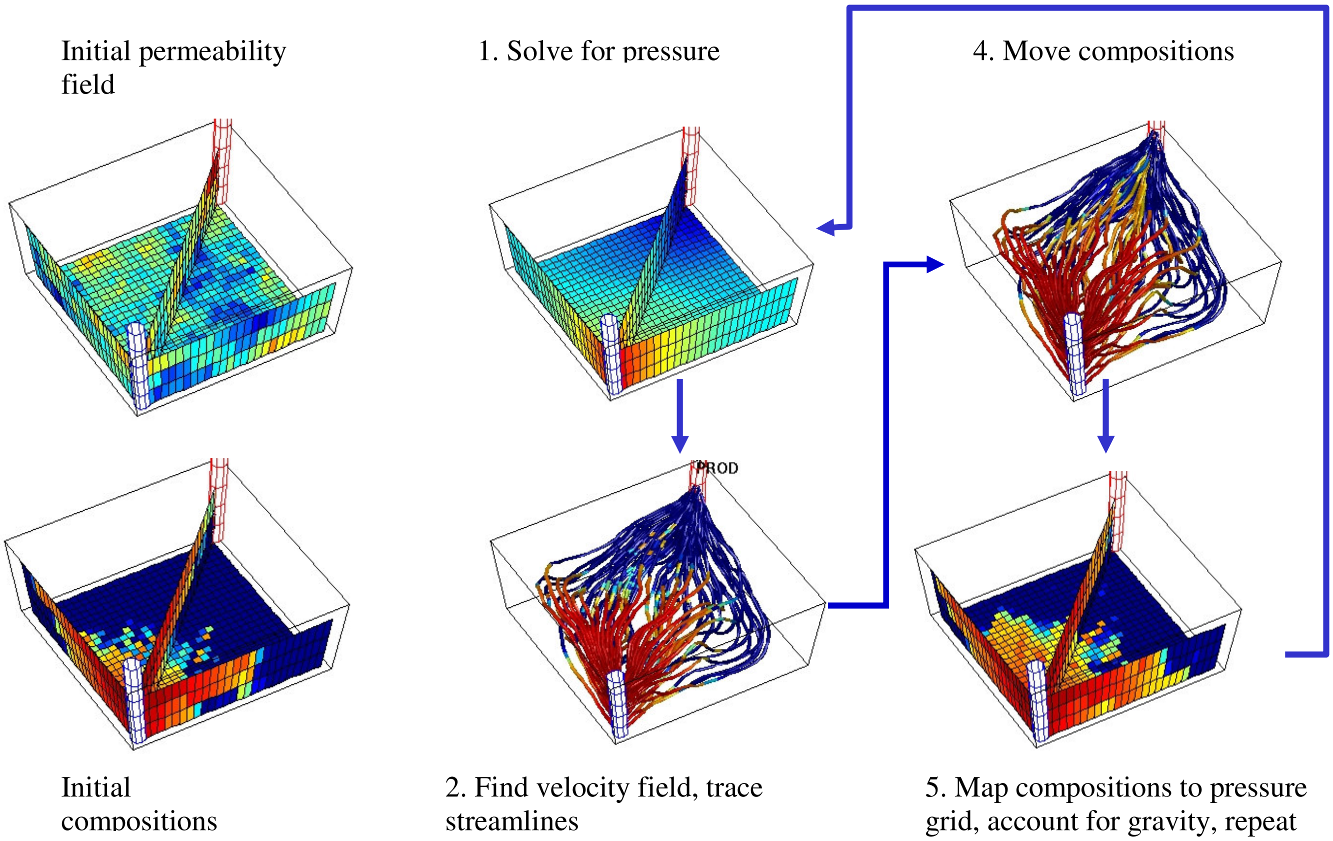

Initial permeability field
1. Solve for pressure
4. Move compositions
PROD
Initial compositions
2. Find velocity field, trace streamlines
5. Map compositions to pressure grid, account for gravity, repeat

Figure 1. The steps in a streamline simulation process

2.3 An adaptive streamline framework

In the traditional streamline method, each streamline is seen as a fluid carrier between global pressure updates: it is assigned a volumetric flow rate when it leaves the injection well, and carries this with it until it reaches the producer. At the production ends, total production is computed by adding the various rates. At pressure updates, solutions values are mapped from streamlines to pressure grid blocks by weighting the contributions from the set of streamlines crossing the block based on their respective volumetric rates. Although this approach simplifies to some extent handling at the production end, it has as disadvantages that each pressure grid block must be crossed by at least one streamline for the solution to be known in that block, and that all streamlines must be traced from source to sink. This leads to excessive streamline densities near wells or other flow converging areas. We also remark that although the number of streamlines needed to cover a domain this way increases linearly with the number of grid cells N, the number of streamline segments crossing a pressure grid block, which determines the workload in a streamline solve unless grid coarsening is applied, also increases with N. Hence, for the streamline method to scale favorably, the number of streamlines should increase only sublinearly with N and/or grid adaptation strategies should be applied along streamlines to reduce the number of grid points per streamline.

In our streamline method, we do not assign volumetric flow rates to streamlines. Equation (11) can be solved along each streamline without this assumption. We compute solution values in each grid point along the streamlines that can be interpolated to find values in other points in the domain. In our case, we use Kriging interpolation, which is highly suitable for unstructured data, to map solutions from the streamlines to the pressure grid. It is not necessary in this approach for each grid block to be insersected by a streamline. Also, the mapping is now independent of the volumetric flow rate traditionally associated with each streamline. This approach is not new in that it is the general approach in Euler-Lagrange methods. However, it has not been used in streamline solvers before. The other advantage of this rate-free approach is that it allows us to trace partial streamlines to locally increase streamline density: streamlines do not have to start and end at wells or domain boundaries, but can be added anywhere in the domain, as discussed below.

We suggest an iterative coverage control algorithm separated in four main steps:

1. We create a relatively coarse streamline grid, in which streamlines are traced from well to well. If desired, volumetric rates could be assigned to this coarse set to aid production calculations. At the current time, t^n, compositions are mapped on this coarse set. We move compositions along the streamlines from t^n to t^{n+1}.
2. We locate areas where the scattered point density resulting from the coarse streamline grid leads to insufficient accuracy in the solutions at t^{n+1} according to some measure (such as the one discussed below).
3. We trace partial streamline segments downstream from a number of selected launch points in the high error regions until they pass into regions where a predefined accuracy threshold has been reached. A segment must also be traced some distance on the upstream end to ensure a proper handeling of the boundary condition for the streamline at this end. The upstream segment is not used when mapping new compositions back to the pressure grid.
4. We map the compositions at time t^n onto these new streamline segments, and move the compositions forward in time until t^{n+1}. The upstream boundary condition of the partial streamline segment is picked up from surrounding streamlines by spatial and, if necessary, temporal interpolation.

   Steps 2 – 4 are repeated until we are satisfied with the solution quality everywhere.

   Adaptive mesh refinement along the streamlines helps retain sharp gradients and reduce smoothing.

An indicator is needed to determine the optimal number and launching locations of the streamlines. One criterion that can be used for streamline density is that the streamline-to-pressure grid mapping is sufficiently accurate. For that reason, we can, for example, use an indicator based on the Kriging variance given by

$$\sigma_K^2 = E\{[Z(\underline{x}) - Z^*(\underline{x})]^2\} = Var\{Z(\underline{x}) - Z^*(\underline{x})\} \quad . \tag{14}$$

Here *Z ($\underline{x}$)* is the true value of a property, such as saturation or compositions, at the location $\underline{\text{x}}$. $\text{Z}^*(\underline{\text{x}})$ is the Kriging estimate of this property given by the weighted average

$$Z^*(\underline{x}) = \sum_{\alpha=1}^{N_\alpha} \lambda_\alpha Z(\underline{x}_\alpha) \,, \tag{15}$$

where λ_α are the Kriging weights and $Z(\underline{x}_\alpha)$ are the N_α known data points located on the streamlines in the search neighborhood. Information on actual data values could be included by combining this Kriging variance with the data variance as suggested in Matringe *et al.* (2005).

The adaptive streamline method is illustrated using the miscible test problem for M=10. The permeability field is shown in Figure 2, together with a sample of streamlines. It was taken from layer 5 of the SPE Tenth Comparative Project described in Christie and Blunt (2001). Injection takes place at the left boundary with a fixed inflow rate. We produce from the right boundary, which is kept at fixed pressure. No-flow boundary conditions are used at the top and bottom boundaries. Figure 3 shows the adapted streamline grid and saturation after 10 global time steps. The computed mass balance errors are 7.9%, 1.4% and 0.2% with sparse, medium and dense coverage, respectively. Clearly, mass balance errors can be controlled with streamline density. Partial streamline segments allow this control for reduced costs. Adaptive mesh refinement along the streamlines and the adaptivity in the streamline coverage help retain sharp gradients and reduce smoothing.

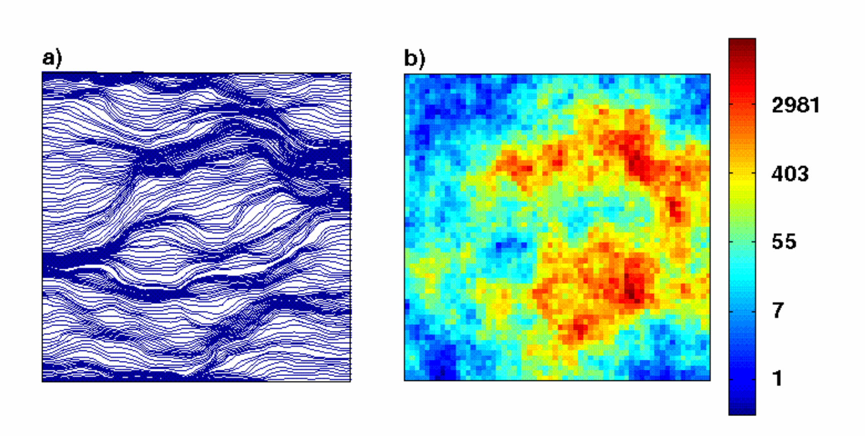


Figure 2. (a): Streamlines; (b) Permeability field used for simulation shown in Figure 3.

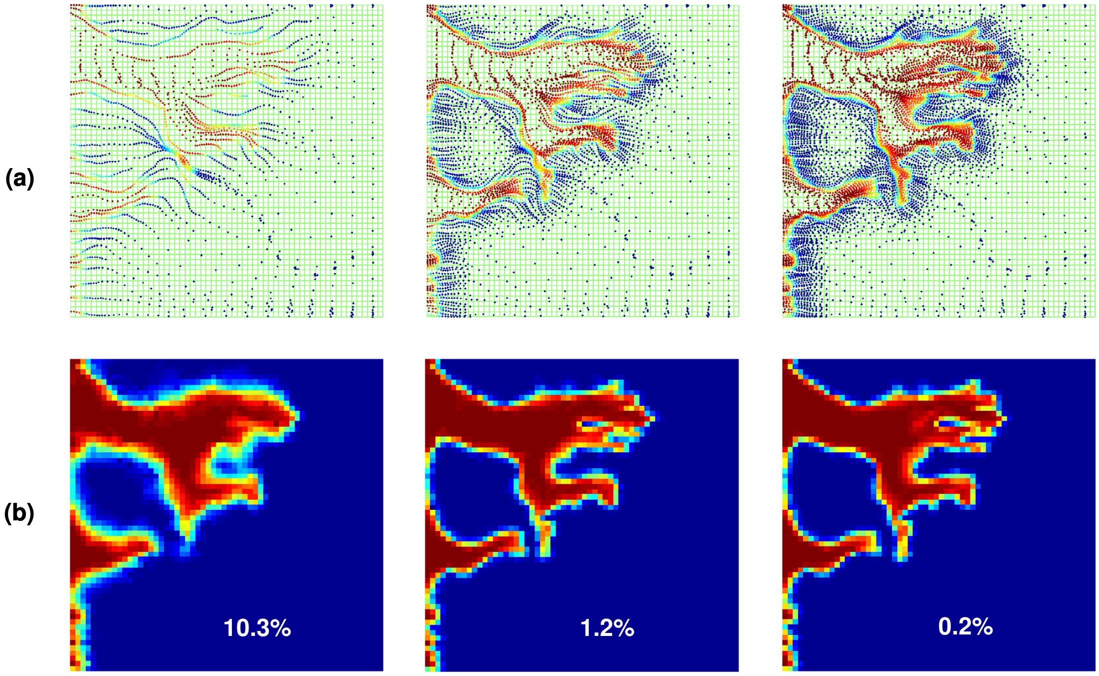


Figure 3. Illustration of adaptive streamlines. As streamline segments are added (a), numerical smearing and mass balance errors are reduced (b). Adaptive Mesh Refinement is used along streamlines. The mass balance errors are given for each scenario in the bottom figures.

2.4 Mappings to and from streamlines

When a streamline is traced across the pressure grid the intersections of the streamline path with the cell faces of the background grid form an irregular 1D grid that can be parameterized in terms of time of flight or arc length. For the purposes of mapping and calculating arc length (but not tracing), we utilize a linear approximation of the streamline path within each cell of the background grid. Our experiments suggest that the error introduced by this approximation is not a leading source of error in the overall method. Compositions must be mapped from the pressure grid to this 1D grid in order to initialize a 1D solve. In addition, this irregular grid is remapped onto a regular 1D grid in order to avoid small cells formed when streamlines cut through corners of grid cells and to simplify the application of adaptive mesh refinement. In our examples we choose the number of cells in the regular 1D grid to be twice the number of irregular cells, although this parameter is adjustable. The methods presented in this section and the following section can be applied to either the arc-length or time-of-flight formulation.

The commonly used mapping routine from pressure grid to streamlines assumes that compositions (or saturations) are piecewise constant on cells of the background grid. These constant values are assigned to the irregular cells as shown in Figure 4a, and then conservatively averaged over cells on the regular 1D grid. This same piecewise constant assumption is also made, for example, by first order Eulerian discretizations. It typically results in strong smoothing of the solutions because of numerical diffusion (Mallison *et al.* (2005b)), when the number of global time steps is large. The smoothing artificially stabilizes fronts and so leads to an exaggerated global sweep prediction. It affects the predicted local displacement efficiency as well because of the strong sensitivity of the transport solves to numerical errors.

A piecewise linear representation of compositions on the background grid helps to minimize numerical diffusion, and in particular transverse diffusion. To avoid oscillations and overshoots, slope limiting is used, as is common with total variation diminishing (TVD) finite difference schemes. We compute slopes independently for each component and each coordinate direction. After assigning compositions to the irregular 1D grid, a second linear reconstruction is employed to conservatively average the compositions over cells the regular 1D grid. This mapping is second order accurate in smooth regions away from extrema and reduces the smearing of sharp fronts. Figure 4b gives an illustration. For the tests presented in subsequent sections we have used the minmod limiter to minimize overshoots and oscillations although other limiters can be used. An overview is given in Van Leer (1979).

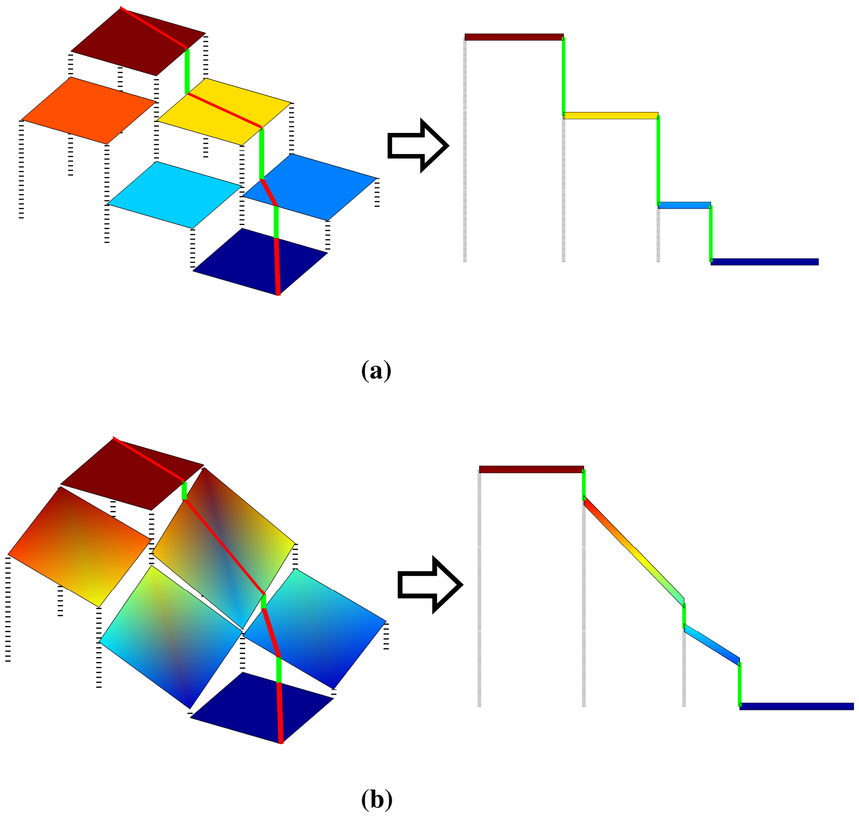
Within most streamline simulators, the primary variables of a background (pressure) grid cell is chosen as average of the compositions in the neighborhood of the cell, as in

$$\underline{C}^{cell} = \sum_{k=1}^{N_{SL}} w_k \underline{C}_k . \qquad (16)$$

The total number of streamlines required in a simulation strongly depends on the accuracy of this mapping. A choice must be made as to which streamlines to include in the average and how the weights are assigned. Typically, the average it taken over all streamlines crossing a grid cell, which means that each grid cell must contain at least one streamline. The weights are generally chosen to reflect the relative volumetric contribution of the streamline to the cell (Batycky (1997)). The calculation of this volumetric contribution can be performed if the total flow rate (flux) for the streamline is known. Flow rates can be estimated at wells but this requires that a streamline begins and ends at wells or domain boundaries. This combination of a minimum of one streamline per cell and tracing streamlines from well to well, may lead to rather dense streamline coverage, especially in converging areas such as near-well regions. In compositional simulation this is computationally unattractive. Another disadvantage of volumetric weighting is the sensitivity of the mapping to the streamline location: if a streamline passes closely to the edge of a cell, a slight perturbation in streamline location can lead to large perturbations in the weights.

Because in our approach, we do not have to assign volumetric flow rates with streamlines. we can use different interpolation techniques. In Mallison *et al.* (2005b), we suggest using block Kriging to map compositions from streamlines to the background grid, but other mapping strategies are possible. The main motivation for using Kriging is that is can be high order accurate in smooth regions. Any nearby streamline can be included in the mapping so every cell need not contain a streamline. This allows for coarser streamline coverage and flow rates need not be assigned to streamlines. The Kriging weights depend on the proximity of streamlines to the cell so that greater weight is given to streamlines passing through the center of a cell and less weight is given to distant streamlines. In addition, Kriging accounts for the proximity of streamlines to one another so less weight is given to streamlines passing near one another. Secondary data could be included in a number of ways, although we have not fully pursued this additional functionality, apart from a preliminary investigation in which we used Kriging with an external drift. Our Kriging implementation is based on routines available in GSlib (Deutsch and Journel (1992)).

In our experience, Kriging helps reduce mass balance errors that are typically introduced by the traditional mapping from streamlines to pressure grid. These mass balance errors can also be controlled by increasing the number of streamlines, but for compositional problems it is important to minimize the total number to



(a)

(b)

Figure 4. (a) The commonly used mapping to streamlines assumes a piecewise constant representations of saturations on the background grid (left). The resulting 1D profile along streamlines is also piecewise constant (right); (b) The improved mapping to streamlines utilizes a piecewise linear representation of saturations on the background grid (left). The resulting 1D profile along streamlines is also piecewise linear (right).

avoid excessive computational costs. Its main advantage however is that when using Kriging instead of the commonly used flux-based interpolation algorithm, grid cells need not be crossed by streamlines so coarser streamline coverage is possible. Comparisons between the commonly used mappings and the improved mappings are given in detail in Mallison *et al*. (2006).

The computational cost of Kriging can be significant due to the search required to locate nearby points on the streamline grid and the solution of the local system to determine the Kriging weights. We have not yet optimized our search algorithms. For problems with large number of components, the cost of Kriging does not increase significantly since the same set of weights can be used for mapping all components.

2.5 A higher order global time stepping method

Streamline methods, like many methods including IMPEC based methods, generally treat pressure and velocity as fixed over global time steps. Temporal discretization error and operator splitting errors should be carefully monitored because streamline methods typically use global time steps that are large as compared to Eulerian methods. Because of the strong coupling between flow and transport, errors in the pressure distribution, e.g. near propagating fronts, may affect phase behavior and hence the accuracy of the predicted local displacement efficiency. In Ichiro *et al.* (2003), a strategy was proposed for choosing the global time step based on monitoring changes in the total velocity field. However, ideally, the global time step should be chosen to balance the global time discretization error and operator splitting errors resulting from cross flow treatment with other errors in the method. For example, the mappings in stages three and five of the streamline method (Figure 1) introduce smoothing and mass balance errors which can grow as the number of global time steps is increased, and frequently dominate other numerical errors (Mallison *et al.* (2005b)). Robust selection of global time steps is feasible only when reliable estimates of errors introduced by mappings, constant pressure, discretizations, and cross flow are available, and should take into account also the accuracy of production calculations. This is work in progress. We have made two contributions in this area. First, the new higher order global time stepping method discussed below improves the solutions and allows larger time steps to be used for the same accuracy. Second, the improved mappings discussed in the previous section reduce smoothing and mass balance errors so that more frequent pressure updates are acceptable if needed.

Ingebrigtsen *et al.* (1999) and Crane *et al.* (2000) suggested updating the pressure along streamlines during a global time step to better honor the strong coupling between the pressure and transport equations. For gas injection problems, a similar approach may help to account for volume change at gas fronts. Transmissibilities must be calculated along streamlines, and suitable pressure boundary conditions must be found that for partial streamlines will require space-time interpolation. Future work is needed to investigate these issues.

In our current approach to improve the numerical treatment of the coupling between flow and transport we do not use localized one-dimensional pressure updates, but instead seek a higher order global time stepping method that updates the velocity field and streamlines globally. Fixing the total velocity field at the beginning of a global time step is analogous to forward Euler time stepping and is based on a rectangle rule for the global time integration. In operator notation this can be written,

$$\begin{aligned} \left[p^n, u^n\right] &= \mathrm{FLOW}\left(C^n\right) \\ \left[C^{n+1}\right] &= \mathrm{SL}\left(C^n, p^n, u^n, \Delta t\right) \end{aligned}. \tag{17}$$

Here the *FLOW* operator represents the first two stages of the streamline method and the *SL* operator encompasses the final three stages. For any step size, the temporal error can be reduced by considering a more accurate representation of the total velocity field. For example, if the velocity field is fixed at the midpoint in the global time step then the temporal discretization will be second order accurate. The midpoint rule can be incorporated using a two stage Runge-Kutta time discretization using

$$\begin{aligned} \left[p^n, u^n\right] &= \mathrm{FLOW}\left(C^n\right), \\ \left[C^{n+1/2}\right] &= \mathrm{SL}\left(C^n, p^n, u^n, \Delta t/2\right), \\ \left[p^{n+1/2}, u^{n+1/2}\right] &= \mathrm{FLOW}\left(C^{n+1/2}\right), \\ \left[C^{n+1}\right] &= \mathrm{SL}\left(C^n, p^n, u^{n+1/2}, \Delta t\right). \end{aligned} \tag{18}$$

Consider using (18) to advance the solution over Δt and compare to (17) with a time step Δt/2. Both methods require two pressure solves. The new method (18) has a 50% overhead in transport calculations. However, the compositions in (18) undergo only one mapping, whereas in (17) two mappings are applied to the components.

We illustrate the potential improvements that can be made by employing this new global time-stepping method to the simplified model for miscible flow on the permeability field given in Figure 2. Again, we take M=10, inject at the left boundary with a fixed inflow rate and produce from the right boundary which is kept at fixed pressure, with no-flow conditions at the top and bottom boundaries. An IMPEC-ENO finite difference solution on a 60x60 grid is shown in Figure 5a for 0.15PVI. This scheme employs the 3rd order component-based ENO discretization in space combined with a 3rd order Runge-Kutta time stepping algorithm. Figures 5b and 5c show saturation profiles at that same time for the streamline method with 10 and 20 global time steps, respectively, using the traditional global time stepping approach and the same 3rd order numerical scheme along the streamlines. Figure 5d shows the streamline simulation result for the new global time stepping approach with 10 global time steps.

For this test case, clearly the new global time stepping method gives improved results. We expect that this method will show stronger improvements for gas injection processes. In highly heterogeneous reservoirs in which gravity segregation has been established, streamline positions generally change little even when displacements involve large adverse mobility ratios. However, the magnitude of the velocity will increase in swept regions. Fixing the velocity field at the start of a global time step may therefore lead to the advancement of fronts being under estimated. The higher order global time stepping partially corrects for this by fixing the velocity at the midpoint instead. We are testing this scheme on more challenging gas injection processes to verify this hypothesis.

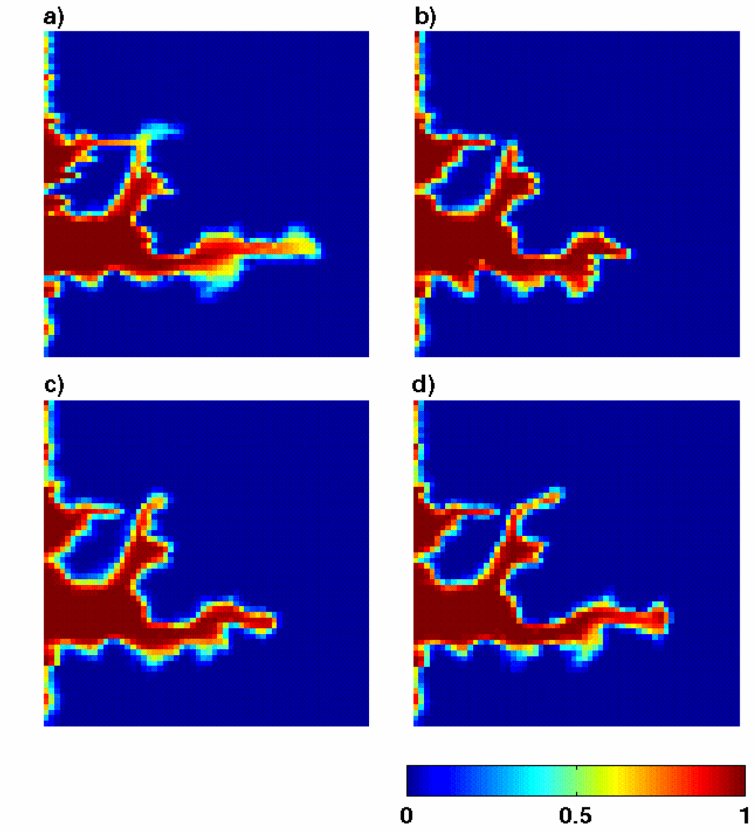


Figure 5. The effect of global time discretization is illustrated for the simplified model for miscible flooding on the permeability field in Figure 2 at 0.15 PVI. Saturation profiles are shown for (a): IMPES-ENO finite difference scheme; (b) Streamline with conventional global time stepping for 10 time-steps; (c): Streamline with conventional global time stepping for 20 time steps; (d) New midpoint global time stepping for 10 time steps.

2.6 Higher order upwind methods for transport along streamlines

We prefer explicit finite difference schemes in the general case for their robustness. In Mallison *et al.* (2005a), we proposed a third order finite difference scheme for two-phase, multicomponent flow. This 1D solver was extended to treat three-phase flow in Valenti *et al.* (2004, 2005). A conservative spatial discretization of equation (11) for grid cell *k* takes the form,

$$\underline{L}_k(\underline{C}) = -\tfrac{1}{\Delta\tau}[u_{d,k+\frac{1}{2}}\underline{F}_{k+\frac{1}{2}} - u_{d,k-\frac{1}{2}}\underline{F}_{k-\frac{1}{2}}]. \qquad (19)$$

In (19) we have used vector notation to represent the compositions and overall fractional flows of each component. To compute the fractional flows in (19), each cell on the grid is flashed and phase properties are computed. The overall fractional flow is then computed for each cell from (8). The spatial accuracy of the finite difference scheme depends on the method used to reconstruct the overall fractional flow at the cell faces. For flow from left to right, the first order SPU scheme uses the approximation $\underline{F}_{k+1/2} = \underline{F}_k$. The third order ENO scheme utilizes a quadratic reconstruction involving a three-point stencil. To minimize oscillations, ENO adaptively chooses the stencil of the reconstruction. We found that component wise, second order TVD reconstructions generated nonphysical behavior when applied to some two-phase multicomponent systems as discussed in Mallison *et al.* (2005a). A third order ENO scheme, also based on a component wise reconstruction, was more robust in these situations and more accurately predicted displacement efficiency. A fifth order weighted ENO (WENO) scheme gave similar results as the 3rd order scheme, but displayed a somewhat greater tendency to oscillate, and did not lead to cost reductions. Also in the three-phase flow studies we performed, the ENO scheme was found to be most robust. We note that implementation of high order flux reconstructions based on a characteristic decomposition is not straightforward due to the weak hyperbolicity of the transport equations. We use a component wise reconstruction of the flux for simplicity, efficiency and because it has performed well in extensive tests. Our flux calculation does not require the use of a Riemann solver because all components move in the direction of the flow.

After discretizing the spatial derivatives the problem resembles a system of ordinary differential equations in the form, $\partial \underline{C}/\partial t = \underline{L}(\underline{C})$. If the SPU scheme is used for the spatial discretization, then the time integration can be performed using forward Euler time stepping,

$$\underline{C}_k^{n+1} = \underline{C}_k^n + \Delta t\, \underline{L}_k(\underline{C}^n), \quad k = 1,\ldots,N_{cells}. \tag{20}$$

The local time step Δt is governed by the usual CFL restriction. If the ENO reconstruction is used, forward Euler time stepping is not sufficiently accurate and can lead to stability problems. Therefore, we use an explicit third order Runge-Kutta scheme that has been suggested and extensively tested for use with the ENO reconstruction. In this scheme, for each grid cell k we have

$$\begin{array}{rcccccl} \underline{C}_k^{(1)} & = & \underline{C}_k^n & + & & & \Delta t\ \underline{L}_k(\underline{C}^n) \\ \underline{C}_k^{(2)} & = & \frac{3}{4}\underline{C}_k^n & + & \frac{1}{4}\underline{C}_k^{(1)} & + & \frac{\Delta t}{4}\ \underline{L}_k(\underline{C}^{(1)}). \\ \underline{C}_k^{n+1} & = & \frac{1}{3}\ \underline{C}_k^n & + & \frac{2}{3}\ \underline{C}_k^{(2)} & + & \frac{2\Delta t}{3}\underline{L}_k(\underline{C}^{(2)}) \end{array} \tag{21}$$

The time step restriction for this scheme is the same as that of forward Euler and when used with appropriate spatial discretizations this time integration scheme tends not to introduce oscillations. For various flows with miscible and near-miscible reservoir fluids, varying WAG ratios and different relative permeability models (Stone 1, Stone 2 and Baker) the ENO scheme out performs the TVD scheme in accuracy and robustness regardless of the selection of limiter (Valenti (2005)). Figure 6 gives an example of a 7-component near-miscible WAG process in 1D comparing coarse grid simulations with SPU, TVD-Van Leer and ENO using 160 grid points relative to a reference solution using 10240 grid points computed with the second order TVD method with minmod limiter.

If the pressure (volume balance) is not updated along a streamline during the propagation of fluids between global pressure solves additional care must be taken to conserve both mass and volume. As components transfer between phases in the course of transport, new mixtures form with partial molar volumes different from the original mixtures along the streamline. Consequently, the mixtures may swell or shrink creating a volume discrepancy at the end of a local time step. Volume change on mixing acts to slow down or speed up

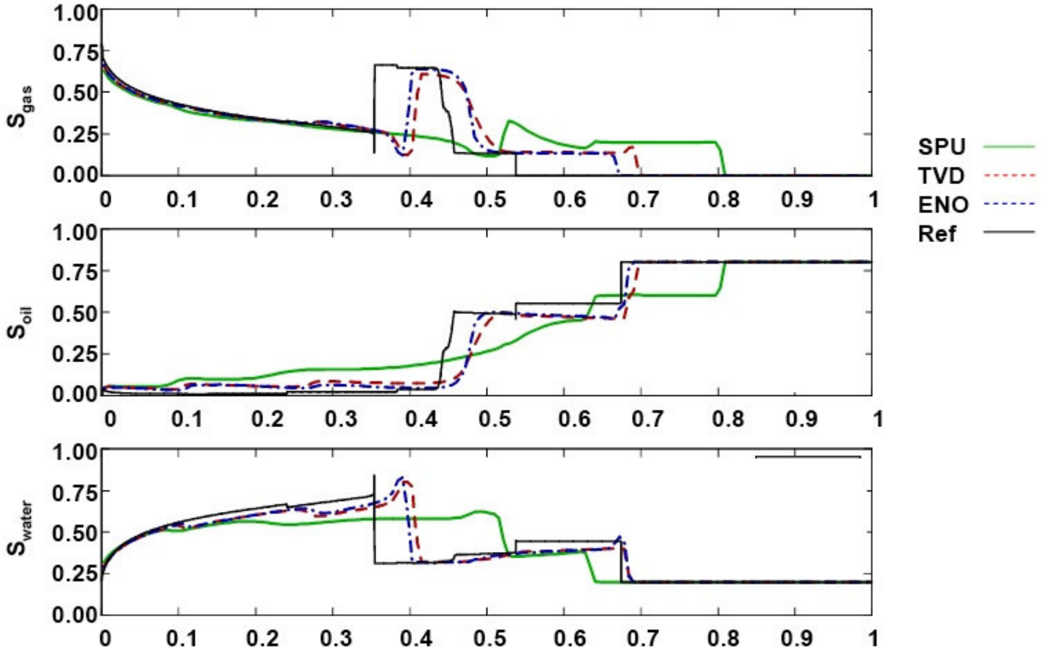


Figure 6. Near miscible WAG. Comparison of SPU, TVD and ENO scheme for 160 grid points compared to a high resolution numerical solution (Valenti et al. 2005).

the propagation velocity of compositions (and saturations) in the porous media. This effect can be illustrated by considering CO_2 injection into an undersaturated reservoir fluid. In undersaturated oils, significant amounts of CO_2 can be dissolved in the liquid hydrocarbon phase without forming an equilibrium vapor phase. The high solubility of the CO_2 in the liquid hydrocarbon phase will reduce the propagation speed of the CO_2 front. Accordingly, accurate representation of this effect is required to predict the front propagation adequately. Orr (2007) reports various analytical solutions to compositional 1D displacements with and without volume change on mixing and demonstrate that ignoring the effect may lead to significant errors (20% or more) in the prediction of propagation velocities.

In finite difference/volume compositional simulation, the volume discrepancy caused by volume change on mixing is typically accounted for by appropriate source/sink terms in the pressure solve (Bell & Trangenstein, 1989). However, in compositional streamline simulation, the pressure may not be updated along the streamlines between global time steps and care must be taken to reduce the potential error in propagation velocities. The volume discrepancies along a single streamline can be eliminated by an implicit or an explicit correction of the local dimensionless flow velocities.

The implicit correction suggested by Dindoruk (1992) involves an iterative scheme for correction of the effluent flow velocity $u_{k+1/2}$ of a given grid cell. The correction converges after a few iterations. However, each iteration involves a re-evaluation of the phase equilibrium and consequently renders the approach less efficient than solving the 1D pressure equation at each time step during the transport along a streamline. We propose to instead use an explicit approach for correction of the volume discrepancies. Here, the volume discrepancy is carried forward in time and the dimensionless velocity u_d is adjusted between local time steps by

$$u_{d,k+1/2} = u_{d,k-1/2} + \varepsilon \frac{\Delta\tau}{\Delta t}(q_{vc} - 1)\,, \tag{22}$$

where $q_{vc} = V_{fluid}/V_{cell}$, and V_{fluid} and V_{cell} the fluid and cell volumes, respectively. The coefficient ε satisfies $\varepsilon < 1$, and is selected to ensure stability of the overall numerical scheme. In the explicit correction of flow velocities, the emphasis is to ensure that any volume discrepancy developing in the course of the 1D solve is reduced between local time steps. This approach is sufficiently accurate and more efficient than the implicit correction of the flow velocities. A comparison of the proposed method for accounting for VCOM is given in Figure 7 for a near-miscible diaplcement of a 6-component oil by pure CO_2. The two approaches are seen to be in good agreement.

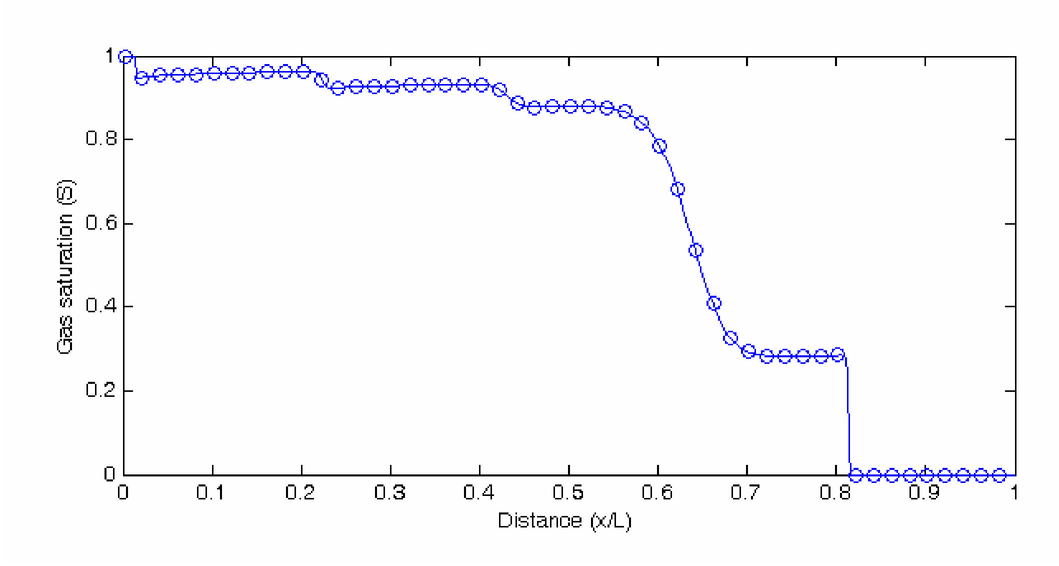


Figure 7. Comparison of iterative (line) and non-iterative (symbol) treatment of VCOM.

We stress that accounting for volume change on mixing is not equivalent to viewing streamlines as fluid carriers. Additional discussion of volume change on mixing can be found in the book by Orr (2007).

The largest computational cost associated with both the SPU and ENO schemes is due to flash calculations, even if these are computed using optimized methods and only where needed. As a consequence of involving three stages, the computational costs of the standard ENO scheme would be three times larger than of SPU scheme per grid cell in the two-phase regions. The added costs are not that severe as ENO can be used with coarser 1D grids and still provide greater accuracy. However, the number of stages need not dictate the frequency of flash calculations. Rather, the time scale for performing flash calculations should be determined by physical considerations. To improve the efficiency of the ENO scheme we have investigated a more efficient, but approximate strategy for evaluating phase behavior at the two predictor stages of the Runge-Kutta scheme given by equation (21). Rather than performing full flash calculations we assume that the equilibrium K-factors, $K_i = x_{ig}/x_{io}$, remain fixed at the sub-time-step scale. The evaluation of phase equilibrium can thus be reduced to that of one step of the successive substitution algorithm. We utilize this approximation everywhere except near phase boundaries and critical points. Overall, this modification improves the efficiency of ENO by approximately a factor of two in our tests without degrading accuracy or stability.

We also implemented adaptive mesh refinement along streamlines to allow a coarser grid density to be used away from propagation fronts that require a fine grid for accuracy. We utilized a patch-based AMR method, which construct a hierarchy of nested grids (Berger & Oliger, 1984). For hyperbolic problems, patch-based AMR has the advantage that refinements and coarsening can be performed in time as well as space. At the coarsest level, a single grid covers the entire computational domain. We flag regions that require refinement based on composition gradients between cells. Flagged regions are buffered by one cell in the upwind direction to ensure smoothness at the borders of patches. Downwind buffering is determined by an estimate of the maximum propagation speed and time step size. Patches separated by fewer than four coarse cells are merged into a single patch. We allow for integer refinement ratios of two, three, or four between patches and their parent grid. We initialize compositions on patches through quadratic interpolation from the parent grid. Time integration begins by taking a single step on the coarsest level. Patches are then advanced to the same time level by taking a number of steps equal to the refinement ratio. This process is recursively repeated until all grid levels have been advanced to the same time level. Solutions on coarse levels are corrected by conservatively averaging the fine solutions over coarse cells. A critical step in the AMR algorithm is to specify boundary conditions at the borders of patches. We introduce two ghost cells at patch borders specify the compositions at the ghost cells by quadratic interpolation in space and time.

AMR was applied to find the solutions depicted in Figure 3, with a refinement ratio of two and three grid levels. The computational savings can be large when fronts, at which refined grids are necessary, are well separated.

2.7 Operator splitting for gravity

Streamline methods use operator splitting to include the effects of gravity segregation. This approach is described, amongst others, by Gmelig-Meyling (1990, 1991) and Bratvedt et al. (1996). In Jessen et al. (2004), we developed an adaptation of gravity operator splitting for compositional problems. Above, we derived the transport equation as equation (9) given by

$$\varphi \frac{\partial C_i}{\partial t} + \left|\underline{u}_t\right| \frac{\partial (u_d F_i)}{\partial \xi} + \left|\underline{g}\right| \frac{\partial G_i}{\partial z} = 0, \qquad i = 1, ..., N_c, \tag{23}$$

where ξ is arc length measured along a streamline and z = z(ξ) is depth. Here, we assume that the gravity vector is aligned with the z-direction. In the traditional operator splitting framework, this equation is first solved without the gravity term, and then followed by a gravity step in which

$$\varphi \frac{\partial C_i}{\partial t} + \left|\underline{g}\right| \frac{\partial G_i}{\partial z} = 0, \qquad i = 1, ..., N_c, \tag{24}$$

is integrated numerically along gravity lines over the global time step. In compositional problems, care must be taken when implementing this operator splitting as the performance of compositional displacements may depend strongly on the displacement path. Rather than taking into account phase behavior when moving the compositions along gravity lines, the initial approach as described in detail in Jessen and Orr (2004) is to transport phases along gravity lines in a pseudo-immiscible fashion.

The pseudo-immiscible approach is quite simple in character. At the start of the gravity correction step, the properties of the liquid and gas phases in each grid block along a gravity line (a vertical line of grid cells) are determined. Each of these grid block phases, which we refer to as segments, is then moved from grid block to grid block under gravity segregation for a number of local time steps using the method for immiscible fluids outlined by Batycky (1997).

At each local time-step, representative phase saturations in grid cell *k* are calculated by summation over all phase segments n_k present in cell *k*. For example, the gas saturation S_k^g is given by $S_k^g = \sum_{i=1}^{n_k} S_i^g$. Representative phase viscosities and densities are then computed for the grid cell using saturation averages. At the end of the gravity step, the gas and liquid segments in each grid block are combined to form new overall mole fractions of the components. Using this approach, phase behavior is only taken into account fully during the convective step along streamlines. The added costs of this pseudo-immiscible splitting approach are low as flash calculations only occur at the start of the gravity correction step.

The initial results are encouraging. Here, we include one illustration using a 2D heterogeneous displacement of the reservoir fluid by CO_2 at a constant reservoir temperature of 387K. The conditions are near-miscible: the saturation pressure of the reservoir fluid at the given temperature and the Minimum Miscibility Pressure for injection of CO_2 are 220 and 247 atm., respectively. The permeability field is shown in Figure 8. The injector is again completed over the entire left-most column. The producer is completed in the lower right hand corner and is operated at a bottom-hole pressure of 225 atm. The injection rate is 1.5 Rm^3/day. Figure 9 shows the gas saturation maps after 0.25 and 0.5 PVI as predicted by the compositional streamline simulator with gravity operator splitting and the commercial reservoir simulation package Eclipse 300. There is good agreement between the predicted saturation distributions. After gas breakthrough, the recovery predicted by CSLS in this case is slightly higher than that predicted by E300, which is due to numerical diffusion in the finite difference simulation. Here, numerical diffusion acts to smear the displacement front reducing the local displacement efficiency relative to that predicted by the streamline approach, while at the same time marginally delaying the breakthrough of the injected gas.

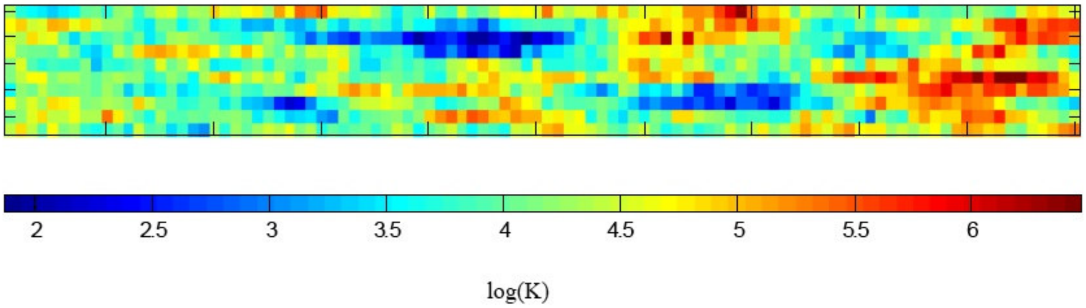


Figure 8. Permeability field used for the gravity operator splitting example in section 4.3. The injection well is completed over the entire left column, the producer in the lower right hand corner.

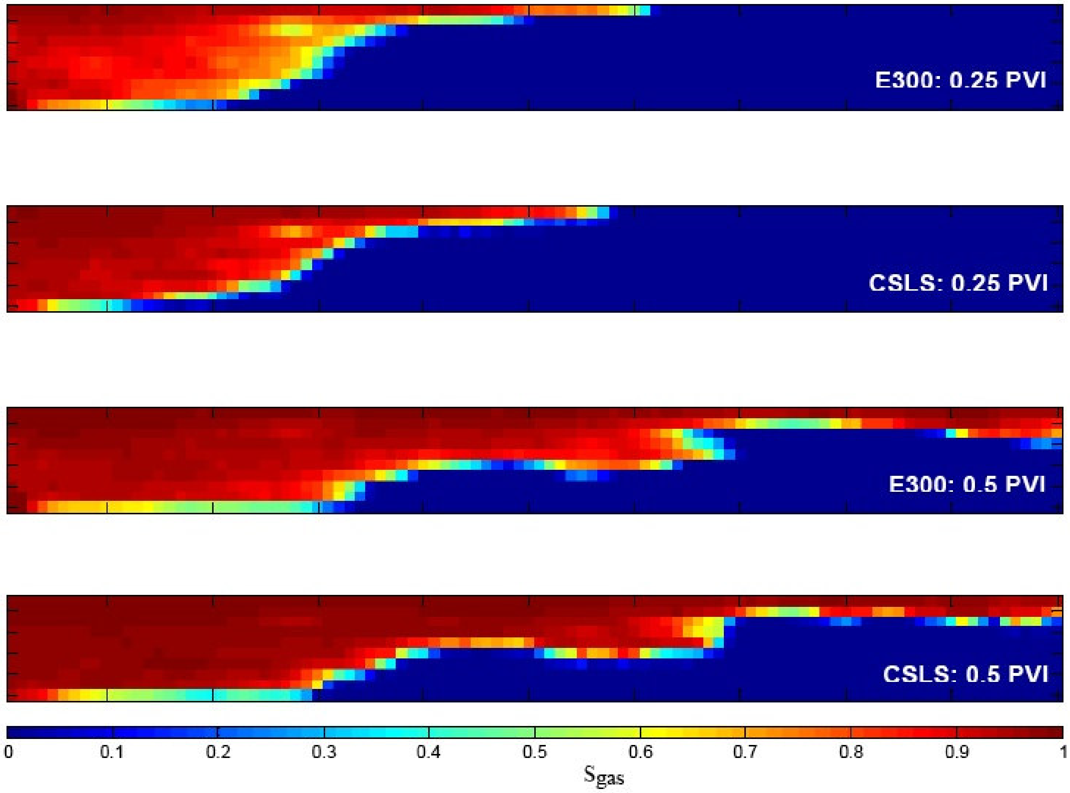


Figure 9. Gas saturation maps as predicted by CSLS and E300 after 0.25PVI and 0.5PVI for injection into the reservoir of Figure 8.

In this and other gravity operator splitting approach, global time step sizes generally must be constrained by requiring that a front moves only a limited number of grid blocks during a global step. In some cases, Batycky (1997) recommended limiting the movement of fronts to a single grid block per time step. Naturally with small global time step sizes, mapping errors will quickly dominate and may strongly influence performance of the streamline simulator. There is certainly a need for a rigorous control of operator splitting errors. Alternative splitting techniques could be investigated. For example, the order in which convective and gravity updates are performed could be alternated, or results of the two different orderings could be combined. Another approach is to investigate the use of a hybrid approach where an Eulerian method is used in the early stages to handle initial gravity segregation, followed by a streamline method in the later stages.

2.8 Applications

In the first application we take the permeability field from layer 19 of the Tenth SPE Comparative Solution Project of Christie and Blunt (2001). It is shown along with a sample of streamlines in Figure 10. The extent of the domain is 670.56 m (2200 ft) in the x-direction and 365.76 m (1200 ft) in the y-direction. The model was discretized with a uniform Cartesian grid containing 220 cells in the x-direction and 60 cells in the y-direction. The size of each grid cell was 3.048 m in x, 6.096 m in y and 0.6096 m in the vertical direction. The porosity was uniform and equal to 0.2. We injected CO_2 along the boundary x=0 at a total rate of 5 m^3/day. Production took place at the boundary x= 670.56 m where the pressure was fixed at 235 atm. Zero flux boundary conditions were applied at y=0 and y=365.76 m.

Four numerical methods were used: a standard IMPEC scheme with an SPU discretization of the transport equations, an IMPEC scheme with an ENO-3 discretization of the transport equation, a streamline method using the SPU scheme along streamlines with standard mappings, and a streamline method using the ENO-3 scheme along streamlines and our improved mappings. We should mention that these are all home brew. Both streamline methods were tested with 10, 40 and 160 global time steps. Figure 11 shows the results of applying each method to this displacement after 1300 days of injection. The mole fraction of CO_2 is shown. As expected, the IMPEC-ENO finite difference scheme produced less numerical diffusion than the IMPEC-SPU scheme. The amount of numerical diffusion introduced by both streamline methods increased with the number of global time steps. However, the increase was less pronounced in the cases where the improved mappings were used. Mass balance errors are small (below 1%) for all of the streamline runs in this example.

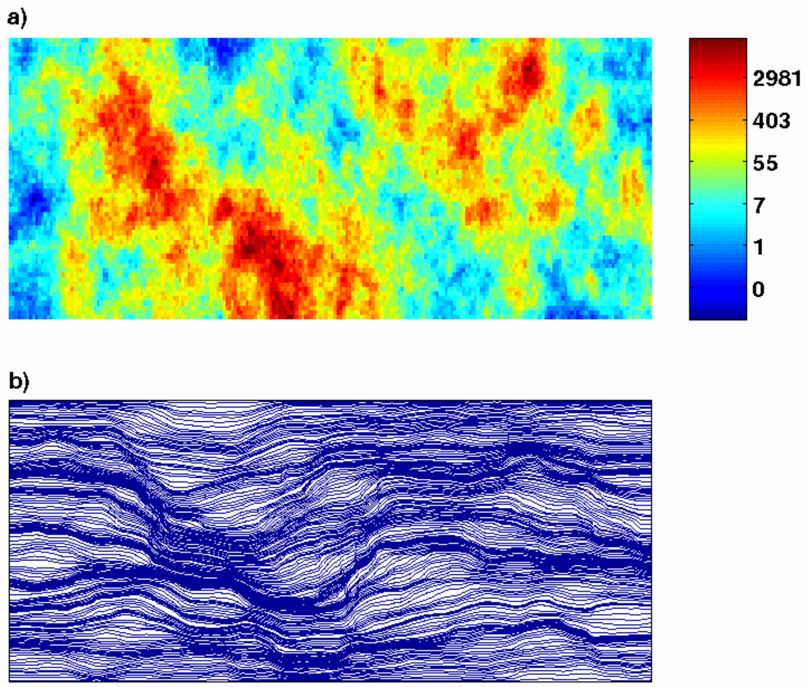


Figure 10. (a) Permeability field, layer 19 SPE 10 Comparitive project; (b) sample streamlines on permeability field given in (a).

To demonstrate an application of our compositional streamline simulation approach at larger scale, we present a simulation example where CO_2 is injected into an oil reservoir with the composition given in Tables 1 and 2. The dimensions of the simulated reservoir are 360m x 360m x 32m represented by a 180x180x16 cell simulation grid ($2x2x2m^3$) with the spatial permeability variation shown in Figure 12. Pure CO_2 is injected in 9 wells and oil/gas is produced from 27 production wells in a repeated inverted 5-spot pattern. All wells are completed over the entire thickness of the reservoir. The reservoir is isothermal at 373.15 K and the initial average pressure in the top layer is 220 bar. The injectors are operated at a constant bottomhole pressure of 240 bar and the production wells are producing at a constant bottomhole pressure of 210 bar. Multicontact miscibility between the reservoir oil and the injected gas developes at 260 bar and hence the presented simulation is a near-miscible displacement. Figure 13 compares the saturation distribution after 100 days of injection as predicted by compositional streamnline simulation and a commercial finite difference simulator run in adaptive implicit mode. The predicted spatial distribution of the injected gas from the two simulation approaches is found to be in good agreement. The CPU requirement for the streamline simulation was approximately 3000 sec wheras the simulation time for the FD simulation was approximately 18000 sec. Both simulations were run on a single 2.8 GHz processor. We note thate a direct comparison of the CPU time may not give a clear picture of the efficiency of the two simulation approached. The commercial finite difference code, ECLIPSE 300, was run with the default convergence settings and additional speed-up might be available through tuning of these parameters. The pressure solver used in the streamline code, on the other hand, is an older source code version of the AMG solver release 1.5. Hence, additional speed-up of the streamline code is available through use of more efficient pressure solvers. However, the efficiency of our research code (CSLS) is clearly demonstrated in this calculation example.

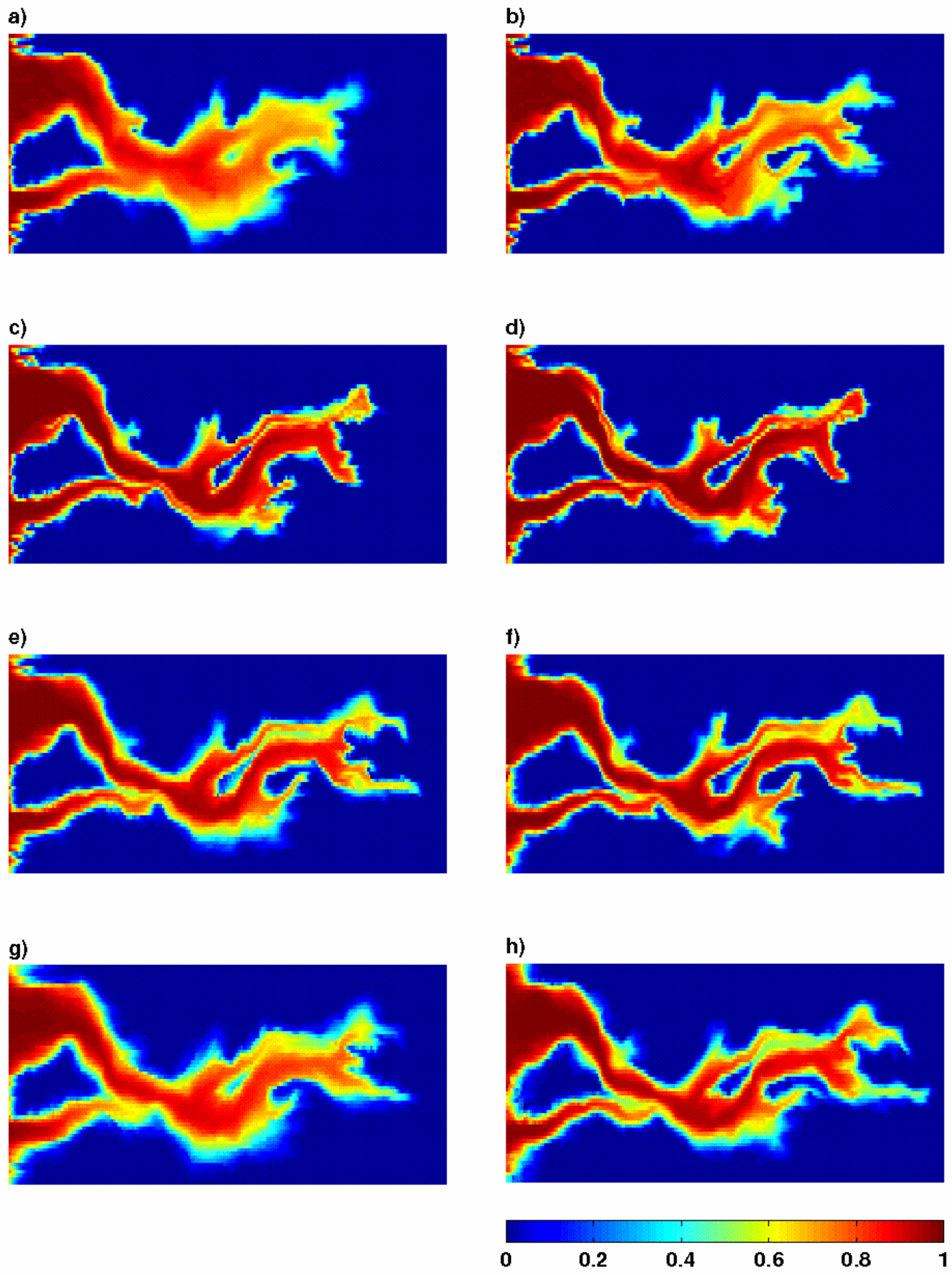


Figure 11. Mole fraction of CO_2. Mole fraction of CO_2 after 1300 days of injection computed with the following numerical methods: IMPEC-SPU (a), IMPEC-ENO (b), streamline method with SPU and standard mappings with 10 (c), 40 (e) and 160 (g) time steps, streamline method with ENO-3 and improved mappings with 10 (d), 40 (f) and 160 (h) time steps.

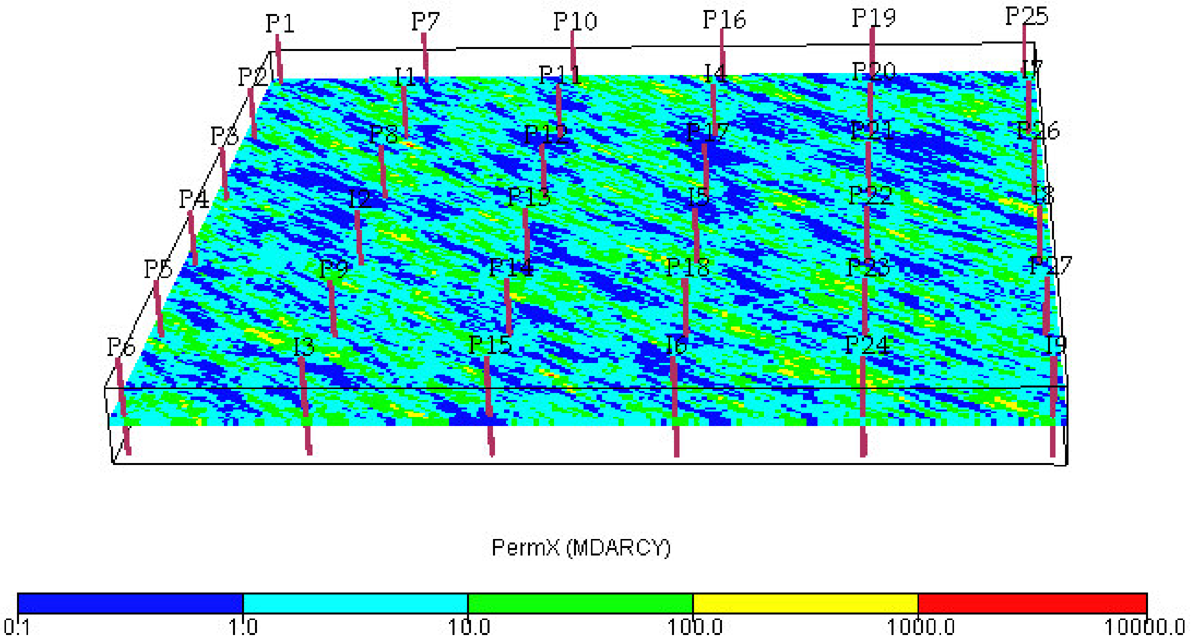


Figure 12. Permeability field for 3D simulation example (layer 8 of 16)

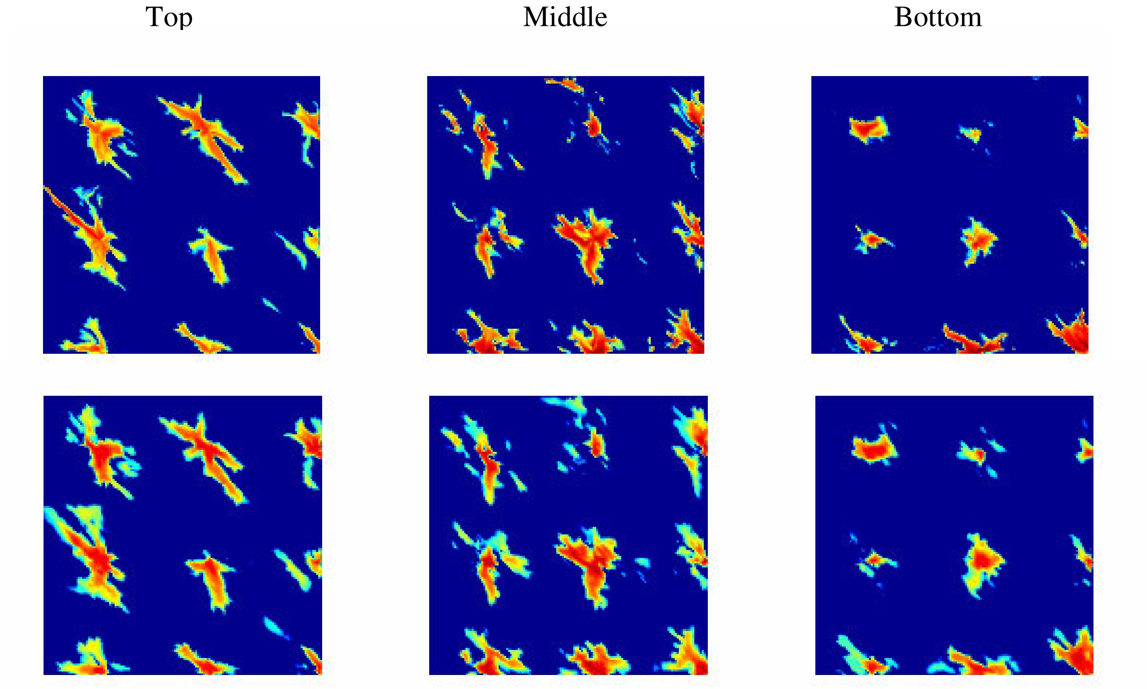


Figure 13. Gas saturation after 100 days of injection: CSLS (top) and AIM FD (bottom).

CPU time: CSLS ~3000 sec, AIM FD ~18000 sec.

	P_c (atm)	T_c (K)	ω	M_w	Z_c	x_{oil}	y_{sep}
$N_2\backslash C_1$	44.61	189.5	0.0085	16.16	0.2898	0.4630	0.8027
CO_2	71.95	304.2	0.2280	44.01	0.2706	0.0164	0.0255
C2-C5	39.89	387.6	0.1673	45.57	0.2759	0.2052	0.1582
C6-13	32.58	597.5	0.3861	117.74	0.2567	0.1911	0.0136
C14-24	17.22	698.5	0.8078	248.83	0.2197	0.0811	0
C25-80	11.39	875.0	1.2314	481.52	0.1825	0.0432	0

Binary interaction coefficients (lower triangle)

0.11883				
0.00071	0.15			
0.000778	0.15	0		
0.01	0.15	0	0	
0.011	0.15	0	0	0

Table 1. Fluid description at T_{res} = 387.45 K

Name	Mole fraction	Name	Mole fraction	M_w (g/mole)	ρ (g/cm^3)
N2	0.0045	C7	0.0377	92	0.7294
CO2	0.0164	C8	0.0428	106	0.7509
H2S	0.0000	C9	0.0270	120	0.7739
Methane	0.4585	C10	0.0169	137	0.7835
Ethane	0.0715	C11+	0.1658	288	0.8835
Propane	0.0674				
i-Butane	0.0084				
n-Butane	0.0311				
i-Pentane	0.0103				
n-Pentane	0.0165				
Hexane	0.0252				

Table 2. Compositional description underlying the fluid description in Table 1.

3 New compositional solvers: relaxation schemes

Solving the transport equations along the streamlines is challenging due to strong non-linear coupling and weak hyperbolicity of the problems in phase space. Relaxation schemes aim to overcome these problems by reformulating the governing equations as a linear system of hyperbolic equations, with the specific intent of controlling hyperbolicity; the nonlinearity is restricted to the source term only. In this work, we investigated whether these methods are useful in the context of miscible gas injection processes. Indeed, they provide a robust way to avoid problems with the loss of strict hyperbolicity commented on earlier.

3.1 The relaxation formulation

Relaxation schemes (Jin & Xin 1995) and central schemes (Nessyahu & Tadmor 1990, Kurganov & Tadmor 2000) avoid the use of nonlinear Riemann solvers and do not rely on the specific eigenstructure of the problem. Relaxation schemes overcome the dependency on Riemann solutions by solving an approximate problem, called the relaxation system, instead of solving the original problem. The relaxation system replaces the original nonlinear and homogeneous conservation system with a linear hyperbolic system, with a stiff nonlinear source term.

As an example, consider the scalar conservation law

$$u_t + f(u)_x = 0,\ u(x,0) = u_0(x)\,. \tag{1}$$

The relaxation system as proposed by Jin and Xin for Equation (1), takes the form

$$\begin{bmatrix} u \\ v \end{bmatrix}_t + \begin{bmatrix} 0 & 1 \\ a^2 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}_x = \begin{bmatrix} 0 \\ -\frac{1}{\varepsilon}\left(v - f(u)\right) \end{bmatrix}, \tag{2}$$

$$u(x,0) = u_0(x)\,,\ v(x,0) = f(u_0)\,,$$

where the relaxation rate ε is a constant satisfying $0 < \varepsilon << 1$, and the parameter a, which is also a constant and referred to as the sub-characteristic speed, is chosen so that $-a < f'(u) < a$, $\forall u$, i.e. $a = \max\left|f'(u)\right|$.

The relaxation system approximates the original problem with a small dissipative correction controlled by ε. For ε sufficiently small, the original system is recovered. Numerically solving the relaxation formulation has one special advantage. Characteristic decomposition, though still the modus operandi, is now done on a linear hyperbolic structure with a full set of eigenvectors that is constant throughout the entire domain.

Although the relaxation system seems simpler to solve than the original nonlinear system, it now has a stiff source term, additional numerical diffusion and is twice as large as the original system. The stiff source term can be effectively handled by operator splitting (Jin 1995, LeVeque 2002), wherein the system is solved in two separate steps, an advection step, and an ODE step, as explained below. This introduces a splitting error, but it is of the order $\mathrm{O}\left(\Delta t^{r}\right)$, where r is the number of RK stages used for time-stepping. The overhead of computing the additional unknown v can be avoided by resorting to sudden relaxation, i.e. simply setting $v = f(u)$ in the ODE step. The numerical diffusion however, depends on the magnitude of the sub-characteristic speed a. If the sub-characteristic speed is constant for the entire domain, as in the Jin-Xin scheme, the numerical diffusion will always be greater than that of the corresponding upwind schemes.

3.2 Governing equations for 2-phase flows

The phase compositions for 2-phase systems are often expressed in terms of equilibrium ratios (K-values), given for each component p by

$$K_p = \frac{c_{pV}}{c_{pL}} = \frac{\text{volume fraction of component } p \text{ in vapor phase}}{\text{volume fraction of component } p \text{ in liquid phase}} . \qquad (3)$$

Generally the K-values are variable and computed as a part of the flashes. For our test cases we use systems with simplified phase behavior and assume that the K-values are constant. This assumption alters neither the nonlinear nature nor the weak hyperbolicity of the systems. The phase viscosities can be computed by Lohrenz-Bray-Clark correlations (Lohrenz et al.1964). The viscosity ratio can also be assumed to be constant without altering the primary behavior. Volume occupied by a component can change as it transfers between the phases. Volume change on mixing does not change the character of the displacement; only the wave velocities change. Although in realistic settings volume change on mixing is taken into account, and is accounted for by a pressure correction after components are advected (Dindoruk 1982), we will ignore it here to simplify the mathematics and numerics.

To summarize, our simplifying assumptions are that

- the flow is advection dominated, that is, diffusion is negligible,
- the equilibrium ratios (K-values) are constant,
- the viscosity ratio between the two phases is fixed,
- the mixing is ideal, that is, volume change on mixing is negligible.

The conservation equations in 1D, with the above assumptions, are represented by,

$$\frac{\partial C_p}{\partial t}+\frac{\partial F_p}{\partial x}=0\text{, for } p=1,2,\cdots n_c-1 \tag{4}$$

where n_c is the number of components, C_p is the overall volume fraction of component *p* described by $C_p = c_{pV}S_V + c_{pL}S_L = c_{pV}S_V + c_{pL}(1-S_V)$, c_{pV} and c_{pL} are the volume fractions of component *p* in vapor and liquid phases, S_V and S_L are the vapor and liquid saturations, F_p is the overall fractional volumetric flow of component *p* described by $F_p = c_{pV}f_V + c_{pL}f_L = c_{pV}f_V + c_{pL}(1-f_V)$, and f_V and f_L are the vapor and liquid fractional flows. In the single phase region we have $F_p = C_p$. Henceforth we will drop the subscript on the vapor saturation and the fractional flow and represent them as S and f.

An additional constraint is that the volume fractions of each component must sum to 1, so we have

$$\sum_{p=1}^{n_c} C_p = 1,\ \sum_{p=1}^{n_c} c_{pV} = 1\text{, and } \sum_{p=1}^{n_c} c_{pL} = 1. \tag{5}$$

The fractional flow function is given by

$$f=\frac{k_{rV}(S)(1-k_{rL}(S)N_g\sin\theta)}{k_{rV}(S)+Mk_{rL}(S)}, \tag{6}$$

where k_{rV} and k_{rL} are vapor and liquid relative phase permeabilities that depend on saturation S. We use quadratic permeabilities. N_g is the gravity number, θ is the dip angle and M is the constant viscosity ratio. A typical fractional flow function for horizontal flow $(\theta = 0)$, using quadratic relative permeabilities, assuming zero residual liquid saturation, and assuming zero critical vapor saturation looks like

$$f = \begin{cases} \dfrac{S^2}{S^2 + M(1-S)^2} & \text{in two phase region,} \\ 1 & \text{in vapor phase,} \\ 0 & \text{in liquid phase.} \end{cases} \tag{7}$$

Two-component displacements

At the simplest level we have two component (binary) displacements, where a light gas displaces a heavy hydrocarbon. This is a generalization of displacement of oil by water, the Buckley-Leverett problem (Lake 1989). Studying two-component displacement is useful because the solution to the two-component problem reappears as segments of the solution in systems with more components. We use the binary problem to analyze the behavior of the relaxation schemes. Since this is a scalar conservation law we do not have points of weak hyperbolicity.

The governing equation for binary displacement, given in terms of the injected component is

$$\frac{\partial C_1}{\partial t} + \frac{\partial F_1}{\partial x} = 0 \ , \tag{8}$$

with the initial condition $C_1(x,0) = \begin{cases} C_1^J & \text{if } x < 0 \\ C_1^I & \text{if } x > 0 \end{cases}$,

where C_1 and F_1 are described as $C_1 = c_{1V}S + c_{1L}(1-S)$, $F_1 = c_{1V}f(S) + c_{1L}(1-f(S))$,

and $f(S)$ as given in Equation (7) .

Note that for the Buckley-Leverett problem, each component exits only in one phase, and hence the volume fraction of the displaced component and the fractional flow reduce to $C_1 = c_{1V}S$, $F_1 = c_{1V}f(S)$, respectively, and the governing equation becomes $\frac{\partial S}{\partial t} + \frac{\partial f}{\partial x} = 0$. Figures 1 and 2 show the fractional flow curve and the propagation velocity for the binary displacement problem.

In the single phase regions, $F_1 = C_1$. In the two-phase region, the curve is S-shaped, similar to the Buckley-Leverett flow curve.

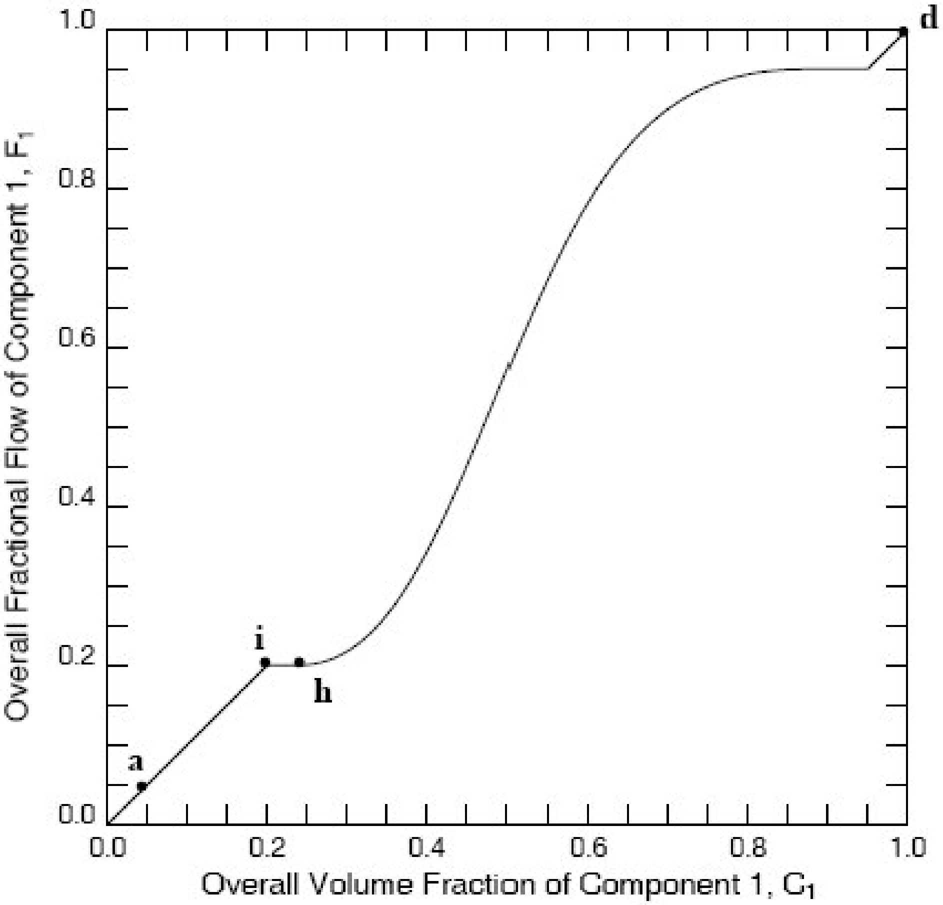


Figure 1: Fractional flow function $F_1(C_1)$. In the two-phase region, the function is S-shaped.

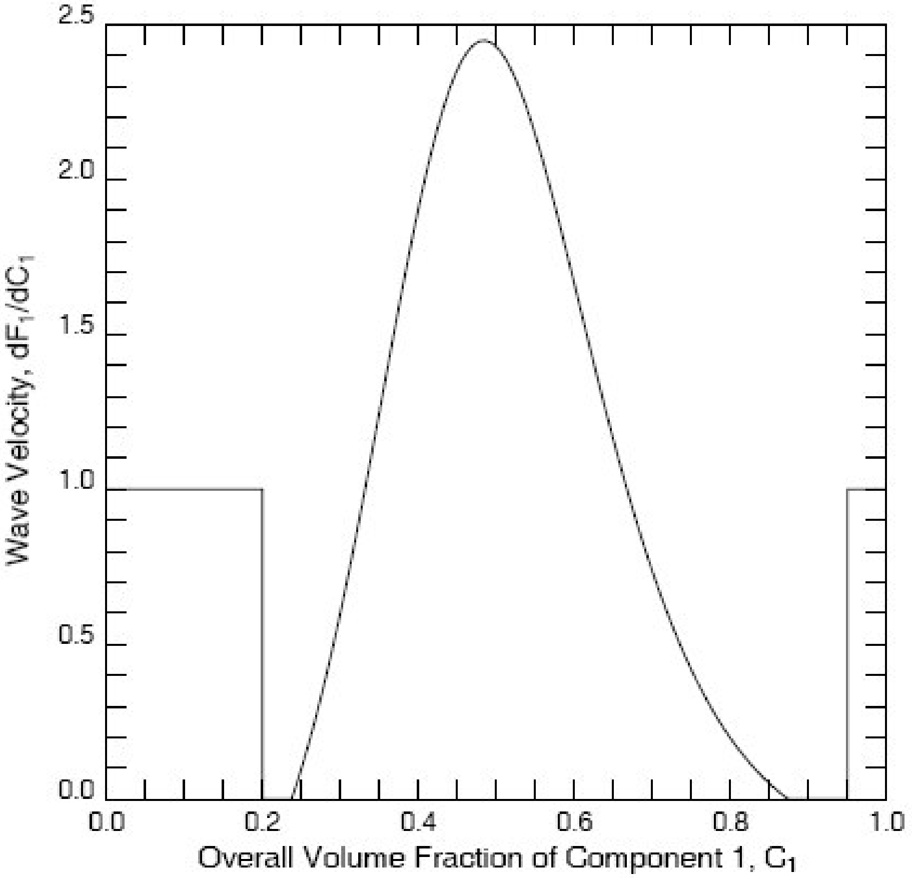


Figure 2: Derivative $\frac{dF_1}{dC_1}$ of the fractional flow function shown in figure 1.

We look at the solution profiles of two test cases shown in Figures 3 and 4. In the first case, a pure light component (point **d** on the fractional flow curve) displaces a pure heavy component (point **a**). The solution for the first case, shown in Figure 3, consists of a leading shock, a spreading wave and a trailing shock. In the second case, a mixture of two components (located by the point **h** on the flow curve) displaces the pure component (point **a**). The solution, depicted in Figure 4, has a leading shock with unit velocity and a trailing shock with zero velocity. The zone between the two shocks is called a zone of constant state. Here the composition remains constant (point **i**) but has two velocities, the leading shock velocity and the trailing shock velocity.

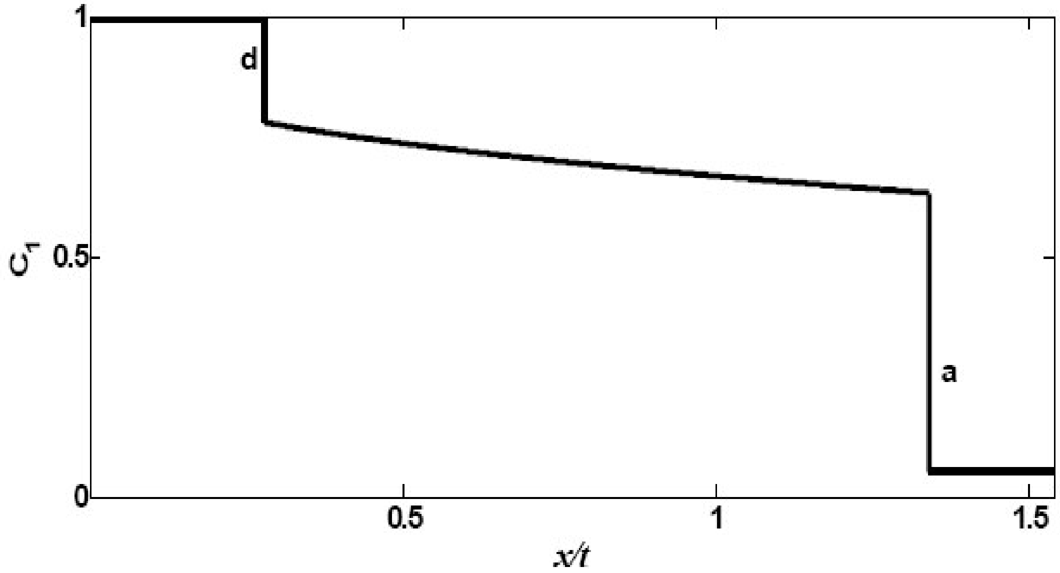


Figure 3: Binary displacement – test case 1, pure light component (d) displaces pure heavy component (a).

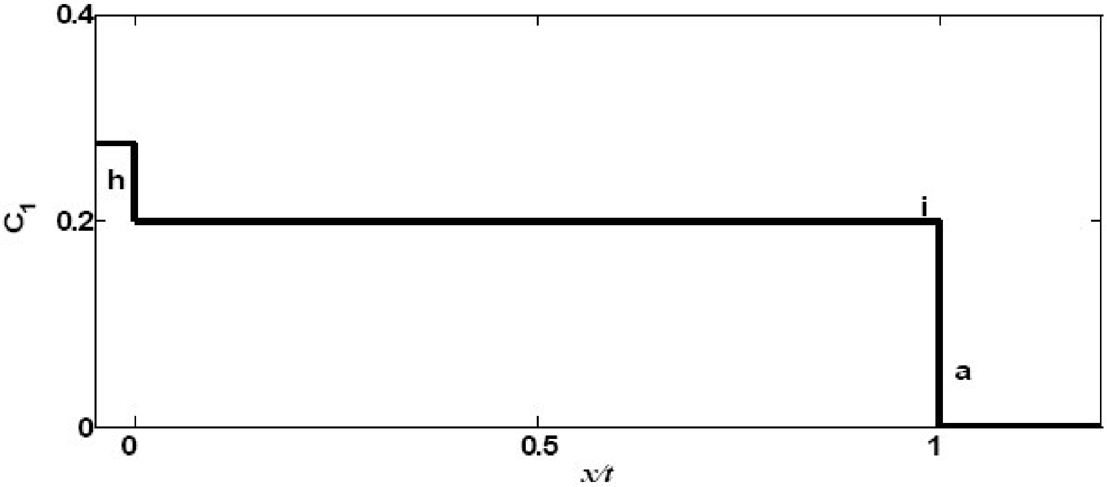


Figure 4: Binary displacement - test case 2, mixture of two components (h) displaces pure component (a). The trailing shock has zero speed.

Ternary systems

For ternary systems, in addition to the composition profiles, we also study the ternary phase diagrams to obtain more information about the phase behavior of the compositions. Ternary phase diagrams represent the component concentrations of all possible mixtures of the three components in a two-dimensional space. Because the volume fractions of the three components sum to one, the phase compositions can be conveniently represented on an equilateral triangle (Lake 1989). Each vertex represents 100% of the component associated with that vertex, and the opposite vertex 0% (Figure 5). Each point within the triangle represents a mixture of the three components; the volume fractions are read from the perpendicular distance from that point to the three sides of the triangle. For gas/oil systems, the component associated with the top vertex of the triangle is usually the lightest, and the component associated with the bottom left vertex is usually the heaviest.

The conservation equations for 3-component nonlinear systems are given by

$$\frac{\partial C_1}{\partial t}+\frac{\partial F_1}{\partial x}=0, \tag{9}$$

$$\frac{\partial C_2}{\partial t}+\frac{\partial F_2}{\partial x}=0,$$

with the initial condition $C_p(x,0)=\begin{cases} C_p^J & \text{if } x<0 \\ C_p^I & \text{if } x>0 \end{cases}$ for $p=1,2$ and with

$$C_1=c_{1V}S+c_{1L}(1-S), \qquad C_2=c_{2V}S+c_{2L}(1-S),$$

$$F_1=c_{1V}f(S)+c_{1L}(1-f(S)), \quad F_2=c_{2V}f(S)+c_{2L}(1-f(S)),$$

and $f\left(S\right)$ as given in (7) .

This system has two eigenvalues given by (Figure 6)

$$\lambda_{\mathrm{t}}=\frac{\partial F_1}{\partial C_1}=\begin{cases} \frac{df}{dS} & \text{in the two phase region,} \\ 1 & \text{in the single phase region,} \end{cases} \tag{10a}$$

and

$$\lambda_{\mathrm{nt}}=\frac{F_1+q}{C_1+q}\text{, with } q=\frac{c_{1L}^2}{\gamma(K_1,K_2,K_3)}. \tag{10b}$$

The corresponding eigenvectors are

$$\vec{e}_{\mathrm{t}}=\begin{bmatrix}1\\0\end{bmatrix} \text{ and } \vec{e}_{\mathrm{nt}}=\begin{bmatrix}1\\ \dfrac{\lambda_{\mathrm{t}}-\lambda_{\mathrm{nt}}}{\frac{\partial F_1}{\partial c_{1L}}}\end{bmatrix}. \tag{11}$$

The eigenvectors correspond to the possible paths a solution can trace in the phase space. The eigenvector $\vec{e}_t$ corresponds to the straight line paths in the phase space, known as the *tie-line* paths (Figure 7). The eigenvector $\vec{e}_{nt}$ corresponds to the curved paths in the phase space, known as the *nontie-line* paths. Within the two-phase region only certain specific volume fractions of liquid and vapor phase (c_{pL} and c_{pV}) can be in equilibrium, and the tie-lines connect these equilibrium volume fractions. So, on a given tie-line, c_{pV} and c_{pL} remain constant for all the components. Tie-lines also connect the vapor locus and the liquid locus in the phase space. The point at which a tie-line intersects the vapor locus has $S_V = 1$ and $S_L = 0$; the point where it intersects the liquid locus has $S_V = 0$ and $S_L = 1$. Nontie-line paths connect the various tie-lines. When the eigenvalues $\lambda_t = \lambda_{nt}$, the eigenvectors $\vec{e}_t = \vec{e}_{nt}$, the system has dependent eigenvectors, and hence is no longer diagonalizable. That is, the system becomes weakly hyperbolic and any numerical method which depends on the existence of a full set of independent eigenvectors may not perform well at such points. This behavior is only for systems with two phases; with three-phase systems we encounter elliptic regions instead (Juanes & Patzek 2004).

The solution to the ternary problem looks like a combination of two binary displacements. Consider the test case shown in Figure 8, which illustrates the salient features of a ternary problem. There are two key tie-lines, one extending through the initial oil (point **a**), and another extending through the injection gas (point **f**). In the two-component case, the entire displacement occurs on a single tie-line. Hence on each tie-line of the ternary system the solution has, like a binary solution, a shock and a rarefaction. As gas is injected, there are two transitions from the single-phase region to the two-phase region: a leading shock (**a-b**) on the initial tie-line, and a trailing shock (**e-f**) on the injection tie-line. Inside the two-phase region there is a small rarefaction (**b-c**), as the composition varies along the initial tie-line. At the point (**c**), the initial tie-line is tangential to a nontie-line curve. Here, the eigenvalues and the eigenvectors coincide, and the system becomes weakly hyperbolic. The composition then traces the nontie-line path as a rarefaction (**c-d**). At point **d** the solution encounters the injection tie-line, where the velocity jumps from the nontie-line eigenvalue λ_{nt} to the tie-line eigenvalue λ_t. The composition remains constant at **d** for the entire jump, forming a zone of constant state. On the injection tie-line there is one more rarefaction (**d-e**) which connects to the trailing shock (**e-f**).

In practical settings the initial and injection conditions are rarely constant, and the problem has to be solved numerically. In a component-wise numerical method with a constant grid spacing $\Delta x = \Delta x_j = \left(x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}\right)$, and a time-step Δt, we update the solution in the j^{th} cell, at time level *n+1*, according to

1. Set $\mathbf{C}_j^{n+1} = \mathbf{C}_j^n - \frac{\Delta t}{\Delta x}\left[\mathbf{F}_{j+\frac{1}{2}}^n - \mathbf{F}_{j-\frac{1}{2}}^n\right]$, where $\mathbf{C} = \left[C_1, C_2 \ldots C_{n_c}\right]$ and $\mathbf{F} = \left[F_1, F_2 \ldots F_{n_c}\right]$,
2. Iteratively solve the nonlinear set of equations-of-state to obtain S_j^{n+1} (the most expensive step),
3. Compute $f_j^{n+1}, \mathbf{F}_j^{n+1}, \mathbf{F}_{j+\frac{1}{2}}^{n+1}$ using the newly computed S_j^{n+1}.

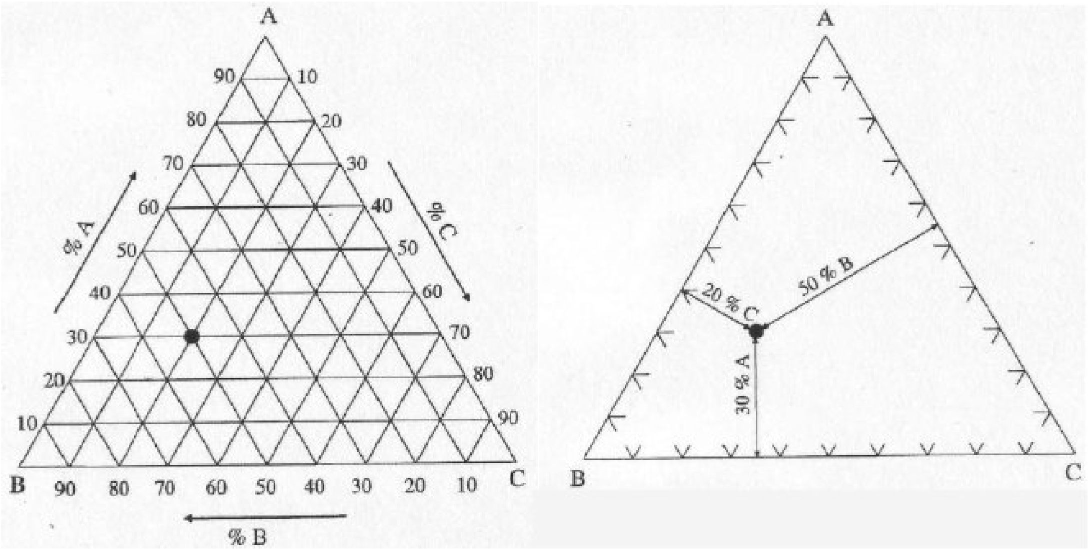


Figure 5: Ternary phase diagram - representation of a ternary composition on an equilateral triangle.

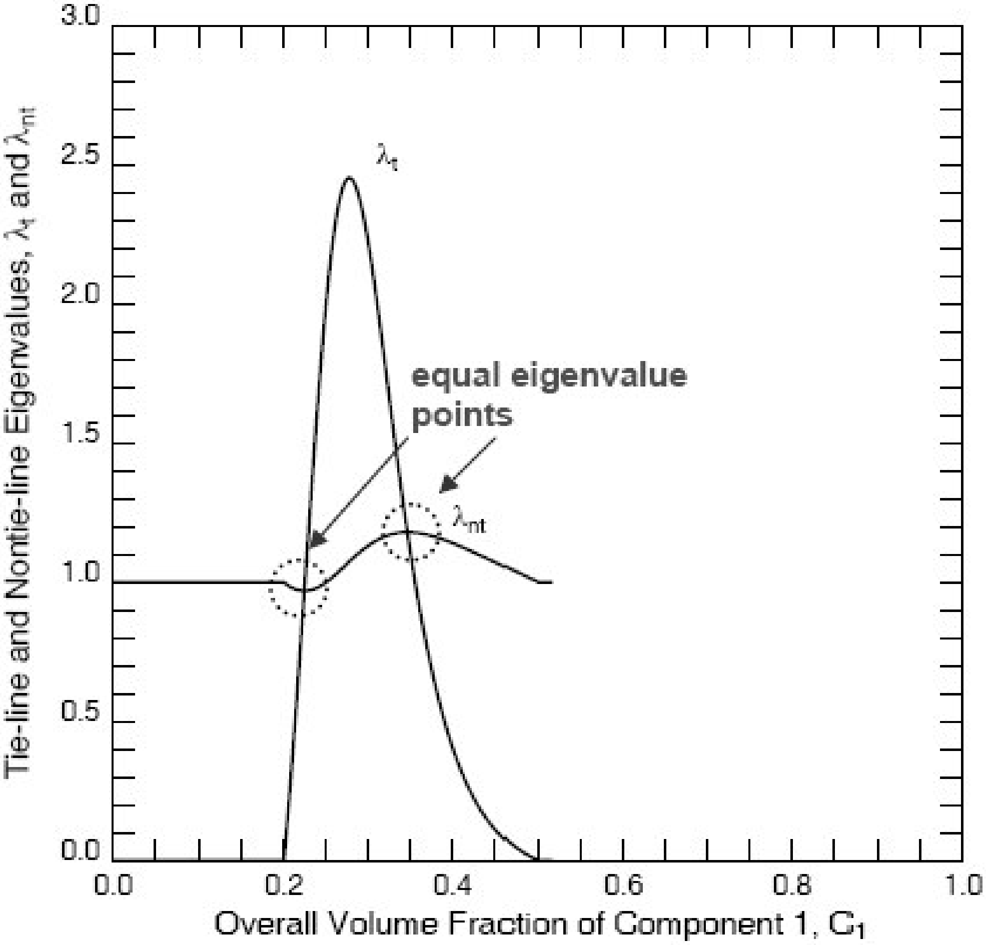


Figure 6: Tieline and non-tieline eigenvalues.

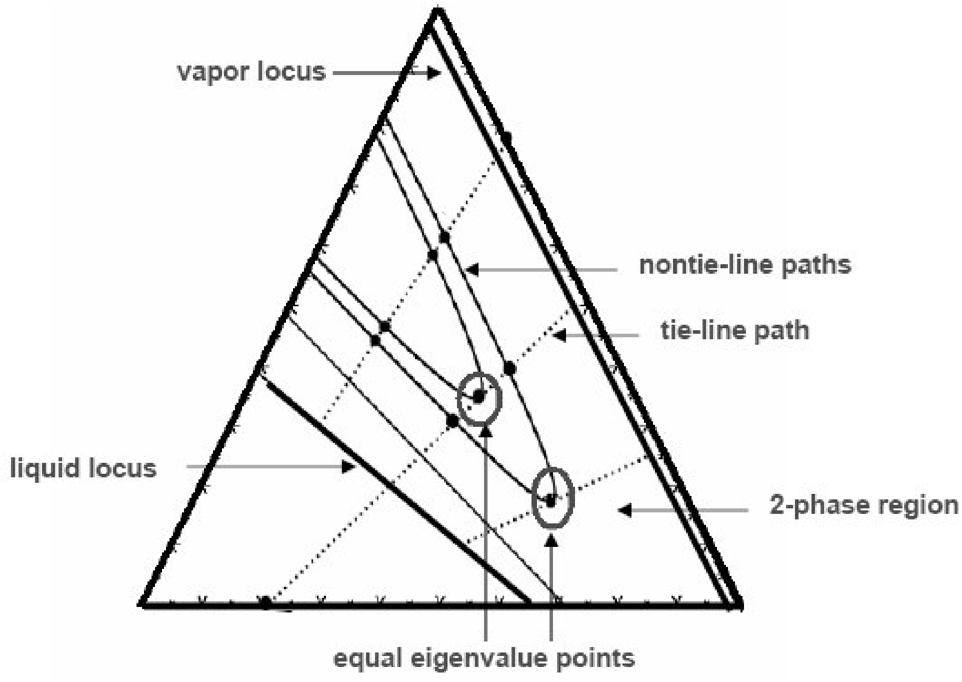


Figure 7: Tie-line and nontie-line paths for a ternary system with constant K-values. At equal eigenvalue points, the nontie-line path is tangent to tie-line path.

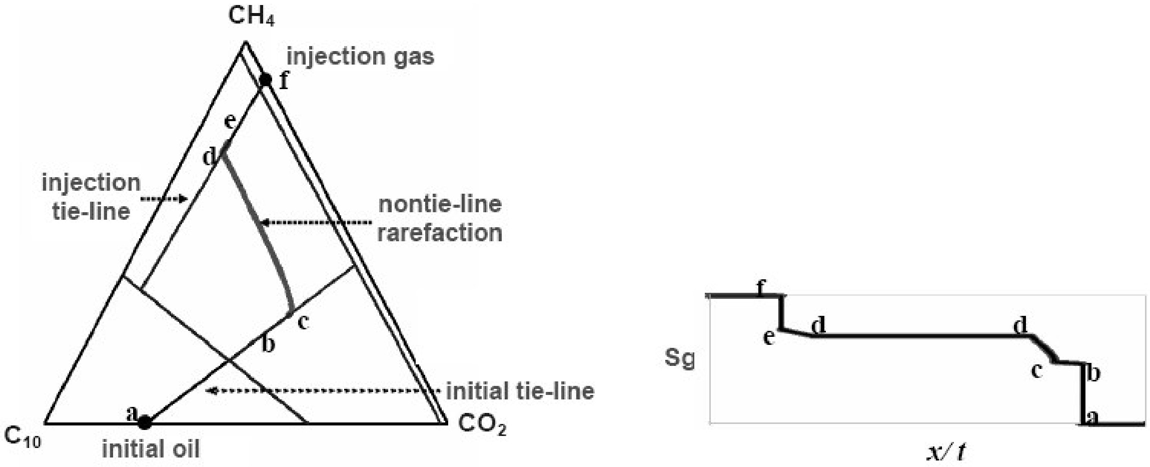


Figure 8: Solution path in the ternary phase diagram and saturation profile of the 3-component displacement.

Figures 9a and 9b show the results of upwind schemes on the two-component displacements. The excessive numerical diffusion of the first order upwind scheme, which is also called the Single Point Upstream weighting scheme (SPU), is more apparent in the leading shock of the second example. SPU requires about 200 grid cells to obtain the same resolution as TVD-RK2 or ENO-RK3 scheme with 40 grid cells. At lower speeds, the numerical diffusion of the upwind schemes decreases. For example 2, where a mixture of components is injected, the trailing shock has zero speed and the upwind schemes resolve the trailing shock with almost no smearing.

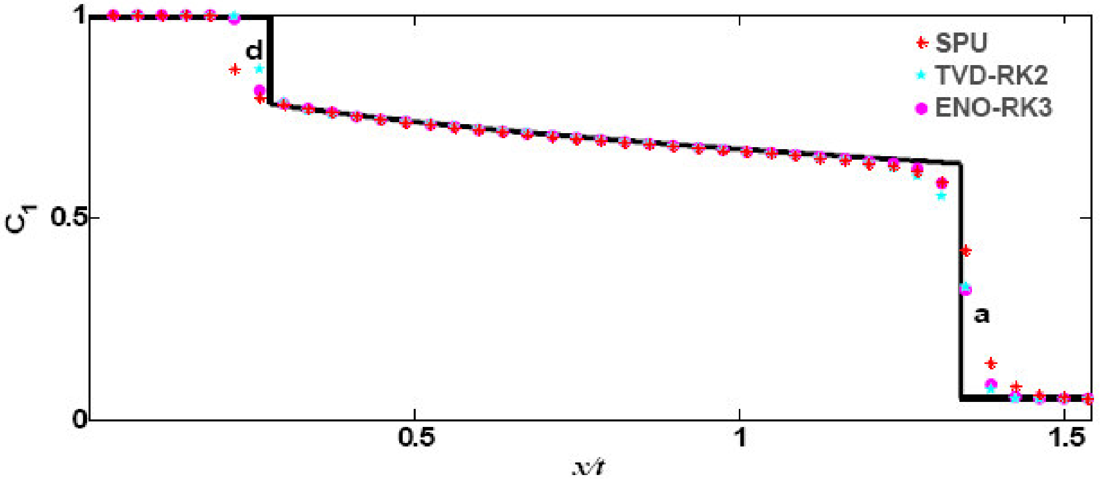


Figure 9a: Results of upwind schemes on test case 1- binary displacement. N= 40, CFL = 0.35.

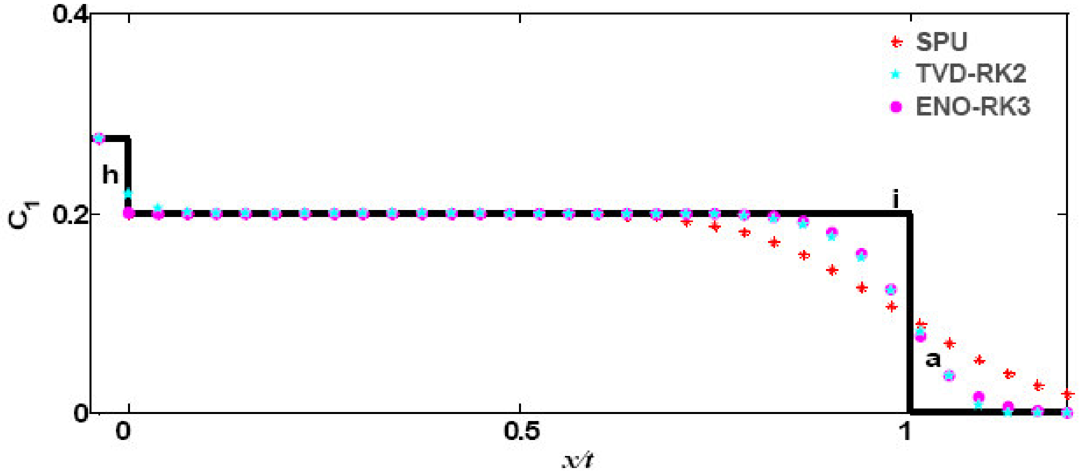


Figure 9b: Results of upwind schemes on test case 2- binary displacement. N= 40, CFL = 0.35. Here, the SPU needs about 200 grid points to resolve as accurately as TVD-RK2 or ENO-RK3.

With the ternary system, the numerical diffusion of the first order scheme not only smoothes the shocks, but also seems to cause errors in the shock speeds. This can be explained by observing the discrepancy between the computed compositional path and the MOC path in the phase space. The numerical diffusion pulls the computed path away from the MOC path, thus altering the speeds of the computed solution. This error mimics the behavior of the system in the presence of physical diffusion. The nonlinear system can also introduce other errors not seen in the scalar two-component problem. For example, the solution may show a dip in the zone of constant state, in other words, too much oil is left behind. This error is frequently seen with nonlinear systems when a discontinuity is present in the initial data and occurs due to the fact that the Hugoniot locus for the nonlinear systems is not a straight line as it is in the scalar case (LeVeque 2002). If MOC is used to compute the solution up to a small time t_1, and this solution is used as the initial state for the numerical scheme, there is a clear reduction in the error at zone of constant state. Though we do not see it in the relatively simple test cases here, component-wise application of upwind schemes may also increase or introduce errors (Qiu & Shu 2002).

Higher order schemes (second order TVD-RK2 and third order ENO-RK3) contain much less numerical diffusion than the first order scheme (Figures 10a and 10b). The shocks are captured more accurately and the error in the zone of constant state is reduced. The solution path in phase space is also closer to the MOC path. Figure 11 gives a closer look at the numerical composition paths in phase space near the equal eigenvalue point. The numerical path computed by the TVD-RK2 scheme jumps from the initial tie-line to an incorrect nontie-line near the equal eigenvalue point. This path is different from the path that would have been traced in the presence of physical diffusion. This error is observed when grids are refined and there is very little numerical diffusion in the scheme. ENO-RK3 behaves better at the equal eigenvalue point.

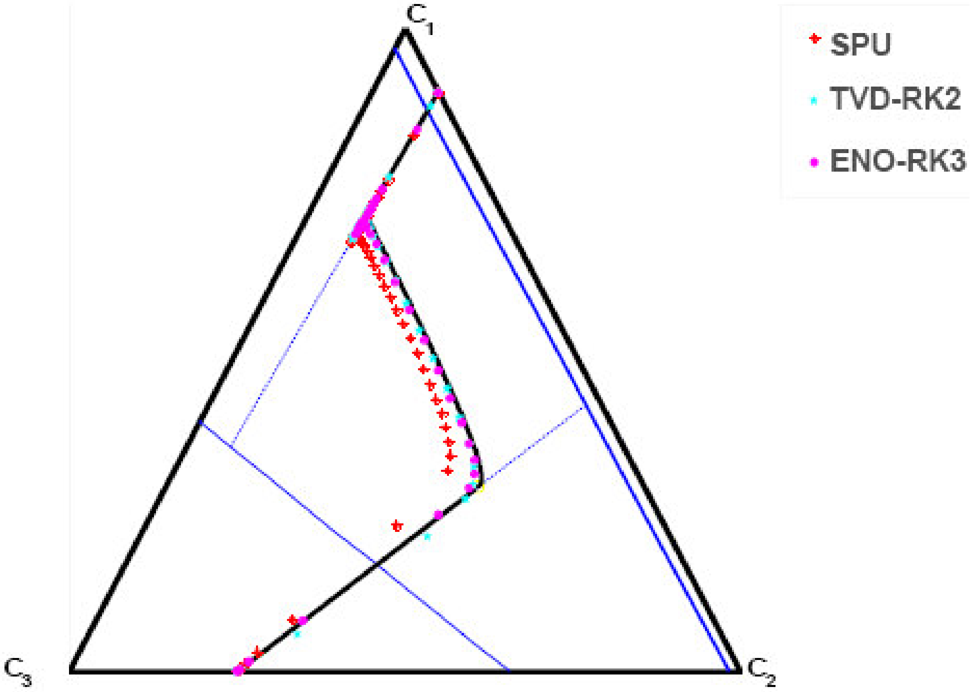


Figure 10a: Results of component-wise application of upwind schemes on ternary displacement in the phase space . SPU is excessively diffusive. N= 80, CFL = 0.4.

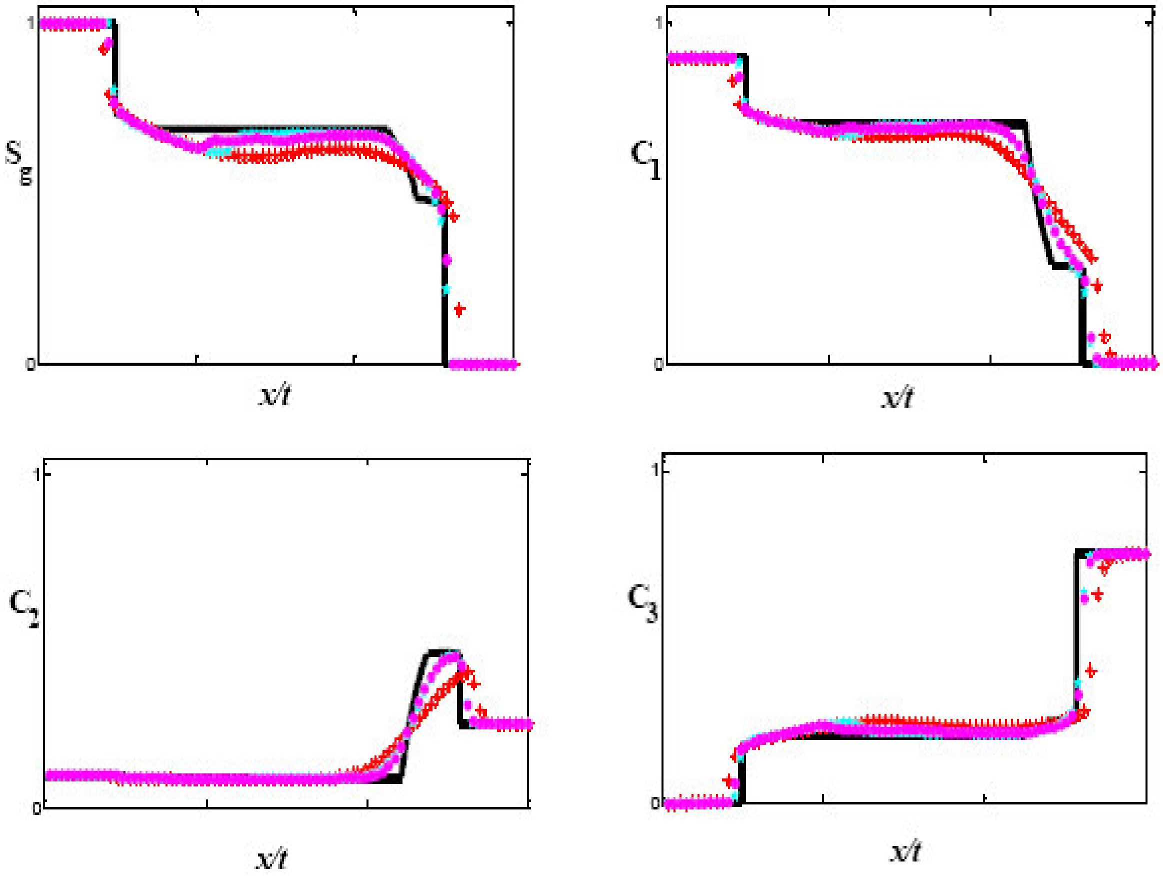


Figure 10b: Solution profiles of ternary displacement got by component-wise application of upwind schemes. SPU smears the shocks and has a prominent dip in the zone-of-constant state. N= 80, CFL = 0.4.

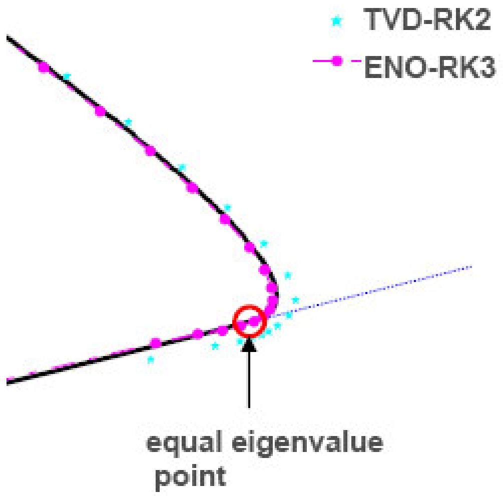


Figure 11: A closer look at the TVD-RK2 and ENO-RK3 composition paths in the phase space, near the equal eigenvalue point. The path computed by the TVD-RK2 scheme jumps to an incorrect nontie-line near the equal eigenvalue point. N= 200, CFL = 0.4.

The improved accuracy of the higher order schemes comes with a price of increased computational cost per time step. For every RK stage introduced in order to maintain a higher temporal accuracy, an additional flash is required. TVD-RK2 requires two flashes per time step and ENO-RK3 requires three flashes per time step. But since the higher order methods allow the usage of coarser grids, these schemes are more efficient than the first order scheme.

3.3 Jin-Xin relaxation

The Jin-Xin relaxation scheme replaces the original system of conservation laws (Equation 4) by

$$\frac{\partial \mathbf{C}}{\partial t} + \frac{\partial \mathbf{V}}{\partial x} = 0, \qquad \mathbf{C},\mathbf{V} \in \mathbf{R}^{n_c-1}, \tag{12}$$

$$\frac{\partial \mathbf{V}}{\partial t} + \mathbf{A}^2 \frac{\partial \mathbf{C}}{\partial x} = \frac{1}{\varepsilon}(\mathbf{F(C)}-\mathbf{V}), \quad \text{with } 0<\varepsilon<<1,$$

$$\mathbf{C} = \left[C_1, C_2 \ldots C_{n_c-1}\right]^{\mathrm{T}}, \text{ and } \mathbf{V} = \left[V_1, V_2 \ldots V_{n_c-1}\right]^{\mathrm{T}},$$

with the initial conditions $\mathbf{C}(x,0) = \begin{cases} \mathbf{C}^J & \text{if } x < 0 \\ \mathbf{C}^I & \text{if } x > 0 \end{cases}$, and $\mathbf{V}(x,0) \;=\; \mathbf{F}(\mathbf{C}(x,0))$.

Here $\mathbf{A}$ is a positive diagonal matrix given by, $\mathbf{A} = diag\left(a_1, a_2 \cdots a_{n_c - 1}\right)$, where the diagonal elements $\left\{a_p\right\}$ are the so-called sub-characteristic speeds. We refer to the parameter ε as the relaxation rate. In the limit $\varepsilon \to 0$, $\mathbf{V} \to \mathbf{F}(\mathbf{C})$ and the relaxation system (Equation 12) reduces to the original conservation law (Equation 4). Using a Chapman-Enskog expansion (Chapman and Cowling 1970) to represent $\mathbf{V}$, the relaxation system can be seen as an approximation to the original conservation law with a small dissipative correction. The Chapman-Enskog expansion

$$\mathbf{V}(x,t) \;=\; \mathbf{F}(\mathbf{C}(x,t)) + \varepsilon\, \mathbf{V}_1(x,t) + \varepsilon^2\, \mathbf{V}_2(x,t) + \varepsilon^3\, \mathbf{V}_3(x,t) + \cdots , \qquad (13)$$

is motivated by the fact that $\mathbf{V} \to \mathbf{F}(\mathbf{C})$ for small ε.

Substituting this expansion for $\mathbf{V}$ in Equation (12), the first order approximation to the relaxation system becomes

$$\frac{\partial \mathbf{C}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{C})}{\partial x} = \varepsilon \frac{\partial}{\partial x}\left((\mathbf{A}^2 - \mathbf{F}'(\mathbf{C})^2) \frac{\partial \mathbf{C}}{\partial x} \right) + \mathrm{O}(\varepsilon^2), \qquad (14)$$

where $\mathbf{F}'(\mathbf{C})$ is the Jacobian of the flux function $\mathbf{F}(\mathbf{C})$.

In Equation (14), $\varepsilon \frac{\partial}{\partial x}\left((\mathbf{A}^2 - \mathbf{F}'(\mathbf{C})^2) \frac{\partial \mathbf{C}}{\partial x} \right)$ is a $\mathrm{O}(\varepsilon)$ dissipative term, with $\varepsilon(\mathbf{A}^2 - \mathbf{F}'(\mathbf{C})^2)$ being the diffusion coefficient matrix. Equation (14) is well-posed only if $(\mathbf{A}^2 - \mathbf{F}'(\mathbf{C})^2)$ is positive semi-definite for all $\mathbf{C}$. This requirement on the diffusion coefficient matrix

$(\mathbf{A}^2 - \mathbf{F}'(\mathbf{C})^2)$ of the relaxation system is called the sub-characteristic condition. In 1D, it is equivalent to

$$\lambda^2 \leq a^2, \quad \text{where } \lambda = \max_{1 \leq p \leq n_C - 1} \left| \lambda_p \right| \quad \text{and} \quad a = \min_{1 \leq p \leq n_C - 1} a_p, \tag{15}$$

where λ_p are the eigenvalues of the Jacobian $\mathbf{F}'(\mathbf{C})$.

For $\mathbf{C}$ in a bounded domain, the sub-characteristic condition can always be satisfied by choosing sufficiently large sub-characteristic speeds. However as we see in the next few paragraphs, the sub-characteristic speeds are the velocities $\left\{\pm a_p\right\}$ of the characteristic variables of the relaxation system. Larger $\left\{a_p\right\}$ imply a reduced CFL number and hence smaller time-steps, i.e. more computational effort. Also, large sub-characteristic speeds increase numerical diffusion. Hence it is desirable to choose the smallest $\left\{a_p\right\}$ that meet the stability criteria (Equation 15). For the Jin-Xin scheme, typically $\left\{a_p\right\}$ is set to λ, the maximum eigenvalue of the Jacobian $\mathbf{F}'(\mathbf{C})$.

The stiff source term introduced by the relaxation formulation can be handled by operator splitting, a simple and a popular technique used in advection-reaction and advection-diffusion problems (LeVeque 2002, Hundsdorfer and Verwer 2003, Younis and Gerritsen 2006). The relaxation system is split into two sub-problems, a homogenous advective system and an ODE system, that can be solved independently.

$$\begin{bmatrix}\mathbf{C}\\ \mathbf{V}\end{bmatrix}_{\mathbf{t}} + \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ \mathbf{A}^2 & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{C}\\ \mathbf{V}\end{bmatrix}_{\mathbf{x}} = \begin{bmatrix}0\\ 0\end{bmatrix}, \qquad \begin{bmatrix}\mathbf{C}\\ \mathbf{V}\end{bmatrix}_{\mathbf{t}} = \begin{bmatrix}0\\ \frac{1}{\varepsilon}(\mathbf{F}(\mathbf{C}) - \mathbf{V})\end{bmatrix}. \tag{16}$$

Homogenous advective system   Stiff system of ODEs

This technique allows the use of high resolution methods for the advective system and standard ODE solvers for the stiff system of ODEs. Though Jin and Xin (1995) solve the stiff ODE system with an implicit RK method, the ODE part in our case can in fact be solved exactly. Using an explicit method would have resulted in very small time steps, dependent on ε. With implicit methods, it would have been necessary to solve a large system of algebraic equations. With operator splitting we avoid both pitfalls. Splitting does introduce an error, and since the source is stiff, naively using either a Godunov or a Strang splitting with a higher order time-stepping introduces an $\mathrm{O}(\Delta t)$ error. A splitting scheme introduced by Jin (1995) ensures the error to be of order $\mathrm{O}(\Delta t^r)$, by interspersing splitting steps with the time-steps of an *r*-stage RK scheme.

The homogenous advective part of the split relaxation system (Equation 16) has the form

$$\begin{bmatrix}\mathbf{C}\\ \mathbf{V}\end{bmatrix}_{\mathbf{t}} + \mathbf{B}\begin{bmatrix}\mathbf{C}\\ \mathbf{V}\end{bmatrix}_{\mathbf{x}} = \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}, \quad \text{where } \mathbf{B} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ \mathbf{A^2} & \mathbf{0}\end{bmatrix}. \tag{17}$$

The matrix **B** has eigenvalues $\left\{ \pm\ a_p \right\}$and eigenvectors of the form $\left[1,0,0\ldots,a_p,0..0\right]^{\mathrm{T}}$ and $\left[1,0,0\ldots,-a_p,0..0\right]^{\mathrm{T}}$. For example, for the scalar conservation law the matrix **B**, the eigenvalues and the right eigenmatrix are given by

$$\mathbf{B} = \begin{bmatrix}0 & 1\\ a^2 & 0\end{bmatrix},\ \Lambda = \begin{bmatrix}a & 0\\ 0 & -a\end{bmatrix},\ \mathbf{R} = \begin{bmatrix}1 & 1\\ a & -a\end{bmatrix} \quad \text{where } \mathbf{B} = \mathbf{R}\,\Lambda\,\mathbf{R}^{-1}.$$

The spatial discretizations in the j^{th} grid cell, for the homogenous system, with a uniform grid spacing $\Delta x = \Delta x_j = \left(x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}\right)$, can be written as

$$\frac{\partial \mathbf{C}_j}{\partial t} + \frac{\left(\mathbf{V}_{j+\frac{1}{2}} - \mathbf{V}_{j-\frac{1}{2}}\right)}{\Delta x_j} = 0, \tag{18a}$$

$$\frac{\partial \mathbf{V}_j}{\partial t} + \mathbf{A}^2 \frac{\left(\mathbf{C}_{j+\frac{1}{2}} - \mathbf{C}_{j-\frac{1}{2}}\right)}{\Delta x_j} = 0. \tag{18b}$$

To get the point values $\mathbf{C}_{j+\frac{1}{2}}$, $\mathbf{V}_{j+\frac{1}{2}}$, whose upwind direction cannot be determined directly, we use a characteristic decomposition on the constant relaxation matrix $\mathbf{B}$. The advective system can be diagonalized as

$$\mathbf{R}^{-1} \begin{bmatrix} \mathbf{C} \\ \mathbf{V} \end{bmatrix}_{\mathbf{t}} + \mathbf{R}^{-1}\, \mathbf{B} \begin{bmatrix} \mathbf{C} \\ \mathbf{V} \end{bmatrix}_{\mathbf{x}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{19}$$

$$\mathbf{W_t} + \Lambda\, \mathbf{W_x} = \mathbf{0}.$$

The vector $\mathbf{W}$ is made up of the characteristic variables $\left\{\mathrm{W}_p^{\pm} = \mathrm{V}_p \pm a_p \mathrm{C}_p\right\}$ that travel with speeds $\left\{\pm\, a_p\right\}$, respectively. While the original system has waves moving only from left to right, in the relaxation system, information flows in both directions. The characteristic variables $\left\{\mathrm{W}_p^{\pm} = \mathrm{V}_p \pm a_p \mathrm{C}_p\right\}$ can be solved by upwind schemes to obtain $\mathbf{C}_{j+\frac{1}{2}}$ and $\mathbf{V}_{j+\frac{1}{2}}$. For example, with a first order upwind scheme the point values $\mathbf{C}_{j+\frac{1}{2}}$, $\mathbf{V}_{j+\frac{1}{2}}$ are obtained by

$$(\mathbf{V} + \mathbf{AC})_{j+1/2} = (\mathbf{V} + \mathbf{AC})_j, \qquad (\mathbf{V} - \mathbf{AC})_{j+1/2} = (\mathbf{V} - \mathbf{AC})_{j+1}, \tag{20}$$

$$\mathbf{C}_{j+1/2} = \frac{1}{2}\left(\mathbf{C}_j + \mathbf{C}_{j+1}\right) - \frac{1}{2}\mathbf{A}^{-1}\left(\mathbf{V}_{j+1} - \mathbf{V}_j\right), \tag{21a}$$

$$\mathbf{V}_{j+1/2} = \frac{1}{2}\left(\mathbf{V}_j + \mathbf{V}_{j+1}\right) - \frac{1}{2}\mathbf{A}\left(\mathbf{C}_{j+1} - \mathbf{C}_j\right). \tag{21b}$$

Using Equation (21), the first order semi-discrete approximation can be written as

$$\frac{\partial \mathbf{C}_j}{\partial t} + \frac{1}{2\Delta \mathrm{x}_j}\left(\mathbf{V}_{j+1} - \mathbf{V}_{j-1}\right) - \frac{1}{2\Delta \mathrm{x}_j}\mathbf{A}\left(\mathbf{C}_{j+1} - 2\mathbf{C}_j + \mathbf{C}_{j-1}\right) = 0, \qquad (22a)$$

$$\frac{\partial \mathbf{V}_j}{\partial t} + \frac{1}{2\Delta \mathrm{x}_j}\mathbf{A}^2\left(\mathbf{C}_{j+1} - \mathbf{C}_{j-1}\right) - \frac{1}{2\Delta \mathrm{x}_j}\mathbf{A}\left(\mathbf{V}_{j+1} - 2\mathbf{V}_j + \mathbf{V}_{j-1}\right) = 0. \qquad (22b)$$

Using forward Euler for time integration Equation (22a) is similar to the Lax-Friedrichs scheme. If we choose the sub-characteristic speeds $\{a_p = a\}$ such that $a(\Delta t/\Delta x) = 1$, this scheme collapses to the Lax-Friedrichs scheme.

The algorithm for the first order scheme is given by the following steps

1. Obtain $\begin{bmatrix}\mathbf{C}^* \\ \mathbf{V}^*\end{bmatrix}$ by solving analytically the ODE system $\begin{bmatrix}\mathbf{C}^n \\ \mathbf{V}^n\end{bmatrix}_t = \begin{bmatrix}0 \\ \frac{1}{\varepsilon}(\mathbf{F}(\mathbf{C}^n) - \mathbf{V}^n)\end{bmatrix}$,
2. Solve the homogenous advective system

$$\mathbf{C}^{n+1} = \mathbf{C}^* - \frac{\Delta t}{\Delta x_j}\left(\mathbf{V}^*_{j+\frac{1}{2}} - \mathbf{V}^*_{j-\frac{1}{2}}\right),$$
$$\mathbf{V}^{n+1} = \mathbf{V}^* - \frac{\Delta t}{\Delta x_j}\mathbf{A}^2\left(\mathbf{C}^*_{j+\frac{1}{2}} - \mathbf{C}^*_{j-\frac{1}{2}}\right),$$

using $\mathbf{V}^*_{j+\frac{1}{2}}$ and $\mathbf{C}^*_{j+\frac{1}{2}}$ as given in (21).

Jin and Xin present a second order scheme by using Van-Leer's MUSCL (Van Leer 1979) scheme for the spatial discretization and a RK-2 time stepping. Instead of using a piecewise constant reconstruction as in the first order method, the MUSCL scheme uses a piecewise linear function. An adaptive limiter controls the linear function used based on the upwind direction. Here the characteristic variables are updated as

$$\left(\mathbf{V}_p + a_p\mathbf{C}_p\right)_{j+1/2} = \left(\mathbf{V}_p + a_p\mathbf{C}_p\right)_j + \frac{1}{2}\Delta x_j \sigma_j^+ \qquad (23)$$
$$\left(\mathbf{V}_p - a_p\mathbf{C}_p\right)_{j+1/2} = \left(\mathbf{V}_p - a_p\mathbf{C}_p\right)_{j+1} - \frac{1}{2}\Delta x_{j+1}\sigma_{j+1}^- ,$$

where

$$\sigma_j^{\pm} = \frac{1}{\Delta x_j}\Big(\big(\mathbf{V}_p \pm a_p\mathbf{C}_p\big)_{j+1} - \big(\mathbf{V}_p \pm a_p\mathbf{C}_p\big)_j\Big)\phi(\theta_j^{\pm}). \qquad (24a)$$

Here $\phi(\theta)$ is the limiter. A popular choice is the van Leer limiter $\phi(\theta) = \frac{|\theta| + \theta}{|\theta| + 1}$. θ is a measure of smoothness given by

$$\theta_j^{\pm} = \frac{\big(\mathbf{V}_p \pm a_p\mathbf{C}_p\big)_j - \big(\mathbf{V}_p \pm a_p\mathbf{C}_p\big)_{j-1}}{\big(\mathbf{V}_p \pm a_p\mathbf{C}_p\big)_{j+1} - \big(\mathbf{V}_p \pm a_p\mathbf{C}_p\big)_j}. \qquad (24b)$$

Some other common high-resolution limiters are the minmod and superbee limiters (LeVeque 2002). The minmod limiter is known to be diffusive and will lead to smeared shocks. The superbee limiter resolves the shocks with little smearing but also tends to steepen and square off smooth variations. The van Leer limiter which resolves shocks well, but does not artificially steepen the smooth slopes is usually is a good choice for a wide class of problems, including ours.

Using the above, the algorithm for the second order scheme, with Jin's splitting, can be formulated as

1. First RK- Stage: Obtain $\begin{bmatrix}\mathbf{C}^*\\ \mathbf{V}^*\end{bmatrix}$ by solving analytically the ODE system $\begin{bmatrix}\mathbf{C}^n\\ \mathbf{V}^n\end{bmatrix}_t = \begin{bmatrix}0\\ \frac{1}{\varepsilon}(\mathbf{F}(\mathbf{C}^n) - \mathbf{V}^n)\end{bmatrix}$,
2. First RK- Stage: Solve the homogenous advective system using the MUSCL scheme

$$\mathbf{C}^1 = \mathbf{C}^* - \frac{\Delta t}{\Delta x_j}\Big(\mathbf{V}^*_{j+\frac{1}{2}} - \mathbf{V}^*_{j-\frac{1}{2}}\Big),$$

$$\mathbf{V}^1 = \mathbf{V}^* - \frac{\Delta t}{\Delta x_j}\mathbf{A}^2\Big(\mathbf{C}^*_{j+\frac{1}{2}} - \mathbf{C}^*_{j-\frac{1}{2}}\Big),$$

where $\mathbf{V}^*_{j+\frac{1}{2}}$ and $\mathbf{C}^*_{j+\frac{1}{2}}$ are obtained using given in (23) and (24),

3. Second RK- Stage: Obtain $\begin{bmatrix} \mathbf{C}^{**} \\ \mathbf{V}^{**} \end{bmatrix}$ by solving the ODE system

$$\begin{bmatrix} \mathbf{C}^1 \\ \mathbf{V}^1 \end{bmatrix}_t = \begin{bmatrix} 0 \\ \frac{1}{\varepsilon}(\mathbf{F}(\mathbf{C}^1) - \mathbf{V}^1) \end{bmatrix},$$

4. Second RK- Stage: Solve the homogenous advective system

$$\mathbf{C}^2 = \mathbf{C}^{**} - \frac{\Delta t}{\Delta x_j}\left(\mathbf{V}^{**}_{j+\frac{1}{2}} - \mathbf{V}^{**}_{j-\frac{1}{2}}\right),$$
$$\mathbf{V}^2 = \mathbf{V}^{**} - \frac{\Delta t}{\Delta x_j}\mathbf{A}^2\left(\mathbf{C}^{**}_{j+\frac{1}{2}} - \mathbf{C}^{**}_{j-\frac{1}{2}}\right),$$

where $\mathbf{V}^{**}_{j+\frac{1}{2}}$ and $\mathbf{C}^{**}_{j+\frac{1}{2}}$ are obtained using given in (23) and (24),

5. Take the RK-average

$$\begin{bmatrix} \mathbf{C}^{n+1} \\ \mathbf{V}^{n+1} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} (\mathbf{C}^n + \mathbf{C}^2) \\ (\mathbf{V}^n + \mathbf{V}^2) \end{bmatrix}.$$

The modified equation (27) for the relaxation scheme indicates that the numerical diffusion can be reduced if the sub-characteristic speed a is varied to closely track $f'(u)$. There are two possible ways to construct a relaxation system with varying sub-characteristic. One is derived using a non-conservative form of the equations, the other using a conservative formulation, written as:

Non-conservative form

$$\frac{\partial \mathbf{C}}{\partial t} + \frac{\partial \mathbf{V}}{\partial x} = 0,$$

$$\frac{\partial \mathbf{V}}{\partial t} + \mathbf{A}(x,t)^2\frac{\partial \mathbf{C}}{\partial x} = \frac{1}{\varepsilon}(\mathbf{F}(\mathbf{C}) - \mathbf{V}),$$

Conservative form

$$\frac{\partial \mathbf{C}}{\partial t} + \frac{\partial \mathbf{V}}{\partial x} = 0,$$

$$\frac{\partial \mathbf{V}}{\partial t} + \frac{\partial}{\partial x}\left(\mathbf{A}(x,t)^2\mathbf{C}\right) = \frac{1}{\varepsilon}(\mathbf{F}(\mathbf{C}) - \mathbf{V}),$$

written compactly as,

$$\mathbf{q}_t + \mathbf{B}(x,t)\mathbf{q}_x = \mathbf{s} \quad ,$$

or

$$\mathbf{q}_t + (\mathbf{B}(x,t)\mathbf{q})_x = \mathbf{s}\,,$$

where $\mathbf{q} = \begin{bmatrix} \mathbf{C} \\ \mathbf{V} \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{A}^2 & \mathbf{0} \end{bmatrix}$ as before.

For simplicity we use the non-conservative form. Of course, we can go back and forth between the forms at the expense of adding source terms. Since we have a stiff source term and we use operator splitting, using either form does not affect our numerical calculations.

3.4 Analysis and numerical solution of the variable relaxation scheme

Using a Chapman-Enskog expansion to represent $\mathbf{V}$, the first order approximation to the variable relaxation system becomes

$$\frac{\partial \mathbf{C}}{\partial t} + \frac{\partial \mathbf{F(C)}}{\partial x} = \varepsilon \frac{\partial}{\partial x}\left((\mathbf{A}(x,t)^2 \text{ - } \mathbf{F}'(\mathbf{C})^2)\frac{\partial \mathbf{C}}{\partial x}\right) + \mathrm{O}(\varepsilon^2)\,. \tag{28}$$

Again the term $\varepsilon(\mathbf{A}(x,t)^2 \text{ - } \mathbf{F}'(\mathbf{C})^2)$ plays the role of diffusion coefficient matrix. Equation (28) is well-posed only if $(\mathbf{A}(x,t)^2 \text{ - } \mathbf{F}'(\mathbf{C})^2)$ is positive semi-definite for all **C**; this defines the sub-characteristic condition for the variable relaxation system. In 1D, this requirement reduces to

$$\lambda(x,t)^2 \le a(x,t)^2\,, \tag{29}$$

where $\lambda(x,t) = \max\limits_{1\le p\le n_C-1} \left|\lambda_p(x,t)\right|$, $a(x,t) = \min\limits_{1\le p\le n_C-1} a_p(x,t)$,

and $\lambda_p(x,t)$ are the local eigenvalues of the Jacobian $\mathbf{F}'(\mathbf{C})$.

The local eigenvalues of the Jacobian $\mathbf{F}'(\mathbf{C})$ represent either the rarefaction speeds or the shock speeds. The sub-characteristic speeds must be chosen so that the positive semi-definiteness of the diffusion coefficient matrix is guaranteed in either case (Liu 1987). In the Jin-Xin relaxation this was done by setting the sub-characteristic speed $a = \max\limits_{1\le p\le n_C-1} \left|\lambda_p(x,t)\right|, \forall \left(x,t\right)$. For the gas displacement example, the tie-line eigenvalue is given by $\dfrac{\partial F_1}{\partial C_1}$ (Equation 10a). The nontie-line eigenvalues hover around 1 and an estimate of the maximum nontie-line eigenvalue can be obtained (Dindoruk 1982). The sub-characteristic speeds are then chosen as

$$a_p(x,t) \ge \max \ \{ \ \frac{\partial F_1}{\partial C_1}(x,t), \frac{F_1(x+\frac{\Delta x}{2},t)-F_1(x-\frac{\Delta x}{2},t)}{C_1(x+\frac{\Delta x}{2},t)-C_1(x-\frac{\Delta x}{2},t)}, \max_{(x,t)} \lambda_{\mathrm{NTL}}(x,t)\}. \qquad (30)$$

*W*e use operator splitting where the relaxation system is split into two sub-problems, a homogenous advective system and an ODE system. The only difference here lies in the way the homogenous system is solved.

In the Jin-Xin relaxation system the matrix $\mathbf{B}$ is diagonalizable and the values of C_p and V_p at the interfaces $\left\{x_{j+\frac{1}{2}}\right\}$ are obtained by using the characteristic variables $\left\{\mathrm{W}_p^{\pm} = \mathrm{V}_p \pm a_p \mathrm{C}_p\right\}$. With the variable system, we cannot diagonalize the homogenous system as easily as before. The eigenmatrix $\mathbf{R}$ of this system depends on its eigenvalues. When the eigenvalues are variable in *(x,t)* the eigenmatrix also becomes variable in (x,t), and $\mathbf{B}(x,t) = \mathbf{R}(x,t)\Lambda(x,t)\mathbf{R}^{-1}(x,t)$. For example, for the 2-component displacement (the scalar conservation law), we have

$$\mathbf{B}(x,t) = \begin{bmatrix} 0 & 1 \\ a(x,t)^2 & 0 \end{bmatrix}, \quad \Lambda(x,t) = \begin{bmatrix} a(x,t) & 0 \\ 0 & -a(x,t) \end{bmatrix},$$
$$\mathbf{R}(\text{x,t}) = \begin{bmatrix} 1 & -1 \\ a(x,t) & a(x,t) \end{bmatrix}.$$

Multiplying the homogeneous advection system $\mathbf{q}_t + \mathbf{B}(x,t)\mathbf{q}_x = 0$ with $\mathbf{R}^{-1}$**,** we get

$$\mathbf{R}^{-1}(x,t)\mathbf{q}_t + \mathbf{R}^{-1}(x,t)\mathbf{B}(x,t)\mathbf{q}_x = \mathbf{0}.$$

This can be written as

$$\mathbf{w}_t + \Lambda(x,t)\,\mathbf{w}_x = \left(\mathbf{R}_t^{-1}(x,t) + \Lambda(x,t)\mathbf{R}_x^{-1}(x,t)\right)\mathbf{R}(x,t)\,\mathbf{w},$$

where $\mathbf{w} = \mathbf{R}^{-1}(x,t)\mathbf{q}$, because

$$\mathbf{R}^{-1}(x,t)\mathbf{q}_t = (\mathbf{R}^{-1}(x,t)\mathbf{q})_t - \mathbf{R}_t^{-1}(x,t)\mathbf{q} \text{ and } \mathbf{R}^{-1}(x,t)\mathbf{q}_x = (\mathbf{R}^{-1}(x,t)\mathbf{q})_x - \mathbf{R}_x^{-1}(x,t)\mathbf{q}.$$

The advection equations are now coupled together by the source terms, and the left and the right going waves do not propagate independently. Rather than working with this system, we rather view the system as a set of Riemann problems at the interfaces $\left\{x_{j+\frac{1}{2}}\right\}$ (LeVeque 2002) that can be easily solved numerically.

We define

$$\mathbf{q}_j^n = \left[\mathbf{C}_j^n \;\; \mathbf{V}_j^n\right]^{\mathrm{T}} = \left[C_{1,j}^n\;, C_{2,j}^n \;\mathrm{K}\; C_{n_c-1,j}^n,\; V_{1,j}^n\,, V_{2,j}^n \;\mathrm{K}\; V_{n_c-1,j}^n\right]^{\mathrm{T}},$$
$$\Delta\mathbf{q}_{j\text{-}\frac{1}{2}}^n = \mathbf{q}_j^n - \mathbf{q}_{j\text{-}1}^n \qquad .$$

Also, we define the right eigenmatrix $\mathbf{R}_{\text{j-}\frac{1}{2}} = [\mathbf{r}_{\text{j-}\frac{1}{2}}^1 \;\; \mathbf{r}_{\text{j-}\frac{1}{2}}^2 \;\; \mathrm{K} \;\; \mathbf{r}_{\text{j-}\frac{1}{2}}^{2(\mathrm{n_c}\text{-}1)}]$ and the left eigenmatrix

$$\mathbf{L}_{j\text{-}\frac{1}{2}} = [\left(\mathrm{l}_{j\text{-}\frac{1}{2}}^1\right)^{\mathrm{T}} \;\; \left(\mathrm{l}_{j\text{-}\frac{1}{2}}^2\right)^{\mathrm{T}} \;\; \mathrm{K} \;\; \left(\mathrm{l}_{j\text{-}\frac{1}{2}}^{2(\mathrm{n_c}\text{-}1)}\right)^{\mathrm{T}}].$$

We set $\Lambda^{+}_{j\text{-}\frac{1}{2}}$ and $\Lambda^{-}_{j\text{-}\frac{1}{2}}$ such that $\mathbf{B}_{j\text{-}\frac{1}{2}} = \mathbf{R}_{j\text{-}\frac{1}{2}}\ \Lambda_{j\text{-}\frac{1}{2}}\mathbf{L}_{j\text{-}\frac{1}{2}}\ = \mathbf{R}_{j\text{-}\frac{1}{2}}\ \left(\Lambda^{+}_{j\text{-}\frac{1}{2}} + \Lambda^{-}_{j\text{-}\frac{1}{2}}\right)\mathbf{L}_{j\text{-}\frac{1}{2}}$,

$\mathbf{B}^{+}_{j\text{-}\frac{1}{2}}\ =\ \mathbf{R}_{j\text{-}\frac{1}{2}}\Lambda^{+}_{j\text{-}\frac{1}{2}}\mathbf{L}_{j\text{-}\frac{1}{2}}$, and $\mathbf{B}^{-}_{j\text{-}\frac{1}{2}}\ =\ \mathbf{R}_{j\text{-}\frac{1}{2}}\Lambda^{-}_{j\text{-}\frac{1}{2}}\mathbf{L}_{j\text{-}\frac{1}{2}}$, where $\Lambda^{+}_{j\text{-}\frac{1}{2}} = diag\left(\lambda^{+}_{1}, \lambda^{+}_{2} \ldots\ \lambda^{+}_{2(\mathrm{n_c}-1)}\right)$

with $\lambda^{+}_{k} = \max(\lambda_k, 0)$, and $\Lambda^{-}_{j\text{-}\frac{1}{2}} = diag\left(\lambda^{-}_{1}, \lambda^{-}_{2} \ldots\ \lambda^{-}_{2(\mathrm{n_c}-1)}\right)$ with $\lambda^{-}_{k} = \min(\lambda_k, 0)$.

We write the update at every time step as

$$\mathcal{D}_{+}\mathbf{q}^{n}_{j} = \frac{1}{\Delta x}(\mathbf{q}^{n}_{j+\frac{1}{2}} - \mathbf{q}^{n}_{j-\frac{1}{2}}) \quad = \quad \frac{1}{\Delta x}(\mathbf{B}^{+}_{j\text{-}\frac{1}{2}}\Delta\mathbf{q}^{n}_{j\text{-}\frac{1}{2}} + \mathbf{B}^{-}_{j+\frac{1}{2}}\Delta\mathbf{q}^{n}_{j+\frac{1}{2}}). \tag{31}$$

The first order variable relaxation method can now be constructed as

1. Choose $a_p(x_j, t_n)$ as given in Equation (30) so as to satisfy the sub-characteristic condition,

2. Obtain $\begin{bmatrix}\mathbf{C}^{*}\\ \mathbf{V}^{*}\end{bmatrix}$ by solving analytically the ODE system $\begin{bmatrix}\mathbf{C}^{n}\\ \mathbf{V}^{n}\end{bmatrix}_t\ =\ \begin{bmatrix}0\\ \frac{1}{\varepsilon}(\mathbf{F}(\mathbf{C}^{n}) - \mathbf{V}^{n})\end{bmatrix}$ as

   $\mathbf{C}^{*}_{j} = \mathbf{C}^{n}_{j}$

   $\mathbf{V}^{*}_{j} = F(\mathbf{C}^{*}_{j})\left(1 - e^{-\Delta t/\varepsilon}\right) + \mathbf{V}^{n}_{j}e^{-\Delta t/\varepsilon}, \qquad \mathbf{q}^{*}_{j} = \begin{bmatrix}\mathbf{C}^{*}_{j}\\ \mathbf{V}^{*}_{j}\end{bmatrix}$

3. Solve the homogenous advective system

   $\mathbf{q}^{n+1}_{j} = \mathbf{q}^{*}_{j}\ - \Delta t\ \mathcal{D}_{+}\mathbf{q}^{*}_{j}$ where $\mathcal{D}_{+}\mathbf{q}^{*}_{\mathrm{j}}$ is the update as given in Equation (31).

If implemented in matrix form, as given above, the method can be computationally expensive. But the right and left eigenvectors have a simple analytical form, which can be exploited to arrive at updates to individual components, in a manner similar to Jin-Xin updates. The right eigenvectors corresponding to the p^{th} component have the form,

$$\mathbf{r}^{k}_{j-\frac{1}{2}} = \Big[0 \ldots \ \underset{\substack{\uparrow \\ \\ p^{\text{th}}\text{ element}}}{1} \ 0 \cdots \ 0 \ \underset{\substack{\uparrow \\ \\ (n_c+p-1)^{\text{th}}\text{ element}}}{a_{p,j}} \ 0 \cdots \ 0 \Big]^{\mathrm{T}},$$

$$\mathbf{r}^{k+1}_{j-\frac{1}{2}} = \Big[0 \ldots \ \underset{\substack{\uparrow \\ \\ p^{\text{th}}\text{ element}}}{1} \ 0 \cdots \ 0 \ \underset{\substack{\uparrow \\ \\ (n_c+p-1)^{\text{th}}\text{ element}}}{\text{-}a_{p,j-1}} \ 0 \cdots \ 0 \Big]^{\mathrm{T}},$$

and the left eigenvectors corresponding to the p^{th} component are of the form

$$\ell^{k}_{j-\frac{1}{2}} = \left[0 \ldots \ \underset{\substack{\uparrow \\ \\ p^{\text{th}}\text{ element}}}{\frac{a_{p,j-1}}{a_{p,j}+a_{p,j-1}}} \ 0 \cdots \ 0 \ \underset{\substack{\uparrow \\ \\ (n_c+p-1)^{\text{th}}\text{ element}}}{\frac{1}{a_{p,j}+a_{p,j-1}}} \ 0 \cdots \ 0 \right],$$

$$\ell^{k+1}_{j-\frac{1}{2}} = \left[0 \ldots \ \underset{\substack{\uparrow \\ \\ p^{\text{th}}\text{ element}}}{\frac{a_{p,j}}{a_{p,j}+a_{p,j-1}}} \ 0 \cdots \ 0 \ \underset{\substack{\uparrow \\ \\ (n_c+p-1)^{\text{th}}\text{ element}}}{\frac{-1}{a_{p,j}+a_{p,j-1}}} \ 0 \cdots \ 0 \right].$$

Here k is related to p as $k = 2p-1$**.** Using this information, the update for the p^{th} component of $\mathbf{C}$ and $\mathbf{V}$**,** in the 3rd step will be

$$\begin{aligned}
C^{n+1}_{p,j} = C^{n}_{p,j} - \frac{\Delta t}{\Delta x}\left(\frac{a_{p,j}a_{p,j-1}}{a_{p,j}+a_{p,j-1}}\left(C^{n}_{p,j}-C^{n}_{p,j-1}\right) + \frac{-a_{p,j+1}a_{p,j}}{a_{p,j+1}+a_{p,j}}\left(C^{n}_{p,j+1}-C^{n}_{p,j}\right) \right) \\
- \frac{\Delta t}{\Delta x}\left(\frac{a_{p,j}}{a_{p,j}+a_{p,j-1}}\left(V^{n}_{p,j}-V^{n}_{p,j-1}\right) + \frac{a_{p,j}}{a_{p,j+1}+a_{p,j}}\left(V^{n}_{p,j+1}-V^{n}_{p,j}\right) \right),
\end{aligned}$$

$$\begin{aligned}
V^{n+1}_{p,j} = V^{n}_{p,j} - \frac{\Delta t}{\Delta x}\left(\frac{(a_{p,j})^2 a_{p,j-1}}{a_{p,j}+a_{p,j-1}}\left(C^{n}_{p,j}-C^{n}_{p,j-1}\right) + \frac{a_{p,j+1}(a_{p,j})^2}{a_{p,j+1}+a_{p,j}}\left(C^{n}_{p,j+1}-C^{n}_{p,j}\right) \right) \\
- \frac{\Delta t}{\Delta x}\left(\frac{(a_{p,j})^2}{a_{p,j}+a_{p,j-1}}\left(V^{n}_{p,i}-V^{n}_{p,i-1}\right) - \frac{(a_{p,j})^2}{a_{p,j+1}+a_{p,j}}\left(V^{n}_{p,j+1}-V^{n}_{p,j}\right) \right).
\end{aligned}$$

3.5 High resolution variable relaxation

A second order high resolution method can be obtained by extending the first order method with appropriate wave limiters as before. But, in this case, the limiters, which depend on the characteristic variables and the magnitude of change in the characteristic waves, must be constructed using the left and right eigenvectors as follows

- Find the change in each of the *2(n_c-1)* characteristic variables using the corresponding left eigenvectors as $\alpha^k_{j-\frac{1}{2}} = \ell^k_{j-\frac{1}{2}} \bullet \Delta\mathbf{q}^n_{j-\frac{1}{2}}$ and the jump in the characteristic waves as $\mathbf{w}^k_{j-\frac{1}{2}} = \alpha^k_{j-\frac{1}{2}}\mathbf{r}^k_{j-\frac{1}{2}}$.

- Set the smoothness parameters of van Leer limiter equal to $\theta_{k,j-\frac{1}{2}} = \dfrac{\mathbf{w}^k_{I-\frac{1}{2}} \bullet \mathbf{w}^k_{j-\frac{1}{2}}}{\mathbf{w}^k_{j-\frac{1}{2}} \bullet \mathbf{w}^k_{j-\frac{1}{2}}}$, where $I = j-1$ for waves moving right (k odd) and $I = j+1$ for waves moving left (k even).

- Compute the second order updates as

$$\mathbf{q}^{n+1}_j = \mathbf{q}^n_j - \frac{\Delta t}{\Delta x}\left(\mathbf{B}^+_{j-\frac{1}{2}}\Delta\mathbf{q}^n_{j-\frac{1}{2}} + \mathbf{B}^-_{j+\frac{1}{2}}\Delta\mathbf{q}^n_{j+\frac{1}{2}}\right) - \frac{\Delta t}{\Delta x}\left(\mathcal{F}^h_{j+\frac{1}{2}} - \mathcal{F}^h_{j-\frac{1}{2}}\right),$$

where the higher order correction terms $\mathcal{F}^h_{j-\frac{1}{2}}$ are given by

$$\mathcal{F}^h_{j-\frac{1}{2}} = \frac{1}{2}\sum_{p=1,2...} a_{p,j}\phi\left(\theta_{k,j-\frac{1}{2}}\right)\mathbf{w}^k_{j-\frac{1}{2}} + a_{p,j-1}\phi\left(\theta_{k+1,j-\frac{1}{2}}\right)\mathbf{w}^{k+1}_{j-\frac{1}{2}}, \quad \text{with } k = 2p-1.$$

We now define the update operator

$$\mathcal{D}_+\mathbf{q}^n_j = \frac{1}{\Delta x}(\mathbf{B}^+_{j-\frac{1}{2}}\Delta\mathbf{q}^n_{j-\frac{1}{2}} + \mathbf{B}^-_{j+\frac{1}{2}}\Delta\mathbf{q}^n_{j+\frac{1}{2}}) \quad + \quad \frac{1}{\Delta x}(\mathcal{F}^h_{j+\frac{1}{2}} \; - \; \mathcal{F}^h_{j-\frac{1}{2}}) \qquad (35)$$

With this, the second order variable relaxation scheme can be given by the following algorithm

1. Choose $a_p(x_j, t_n)$ as given in Equation (30) so as to satisfy the sub-characteristic condition

2. First RK- Stage: Obtain $\begin{bmatrix} \mathbf{C}^* \\ \mathbf{V}^* \end{bmatrix}$ by solving analytically the ODE system

$$\begin{bmatrix} \mathbf{C}^n \\ \mathbf{V}^n \end{bmatrix}_t = \begin{bmatrix} 0 \\ \frac{1}{\varepsilon}(\mathbf{F}(\mathbf{C}^n) - \mathbf{V}^n) \end{bmatrix} \text{ as}$$

$$\mathbf{C}_j^* = \mathbf{C}_j^n$$

$$\mathbf{V}_j^* = F(\mathbf{C}_j^*)\left(1 - e^{-\Delta t/\varepsilon}\right) + \mathbf{V}_j^n e^{-\Delta t/\varepsilon}, \quad \mathbf{q}_j^* = \begin{bmatrix} \mathbf{C}_j^* \\ \mathbf{V}_j^* \end{bmatrix}$$

3. First RK- Stage: Solve the homogenous advective system using the update operator given in Equation (35)

$$\mathbf{q}_j^{(1)} = \mathbf{q}_j^* - \Delta t\ \mathcal{D}_+ \mathbf{q}_j^*, \quad \mathbf{q}_j^{(1)} = \begin{bmatrix} \mathbf{C}_j^{(1)} \\ \mathbf{V}_j^{(1)} \end{bmatrix}$$

4. Second RK-Stage: Obtain $\begin{bmatrix} \mathbf{C}^{**} \\ \mathbf{V}^{**} \end{bmatrix}$ by solving the ODE system

$$\begin{bmatrix} \mathbf{C}^{(1)} \\ \mathbf{V}^{(1)} \end{bmatrix}_t = \begin{bmatrix} 0 \\ \frac{1}{\varepsilon}(\mathbf{F}(\mathbf{C}^{(1)}) - \mathbf{V}^{(1)}) \end{bmatrix} \text{ as}$$

$$\mathbf{C}_j^{**} = \mathbf{C}_j^{(1)}$$

$$\mathbf{V}_j^{**} = F(\mathbf{C}_j^*)(1 - e^{-\Delta t/\varepsilon}) + \mathbf{V}_j^{(1)}\ e^{-\Delta t/\varepsilon}\ , \quad \mathbf{q}_j^{**} = \begin{bmatrix} \mathbf{C}_j^{**} \\ \mathbf{V}_j^{**} \end{bmatrix},$$

5. Second RK-Stage: Solve the homogenous advective system using the update operator given in Equation (35)

$$\mathbf{q}_j^{(2)} = \mathbf{q}_j^{**} - \Delta t\ \mathcal{D}_+ \mathbf{q}_j^{**}, \quad \mathbf{q}_j^{(2)} = \begin{bmatrix} \mathbf{C}_j^{(2)} \\ \mathbf{V}_j^{(2)} \end{bmatrix}$$

6. Take the RK-average

$$\mathbf{q}_j^{n+1} = \frac{1}{2}(\mathbf{q}_j^n + \mathbf{q}_j^{(2)}), \qquad \text{set } \mathbf{q}_j^{n+1} = \begin{bmatrix} \mathbf{C}_j^{n+1} \\ \mathbf{V}_j^{n+1} \end{bmatrix}$$

Again, the update of Equation (35) can be implemented in terms of individual components without matrix-vector operations as

$$\begin{aligned} C_{p,j}^{n+1} = C_{p,j}^n - \; & \frac{\Delta t}{\Delta x}\left(\frac{a_{p,j}a_{p,j-1}}{a_{p,j}+a_{p,j-1}}\Big(C_{p,j}^n - C_{p,j-1}^n\Big) + \frac{-a_{p,j+1}a_{p,j}}{a_{p,j+1}+a_{p,j}}\Big(C_{p,j+1}^n - C_{p,j}^n\Big) \right) \\ & - \frac{\Delta t}{\Delta x}\left(\frac{a_{p,j}}{a_{p,j}+a_{p,j-1}}\Big(V_{p,j}^n - V_{p,j-1}^n\Big) + \frac{a_{p,j}}{a_{p,j+1}+a_{p,j}}\Big(V_{p,j+1}^n - V_{p,j}^n\Big) \right) \\ & - \frac{\Delta t}{\Delta x}\Big(C_{p,j+\frac{1}{2}}^h - C_{p,j-\frac{1}{2}}^h\Big), \end{aligned}$$

where the second order corrections $C_{p,j-\frac{1}{2}}^h$ are computed as

$$\begin{aligned} C_{p,j-\frac{1}{2}}^h = & \frac{a_{p,j}}{2}\phi\Big(\theta_{k,j-\frac{1}{2}}\Big)\left(\frac{a_{p,j-1}}{a_{p,j}+a_{p,j-1}}\Big(C_{p,j}^n - C_{p,j-1}^n\Big) + \frac{1}{a_{p,j}+a_{p,j-1}}\Big(V_{p,j}^n - V_{p,j-1}^n\Big) \right) + \\ & \frac{a_{p,j-1}}{2}\phi\Big(\theta_{k+1,j-\frac{1}{2}}\Big)\left(\frac{a_{p,j}}{a_{p,j}+a_{p,j-1}}\Big(C_{p,j}^n - C_{p,j-1}^n\Big) - \frac{1}{a_{p,j}+a_{p,j-1}}\Big(V_{p,j}^n - V_{p,j-1}^n\Big) \right) . \end{aligned}$$

The corresponding $V_{p,j}^{n+1}$ can be updated as

$$\begin{aligned} V_{p,j}^{n+1} = V_{p,j}^n - & \frac{\Delta t}{\Delta x}\left(\frac{(a_{p,j})^2 a_{p,j-1}}{a_{p,j}+a_{p,j-1}}\Big(C_{p,j}^n - C_{p,j-1}^n\Big) + \frac{a_{p,j+1}(a_{p,j})^2}{a_{p,j+1}+a_{p,j}}\Big(C_{p,j+1}^n - C_{p,j}^n\Big) \right) \\ & - \frac{\Delta t}{\Delta x}\left(\frac{(a_{p,j})^2}{a_{p,j}+a_{p,j-1}}\Big(V_{p,j}^n - V_{p,j-1}^n\Big) - \frac{(a_{p,j})^2}{a_{p,j+1}+a_{p,j}}\Big(V_{p,j+1}^n - V_{p,j}^n\Big) \right) - \frac{\Delta t}{\Delta x}\Big(V_{p,j+\frac{1}{2}}^h - V_{p,j-\frac{1}{2}}^h\Big), \end{aligned}$$

where the second order corrections $V_{p,j-\frac{1}{2}}^h$ are computed as

$$V^{h}_{p,j-\frac{1}{2}} = \frac{a_{p,j}}{2}\phi\left(\theta_{k,j-\frac{1}{2}}\right)\left(\frac{a_{p,j}a_{p,j-1}}{a_{p,j}+a_{p,j-1}}\left(C^{n}_{p,j}-C^{n}_{p,j-1}\right)+\frac{a_{p,j}}{a_{p,j}+a_{p,j-1}}\left(V^{n}_{p,j}-V^{n}_{p,j-1}\right)\right)+$$

$$\frac{a_{p,j-1}}{2}\phi\left(\theta_{k+1,j-\frac{1}{2}}\right)\left(\frac{-a_{p,j}a_{p,j-1}}{a_{p,j}+a_{p,j-1}}\left(C^{n}_{p,j}-C^{n}_{p,j-1}\right)+\frac{a_{p,j-1}}{a_{p,j}+a_{p,j-1}}\left(V^{n}_{p,j}-V^{n}_{p,j-1}\right)\right) \quad .$$

The limiters are computed as

$$\theta_{k,j-\frac{1}{2}} = \frac{\left(\frac{a_{p,j-2}}{a_{p,j-1}+a_{p,j-2}}\left(C^{n}_{p,j-1}-C^{n}_{p,j-2}\right)+\frac{1}{a_{p,j-1}+a_{p,j-2}}\left(V^{n}_{p,j-1}-V^{n}_{p,j-2}\right)\right)\left(1+a_{p,j}a_{p,j-1}\right)}{\left(\frac{a_{p,j-1}}{a_{p,j}+a_{p,j-1}}\left(C^{n}_{p,j}-C^{n}_{p,j-1}\right)+\frac{1}{a_{p,j}+a_{p,j-1}}\left(V^{n}_{p,j}-V^{n}_{p,j-1}\right)\right)\left(1+a_{p,j}a_{p,j}\right)} \quad ,$$

$$\theta_{k+1,j-\frac{1}{2}} = \frac{\left(\frac{a_{p,j+1}}{a_{p,j}+a_{p,j+1}}\left(C^{n}_{p,j+1}-C^{n}_{p,j}\right)-\frac{1}{a_{p,j}+a_{p,j+1}}\left(V^{n}_{p,j+1}-V^{n}_{p,j}\right)\right)\left(1+a_{p,j-1}a_{p,j+1}\right)}{\left(\frac{a_{p,j}}{a_{p,j}+a_{p,j-1}}\left(C^{n}_{p,j}-C^{n}_{p,j-1}\right)-\frac{1}{a_{p,j}+a_{p,j-1}}\left(V^{n}_{p,j}-V^{n}_{p,j-1}\right)\right)\left(1+a_{p,j-1}a_{p,j-1}\right)} \quad .$$

3.6 Experimentation and discussion

Tables 1 and 2 give the comparison of the results of the Jin’s splitting and the Strang splitting with RK-2 scheme on the linear advection equation and Burgers equation with periodic initial data**.** For the linear advection equation, the Strang splitting behaves just as well as Jin splitting. But for the Burgers equation, the order of accuracy of the Strang splitting reduces to $\mathrm{O}(\Delta t)$, while Jin’s splitting maintains $\mathrm{O}\left(\Delta t^{2}\right)$.

TABLE 1

L^1 and L^∞ - rate of convergence for the linear advection equation

$$u_t + u_x = 0, \quad u(x,0) = \sin(x), \text{ CFL} = 0.5$$

S-RK2: 2nd order relaxation scheme with Strang splitting,

J-RK2: 2nd order relaxation scheme with Jin's splitting,

TVD-RK2: 2nd order upwind TVD-RK2 scheme

	L^1 rate of convergence			L^∞ error convergence		
N	S-RK2	J-RK2	TVD-RK2	S-RK2	J-RK2	TVD-RK2
80	2.0155	2.0155	2.0155	1.4120	1.4120	1.4120
160	2.0159	2.0159	2.0159	1.3576	1.3576	1.3576
320	2.0299	2.0299	2.0299	1.3587	1.3587	1.3587
640	2.0277	2.0277	2.0272	1.3213	1.3213	1.3200

TABLE 2

L^1 and L^∞ - rate of convergence for Burgers equation (pre-shock solution)

$u_t + uu_x = 0, \quad u(x,0) = \sin(x)$, CFL = 0.5, T = 0.35

S-RK2: 2nd order relaxation scheme with Strang splitting,

J-RK2: 2nd order relaxation scheme with Jin's splitting,

TVD-RK2: 2nd order upwind TVD-RK2 scheme

	L^1 rate of convergence			L^∞ error convergence		
N	S-RK2	J-RK2	TVD-RK2	S-RK2	J-RK2	TVD-RK2
80	1.0624	1.9230	1.9227	0.7684	1.3972	0.9893
160	1.0162	2.0039	1.9689	0.9083	1.2919	0.9973
320	1.0143	2.0069	1.9787	0.9521	1.3357	0.9993
640	1.0095	2.0183	1.9918	0.9748	1.4094	0.9998

Figure 12 shows the result of first order relaxation schemes on the two-component displacement (Equation 8). Since this is a scalar system we have only two characteristic waves moving with sub-characteristic speeds $\pm a$. By the sub-characteristic condition (Equation 15) the magnitude of the sub-characteristic speed must be at least $\max f'(S)$.

The effect of increasing the sub-characteristic speed can also be seen in Figure 12. It is clear that as the sub-characteristic speed is increased the numerical diffusion increases. While SPU resolves the zero-speed trailing shock sharply for test case 2, the relaxation scheme smears this shock. These behaviors can be better understood by comparing the modified equations for SPU and the first-order relaxation scheme. We performed the analysis on a linear advective system for simplicity. The system is given by

$$u_t \; + \; f(u)_x \;\; = \;\; 0, \text{ with } f(u) = cu, \text{ where } c \text{ is a constant.}$$

The modified equation for SPU is given by

$$u_t \; + \; f(u)_x \quad = \quad \frac{1}{2}(f'\Delta x - \Delta t(f')^2)u_{xx} + O(\Delta x^2),$$

while the modified equation for the first order relaxation scheme is of the form

$$u_t \; + \; f(u)_x \quad = \quad \frac{1}{2}(a\Delta x - \Delta t(f')^2)u_{xx} + O(\Delta x^2).$$

The modified equations show that while the diffusion coefficient of SPU decreases as $f'(u)$ decreases, the diffusion coefficient of the relaxation scheme depends on the sub-characteristic speed a. It increases when a increases, and when $f'(u)$ decreases. When the speed $f'(u)$ tends to zero, the diffusion coefficient of SPU also tends to zero, but the diffusion coefficient of the relaxation scheme tends to $0.5(a\Delta x)$. The diffusion of the relaxation scheme also increases with the relaxation rate ε. However, as shown by Equation (14), this is of magnitude $\mathrm{O}(\varepsilon)$ and since we always take ε very small, this is negligible.

For the ternary systems, the sub-characteristic speed must be decided based on the tieline and non-tieline eigenvalues (Equation 10). The sub-characteristic speeds must be at least $a_p = \max_{\mathrm{x}} \{ \lambda_{\mathrm{TL}}(x), \lambda_{\mathrm{NTL}}(x)\}$. As before, the first order relaxation scheme exhibits more diffusion than SPU. The higher order relaxation scheme has less diffusion and the solution profile (Figure 13b) shows that the results are very close to component-wise upwind TVD-RK2 scheme. However, in phase space (Figures 13a and 14) we see differences in the behavior of the upwind and relaxation schemes. For finer grids, at the equal eigenvalue point the relaxation scheme, unlike the upwind TVD –RK2 scheme, stays close to the MOC path. Along the initial tieline however, the relaxation scheme exhibits slightly more non-physical diffusion. Our goal in the next section is to modify the relaxation scheme to reduce numerical diffusion while retaining strong hyperbolicity.

Figures 15a and 15b show the results of the first order variable relaxation scheme for the two-component displacements (Equation 8). The numerical diffusion of the first order variable relaxation scheme is reduced significantly when the variable sub-characteristic speed is very close to the eigenvalues of the original system. In particular in test case 2, the variable relaxation scheme resolves the zero-speed trailing shock with very little diffusion.

For the ternary systems, the solutions of the first order variable relaxation scheme are very close to those of SPU. The higher order variable relaxation (Figures 16a and 17) scheme has no nonphysical diffusion along the initial tie-line and stays close to the MOC path at the equal eigenvalue point in phase space. In fact, the results are very similar to those obtained with third order ENO-RK3. In general, we have observed that in many cases the second order variable relaxation scheme was competitive with third order component-wise ENO-RK3 reconstruction, and at least as good as the second order component-wise TVD-RK2 with van Leer limiting.

A more general form of the relaxation system can be formulated for the scalar conservation law as

$$\begin{bmatrix} u \\ v \end{bmatrix}_t + \begin{bmatrix} 0 & 1 \\ \alpha & \beta \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}_x = \begin{bmatrix} 0 \\ -\frac{1}{\varepsilon}\left(v - f(u)\right) \end{bmatrix} \tag{39}$$

where $\mathbf{B} = \begin{bmatrix} 0 & 1 \\ \alpha & \beta \end{bmatrix}$ is the coefficient matrix. (40)

Here, $\alpha = -\lambda_1\lambda_2$ and $\beta = \lambda_1 + \lambda_2$ with λ_1, λ_2 being the eigenvalues of the coefficient matrix $\mathbf{B}$.

If $\mathbf{B}$ is constant, we can diagonalize the system as before. The characteristic variables are now $v - \lambda_2 u$ and $v - \lambda_1 u$, moving with speeds λ_1 and λ_2 respectively. Note that, when $\lambda_1 = -\lambda_2 = a$, that is $\alpha = -\lambda_1\lambda_2 = a^2$ and $\beta = \lambda_1 + \lambda_2 = 0$, this system reduces to the Jin-Xin relaxation system and when $\lambda_1 = a$, $\lambda_2 = 0$, it reduces to the upwind scheme of the same order.

The $\mathrm{O}(\varepsilon)$diffusion term, that arises as a result of using the Chapman-Enskog expansion for v, is $\varepsilon\beta(u) = \varepsilon(\lambda_1 - f'(u))(f'(u) - \lambda_2)$. For a system with positive speeds, $0 \le f'(u) \le f'_{\max}$, the sub-characteristic condition requires $\lambda_2 \le 0$ and $\lambda_1 \ge f'_{\max}$. To be as close as possible to the physical flux and yet obey the sub-characteristic condition, we can set $\lambda_1 = f'_{max}$ and $\lambda_2 = -\tau$, where τ is a positive number. Based on the choice of λ_2, the performance of this new scheme is between Jin-Xin relaxation ($\tau = f'_{\max}$) and the upwind scheme ($\tau = 0$). Finding optimal negative eigenvalues that minimize numerical diffusion and yet ensure that the scheme behaves well at the weak hyperbolic points like the Jin-Xin system, is problem dependent. Therefore, the constant version of the general $\alpha - \beta$ relaxation is impractical .

Would a variable $\alpha - \beta$ give improved results? Again looking at the variable $\alpha - \beta$ system for the scalar conservation law, we see that the Chapman-Enskog requires the sub-characteristic condition $\lambda_2 \le f'(u) \le \lambda_1$. One way to reduce numerical diffusion would be to vary both eigenvalues λ_1 and λ_2 such that they are close to the local speeds. For example, we could set

$\lambda_1 = f'(u) + O(\Delta x)$ and $\lambda_2 = f'(u) - O(\Delta x)$. This method however fails because the eigenvectors of the relaxation system are $[1 \;\; \lambda_1]^T$ and $[1 \;\; \lambda_2]^T$, and with $\lambda_1 = f'(u) + O(\Delta x)$ and $\lambda_2 = f'(u) - O(\Delta x)$, the eigenvectors are almost parallel to each other, which re-introduces weak hyperbolicity in the system. A more promising way is to vary only λ_1 so that it is close to the local speeds and keep λ_2 constant and close to $-f'_{max}$. This version is competitive to the variable version of Jin-Xin scheme. Note that for small λ_2 however, we end up with a nearly singular relaxation matrix and the scheme behaves just like the corresponding upwind scheme**.**

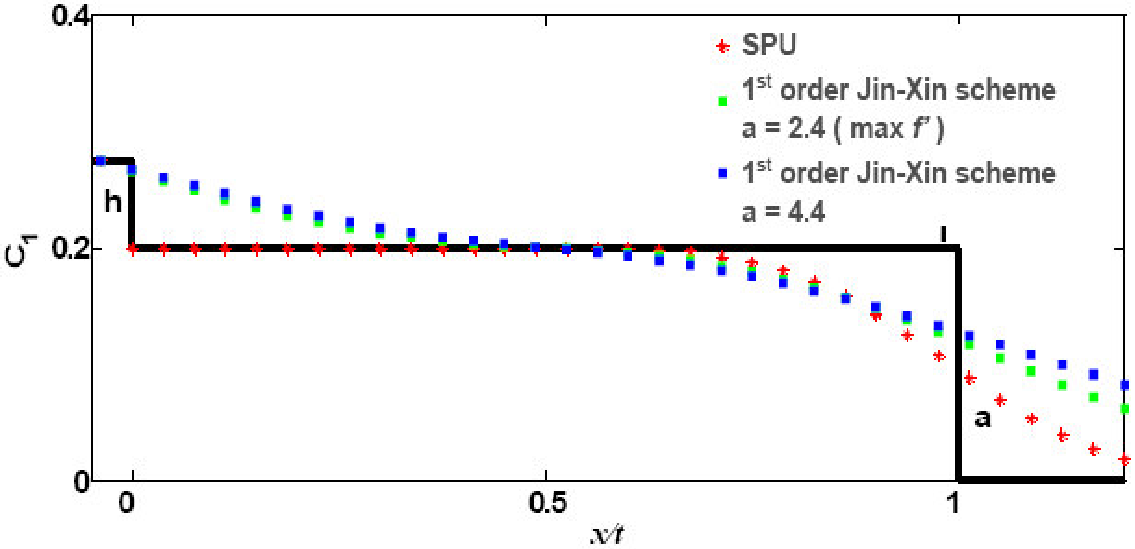


Figure 12: Comparison of 1st order Jin-Xin relaxation scheme and SPU. N= 40, CFL = 0.2.

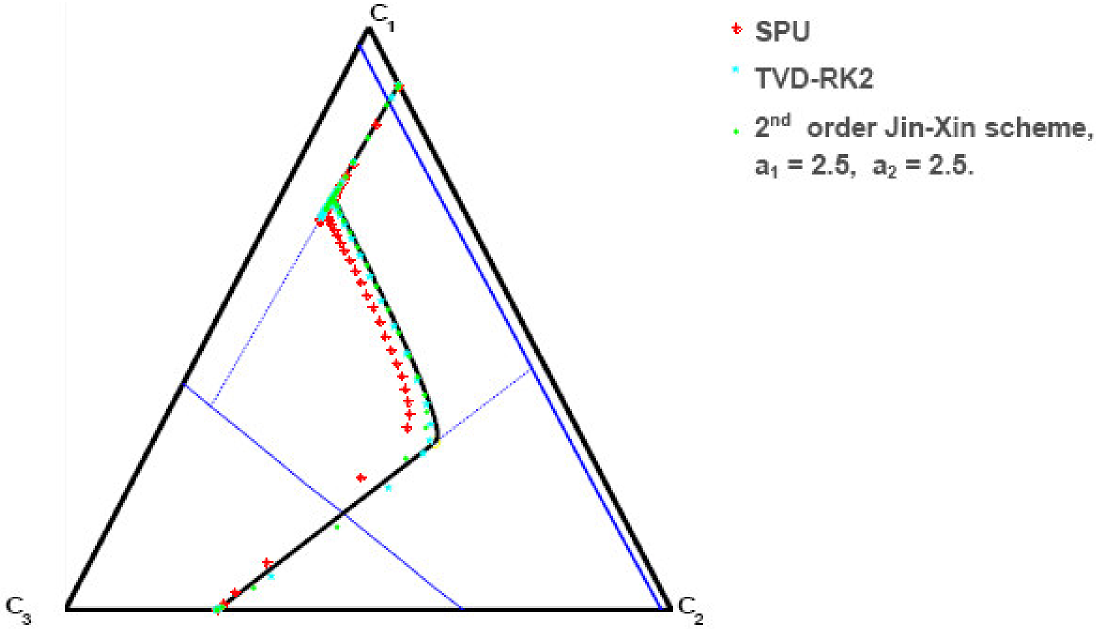


Figure 13a: Phase space comparison of component-wise application of upwind schemes and 2^{nd} order Jin-Xin relaxation scheme on ternary displacement. Composition path of Jin-Xin scheme is close to TVD-RK2 but has slightly more numerical diffusion than TVD-RK2. N= 80, CFL = 0.2.

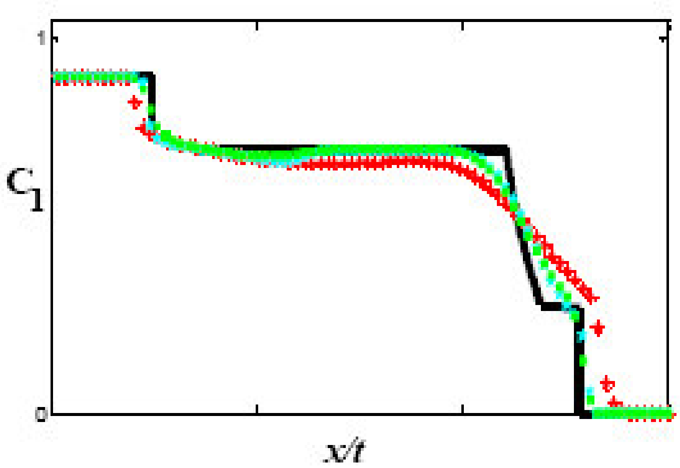


Figure 13b: Solution profile of C_1 obtained by 2^{nd} order Jin-Xin relaxation scheme is close to TVD-RK2. N= 80, CFL = 0.2.

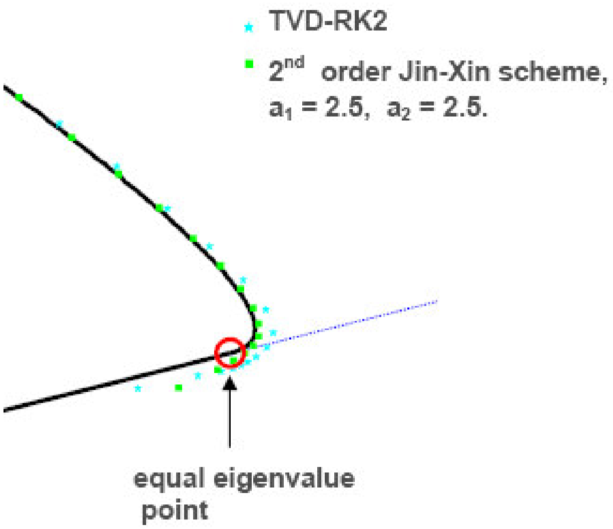


Figure 14: A closer look at the 2^{nd} order Jin-Xin scheme in the phase space near the equal eigenvalue point. The path computed by the Jin-Xin scheme behaves better than TVD-RK2 near the equal eigenvalue point but also has slightly more diffusion on the initial tie-line. N= 200, CFL = 0.2.

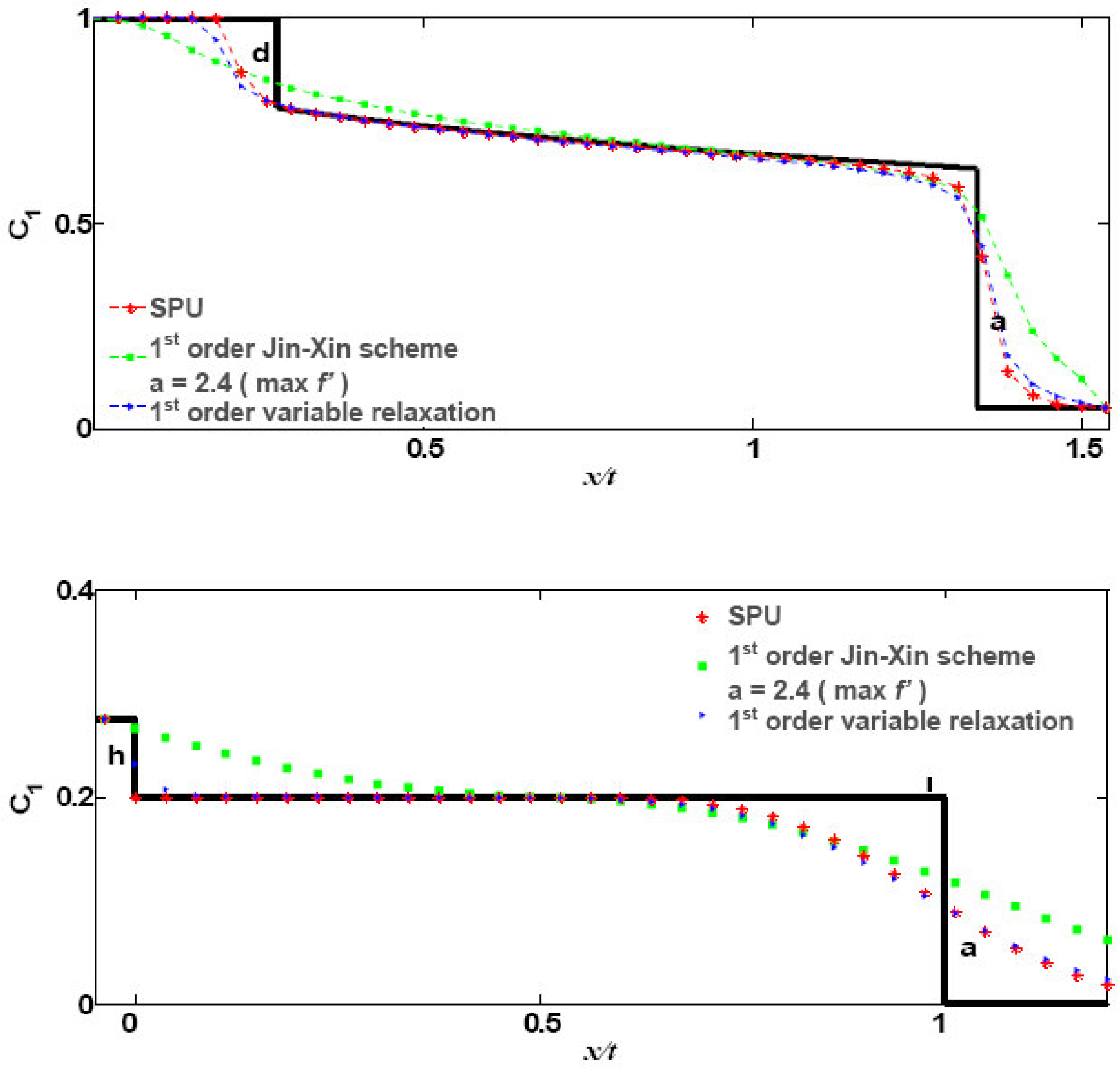


Figure 15a & 15b: Results of variable relaxation scheme on binary displacement. N= 40, CFL = 0.35. Variable relaxation scheme is less diffusive than Jin-Xin scheme and is quite close to SPU.

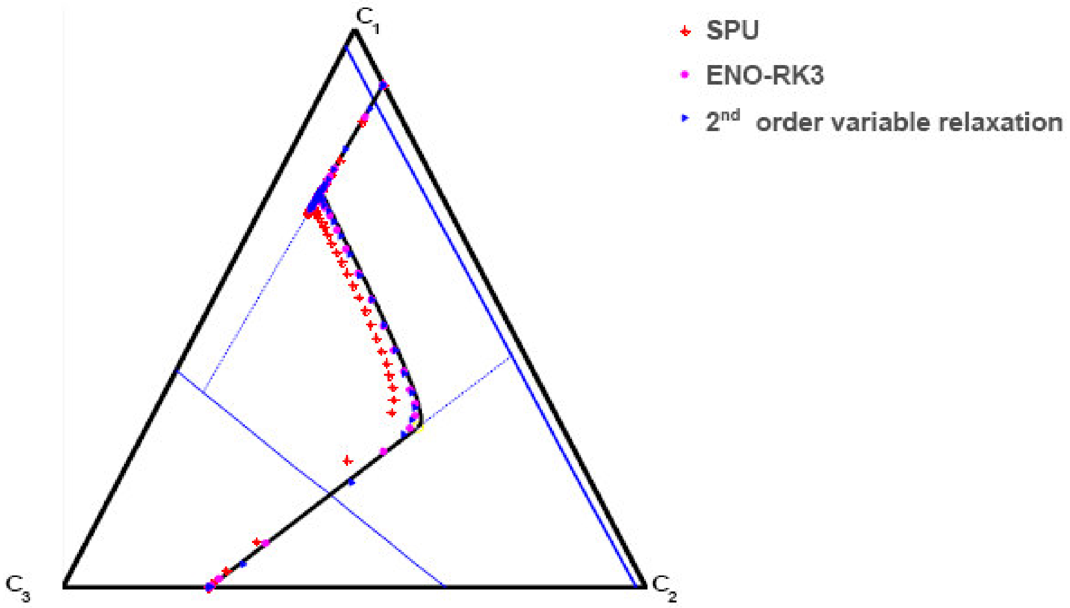


Figure 16a: Phase space comparison of ENO-RK3 and 2nd order variable relaxation scheme on ternary displacement. Composition path of the variable relaxation scheme is close to ENO-RK3. N= 80, CFL = 0.2.

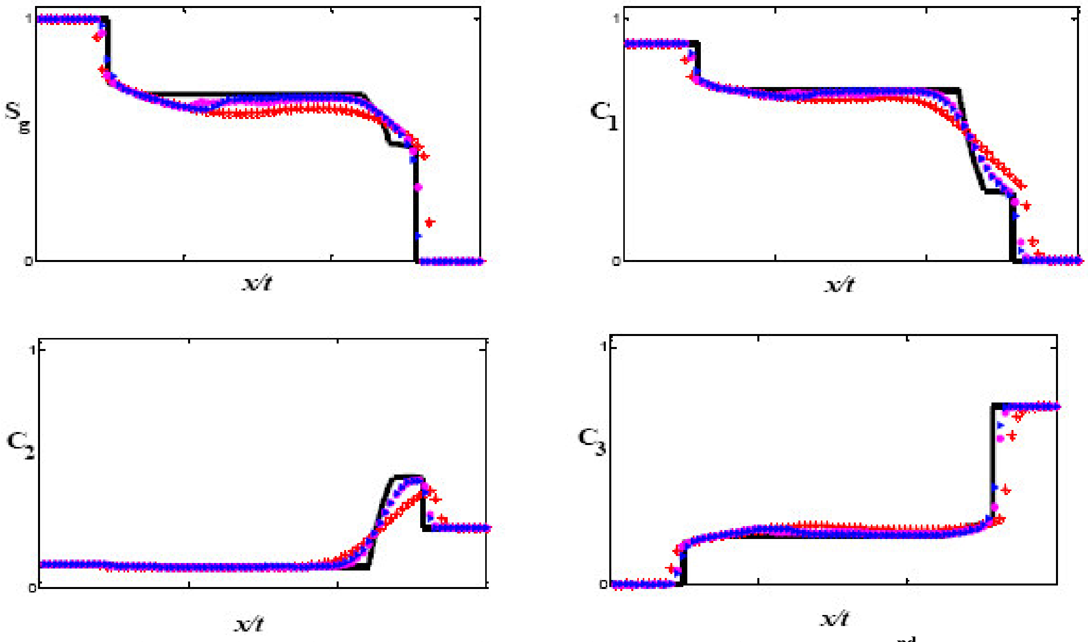


Figure 16b: Comparison of solution profiles obtained by ENO-RK3 and 2nd order variable relaxation. N= 80, CFL = 0.2.

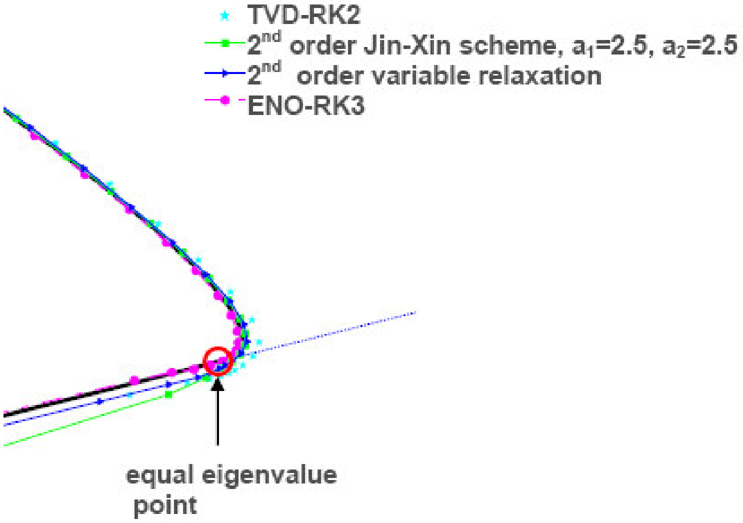


Figure 17: A closer look - 2^{nd} order variable relaxation is close to ENO-RK3 and has much less diffusion than Jin-Xin scheme. It also behaves well near the equal eigenvalue point. N= 200, CFL = 0.2.

4 New compositional solvers: truly multi-dimensional schemes

4.1 Grid orientation effects

In this chapter, we discuss our contributions to the development of compositional solvers on Eulerian grids. With our new scheme, we target a specific challenge in gas injection processes (adverse mobility ratio flows): grid orientation effects.

The main and important motivation for focusing on Eulerian methods is that streamline methods are not suitable for all stages in a gas injection process. This is especially true for the initial injection stages where gravity segregation is an important driver. In an industrial application, it is therefore beneficial to have hybrid capability: a Eulerian solver is used when cross-flow effects, such as gravity segregation, are dominant, and streamline methods when the flow is primarily advection dominated.

The so-called grid orientation effect (GOE) (Todd, 1972) has been an important consideration in many practical reservoir studies involving adverse mobility ratio displacements. Adverse mobility ratio displacements occur when a fluid, such as water, gas or steam, is injected into a reservoir containing less mobile oil with the aim to improve oil production. One of the challenges in simulating such flows is that the physical process, as modeled on the simulation scale, is unstable. As numerical errors are generally not independent of the orientation of the underlying computational grid, solutions to these problems are seen to be highly sensitive to this grid orientation. Though there are many errors made in any simulation project the advective transport errors are amongst the most critical for GOE. One of the primary factors is the grid dependent form of the diffusive numerical error terms.

Extended injection well models can reduce GOE by removing near-well perturbations that are critical in triggering unstable modes in the solution (Kozdon et al, 2008). In this chapter, we seek to better understand the role of numerical diffusion in triggering undesirable modes, and use our improved understanding to design a new upwind discretization scheme for transport that reduces GOE. Our approach distinguishes itself from previous work in that we require our scheme be positive for highly heterogeneous media to prevent nonphysical oscillations. Moreover, we designed our scheme such that only a minimal amount of transverse numerical diffusion is introduced: large enough to reduce GOE, but small enough to allow fingering to be triggered by variations in physical parameters, such as rock permeability and porosity.

We exclusively consider Cartesian grids. Firstly because Cartesian or near-Cartesian grids are used frequently in reservoir simulation, but secondly because we do not want GOE to be an important factor in choosing our gridding strategy as there are many other efficiency and accuracy factors that have to be taken into account in this decision process. We consider conservative finite difference and finite volume methods in this work and restrict our attention to first order upwind schemes. The main results presented here are extensible to any grid topology.

4.2 Modified Equations Analysis

We begin by analyzing upwind finite difference discretizations of the transport equation

$$\frac{\partial c(x,y,t)}{\partial t} + \mathbf{u} \cdot \nabla c(x,y,t) = 0, \tag{1}$$

where $\mathbf{u} = (u,v)^T = \gamma(\cos\theta, \sin\theta)^T$ is constant in space and time, the magnitude of the velocity is $\gamma \equiv \sqrt{u^2 + v^2}$, and the angle of the velocity with respect to the x-coordinate is $\theta \equiv \arctan(u/v)$.

For simplicity, we assume a uniform Cartesian grid with step size h in both spatial directions and nodal grid values $C^n_{i,j} \approx c(x_i \equiv ih, y_j \equiv jh, t^n \equiv n\Delta t)$. Assuming $u, v \geq 0$ the four most physically relevant concentrations surrounding cell (i, j) are $\{C^n_{i,j}, C^n_{i-1,j}, C^n_{i,j-1}, C^n_{i-1,j-1}\}$. Following Roe and Sidilkover (1992), it can be shown that there exists a single parameter family of consistent upwind schemes using these four points in the form

$$C^{n+1}_{i,j} = C^n_{i,j} + \frac{\Delta t}{2h}\left[\begin{array}{c}(-u-v-\kappa)C^n_{i,j} + (u-v+\kappa)C^n_{i-1,j} \\ +(u+v-\kappa)C^n_{i-1,j-1} + (-u+v+\kappa)C^n_{i,j-1}\end{array}\right], \tag{2}$$

where the variable κ is a function of $\mathbf{u}$ and can be understood as defining an interpolation scheme on the four upwind points at the location $(x_i - \Delta tu, y_j - \Delta tv)$. For the scheme to positivite, i.e. the updated solution is a convex combination of the previous solution, we require that

$$\begin{aligned} |u-v| &\leq \kappa \leq u+v, \\ \Delta t \leq & \frac{2h}{u+v+\kappa}. \end{aligned} \tag{3}$$

The expression for κ for four common numerical discretizations and the corresponding time step restrictions are given in Table 1. The Narrow scheme (N-scheme), so called because it depends only on the ``narrow" three points stencil $C_{i,j}$, $C_{i-1,j-1}$ and $C_{i-1,j}$ if $u \geq v$ or $C_{i,j-1}$ if $v \geq u$, has the least restrictive time step according to Equation (3) and Single Point Upwinding (SPU), also known as dimensional upwinding, the most restrictive.

To understand how the errors introduced by the discretization affect the solution we formulate the modified equations and retain the leading order error terms,

$$\begin{aligned} c_t + uc_x + vc_y = & \frac{h}{2}(u - \lambda\cos^2\theta)c_{xx} + \frac{h}{2}(v - \lambda\sin^2\theta)c_{yy} \\ & + \frac{h}{2}(u+v-\kappa-2\lambda\cos\theta\sin\theta)c_{xy} \end{aligned} \tag{4}$$

where $\lambda = \Delta t\gamma/h$.

To interpret the numerically introduced diffusion tensor we transform to the streamline coordinate system denoted by (s,n), where s is the streamline coordinate aligned with the velocity field and n is the coordinate normal to the velocity field. Applying this transform to Equation (4) gives the modified equation in the streamline coordinates

$$\begin{aligned} c_t + \gamma c_s = \quad & \underbrace{\gamma\frac{h}{2}\left[\sin\theta + \cos\theta - \tilde{\kappa}\sin\theta\cos\theta - \lambda\right]}_{d_{ss}} c_{ss} \\ & + \underbrace{\gamma\frac{h}{2}\left[\tilde{\kappa}\sin\theta\cos\theta\right]}_{d_{nn}} c_{nn} \\ & + \underbrace{\gamma\frac{h}{2}\left[\cos\theta - \sin\theta + \tilde{\kappa}(\sin^2\theta - \cos^2\theta)\right]}_{d_{sn}} c_{sn} \end{aligned} \tag{5}$$

where $\tilde{\kappa} = \kappa/\gamma$. The coefficients d_{ss}, d_{nn} can be interpreted as longitudinal and transverse diffusion, respectively, and a nonzero mixed-term d_{sn} indicates that there is some rotation of the tensor relative to the streamline coordinates (Van Ransbeeck et al., 1997). Equation (5) implies that if κ is independent of Δt then the time stepping error only affects the longitudinal diffusion. $\kappa(u,v) = 0$ results in a non-positive scheme with no transverse diffusion. From Equations (3) and (5) it can be seen that the N-scheme has minimal transverse diffusion for the family of positive schemes, and SPU has maximal transverse diffusion.

The above analysis can be extended to general structured grids. For unstructured grids a local analysis can be performed also, but global behavior is likely not predictable due to the changing nature of the error from one grid cell to the next.

4.3 A Family of Positive, First Order Finite Volume Schemes

The analysis to this point has been limited to the constant velocity problems and conservative finite difference schemes. The schemes in the previous section can also be interpreted as finite volume methods. To robustly and more easily extend these methods to general divergence free velocity fields, and later to nonlinear systems, we will stay within the finite volume framework. In this section we show a general family of first order, truly multi-D finite volume schemes based on interaction regions. This family of schemes contains both the single parameter family from above and also a wider class of positive schemes based on the full nine-point nearest neighbors stencil.

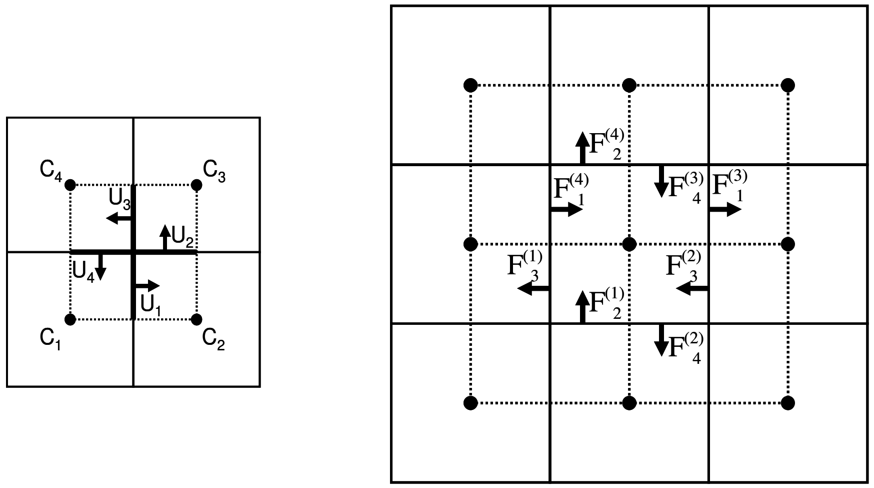
Consider the transport equation for variable velocity in two dimensions, given by

$$c_t + \nabla \cdot (\mathbf{u}c) = 0. \tag{6}$$

Assuming a uniform Cartesian mesh with grid step size h a standard finite volume discretization would be of the form

$$\begin{aligned}
C^{n+1}_{i,j} &= C^n_{i,j} - \frac{\Delta t}{h^2}\left(F^n_{i+1/2,j} - F^n_{i-1/2,j} + F^n_{i,j+1/2} - F^n_{i,j-1/2}\right)\\
C^n_{i,j} &= \frac{1}{h^2}\int_{x_i-h/2}^{x_i+h/2}\int_{y_j-h/2}^{y_j+h/2} c(x,y,t^n)\,dy\,dx\\
F^n_{i+1/2,j} &= \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}}\int_{y_j-h/2}^{y_j+h/2} (uc)\,|_{x=x_i+h/2}\,dy\,dt\\
F^n_{i,j+1/2} &= \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}}\int_{x_i-h/2}^{x_i+h/2} (vc)\,|_{y=y_i+h/2}\,dx\,dt.
\end{aligned} \tag{7}$$

Interaction regions, as commonly used in multipoint flux approximation methods (Aavatsmark, 2002) and control-volume finite-element methods, allow the fluxes to be locally coupled while maintaining positivity. As illustrated in Figure 1 for Cartesian grids, interaction regions form a dual grid whose grid lines connect cell centers and face centers. Such interaction regions can be constructed for any grid topology.



(a) Single interaction region (b) All interaction regions for cell (i,j)

Figure 1: Pictorial view of an interaction region framework. The vectors represent the positive directions.

In the Cartesian case considered in this paper, each interior grid cell (i, j) is part of four interaction regions. We label an interaction region with superscript (m), and number them counterclockwise from the bottom left corner, as illustrated in Figure 1. We denote the four grid cell segments, referred to as sub cells, that make up each interaction region with subscript p. As illustrated in the figure, we number them from $p = 1$ to 4 counter-clockwise starting in the lower left sub cell. Similarly, the half-faces contained in each interaction region are numbered counter-clockwise from $p=1$ to 4 starting with the bottom most half-face. Then, $C_p^{(m)}$ denotes the average scalar concentration of sub cell p in interaction region m, and $U_p^{(m)}$ the volumetric flux through half-face p of interaction region m. In the first order formulations used in this paper, the volumetric flux is held constant over the control volume face. Therefore, each U_p is equal to half the control volume face volumetric flux. With $\mathsf{F}_p^{(m)}$ representing the flux across each half-face p, the update to a control volume (i,j) can be expressed as the following assembly over the four interaction regions of (i,j).

$$C_{i,j}^{n+1} = C_{i,j}^{n} - \frac{\Delta t}{h^2}\left[\mathsf{F}_3^{(1)} - \mathsf{F}_2^{(1)} + \mathsf{F}_4^{(2)} - \mathsf{F}_3^{(2)} + \mathsf{F}_1^{(3)} - \mathsf{F}_4^{(3)} + \mathsf{F}_2^{(4)} - \mathsf{F}_1^{(4)}\right]. \tag{8}$$

We express the fluxes as

$$\mathsf{F}_p^{(m)} = \sum_{q=1}^{4} \alpha_{p,q}^{(m)} C_q^{(m)}. \tag{9}$$

For consistency, we require $\alpha_{p,q}^{(m)}$ to satisfy $\sum_{q=1}^{4} \alpha_{p,q}^{(m)} = U_p^{(m)}$. An attractive feature of the interaction region framework is that the fluxes $\mathsf{F}_p^{(m)}$ depend only on the concentrations within the interaction region, and hence, can be constructed independently for each interaction region in the grid. Then, fluxes can be assembled as is done in Equation (8) for interior Cartesian cells. Similar assemblies can be formulated for other grid topologies. This greatly simplifies the extension of this family of methods to unstructured grids and Cartesian grids with local refinements.

For convenience, we now define $r_{p,q}^{(m)} = \alpha_{p,q}^{(m)} - \alpha_{p-1,q}^{(m)}$ in a cyclic fashion and it can be shown that requiring that $r_{p,p}^{(m)} \geq 0$ and $r_{p,q\neq p}^{(m)} \leq 0$ for all m the scheme is positive. These conditions are sufficient to ensure positivity and are not overly restrictive.

We have not yet discussed how we can assign the weights $\alpha_{p,q}^{(m)}$ in Equation (9). In the following, we will consider an individual interaction region, and therefore drop the superscript (m) for ease of presentation, and also since we will exclusively consider explicit flux expressions, we will drop the superscript indicating the time level also. Consider the subfamily of schemes given by

$$\mathsf{F}_p = U_p \overline{C}_p, \tag{10}$$

where $\overline{C}_p$ denotes the half-face concentration defined as

$$\overline{C}_p = \begin{Bmatrix} (1-\omega_p)C_p & +\omega_p \overline{C}_{p-1} & if U_p \geq 0 \\ (1-\omega_p)C_{p+1} & +\omega_i \overline{C}_{p+1} & if U_p < 0 \end{Bmatrix}, \tag{11}$$

where $p \pm 1$ is defined cyclically on the set $\{1,2,3,4\}$ and ω_p is an interpolation parameter. The scheme can be shown to fit into the above class of positive interaction region based schemes if

$$0 \leq \omega_p \leq \min(1, max(0, U_{uw}/U_p)), \tag{12}$$

where $U_{uw} = U_{p-1}$ if $U_p \geq 0$ and $U_{uw} = U_{p+1}$ if $U_p < 0$.

Assuming that the velocity field is positive and constant this framework is equivalent to the single parameter family of finite difference schemes defined by Equation (2) for $\kappa = u(1-\omega_3) + v(1-\omega_2)$. Hence, any scheme from the single parameter family of finite difference schemes can be represented in the interaction region framework. In Table 2 we present the corresponding ω values for SPU, the N-scheme, and Koren's scheme. For ease of presentation, we defined $w_p^* = \max(0, U_{uw}/U_p)$. We note that if $\omega_p = \omega_p^*$, we obtain the zero transverse diffusion scheme corresponding to $\kappa = 0$, which is not positive. Implementations of the N-scheme and Koren's scheme based on interaction regions have been described previously (Schneider et al, 1986, and Hurtado et al, 2007).

4.4 A Constant Transverse Diffusion Scheme

Riaz and Meiburg (2004) have shown that transverse diffusion has a stabilizing effect on the displacement and suppresses finger formation. This suggests that zero points in the transverse diffusion define the preferential flow angles for a scheme with respect to the grid. An optimal first order scheme might be defined as have zero transverse diffusion for all flow angles however to achieve this it would be necessary to give up the desired positivity of the scheme. However, we can construct a scheme with constant transverse diffusion. Starting with the N-scheme as a base method we add a first order diffusive correction so that the resultant scheme has constant transverse diffusion at the same level as the maximum transverse diffusion value of the N-scheme.

This is a relatively straight forward thing to do for the constant velocity problem, but for the variable velocity interaction region framework care must taken to maintain positivity of the scheme and to not introduce diffusion across faces of cells with low permeability. The resultant scheme is, however, a straightforward extension within the previously defined framework for the N-scheme. Consider half-face p in an interaction region. The advective numerical flux expression for this half-face is

$$\mathsf{F}_p^{adv} = U_p \overline{C}_p = U_p \sum_{q=1}^{4} A_{p,q} C_q, \qquad (13)$$

with $A_{p,q}$ defined by the N-scheme. If $U_{p-1} \geq 0$ and $U_{p+1} \leq 0$, we add to this half-face a first order diffusive flux of the form

$$\mathsf{F}_p^{diff} = \begin{cases} a_p(\overline{C}_{p-1} - \overline{C}_{p+1}) + b_p(C_p \quad - \overline{C}_{p-1}) & U_p \geq 0, U_{p-1} \geq 0 and U_{p+1} < 0 \\ a_p(\overline{C}_{p-1} - \overline{C}_{p+1}) - b_p(C_{p+1} - \overline{C}_{p+1}) & U_p < 0, U_{p-1} \geq 0 and U_{p+1} < 0 \\ 0 & U_{p-1} < 0 or U_{p+1} \geq 0 \end{cases} \qquad (14)$$

The values of $a_p \geq 0$ and $b_p \geq 0$ will depend on U_{p-1}, U_p, and U_{p+1}. Including the half-face concentrations $\overline{C}_{p+1}$ and $\overline{C}_{p-1}$ in the flux expression rather than C_{p-1} and C_{p+2} leads to simpler expressions for (a_p, b_p) that maintain positivity.

In Table 3 we give the values for (a_p, b_p) pairs for half-face p. These values have been chosen so that the scheme is consistent with the finite difference formulation when the velocity field is constant and so that the flux expressions depend on as few cells as possible. Using a harmonic average in the definition of $\widetilde{U}$ introduces less diffusion in cases where the velocity field is rough and avoids smearing across low permeability cells. These pairs of (a_p, b_p) have been formulated in such a manner as not to destroy positivity.

4.5 Experimentation and discussion

As a test problem we present the quarter five-spot problem, which is used throughout the literature to test for grid orientation because it has two natural symmetries and grid-preferred flow directions, see Figure 2.

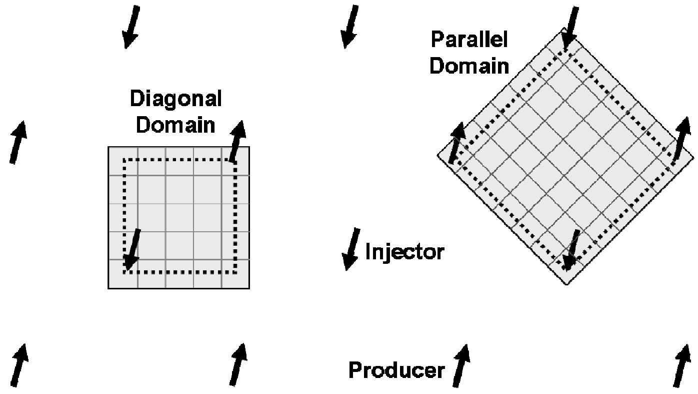


Figure 2: Periodically repeating 5 spot pattern with two quarter-five spot domains.

We model the incompressible, miscible displacement of a resident fluid with viscosity μ_{res} by an injected fluid with viscosity μ_{inj} using the following nonlinear coupled system of equations (Shubin and Bell, 1984)

$$\begin{aligned}
&-\nabla \cdot \frac{K(x,y)}{\mu(c)}\nabla p \equiv \nabla \cdot \mathbf{u} = q, \qquad (x,y,t) \in \Omega \times (0,T),\\
&\varphi \frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{u}c) - \nabla \cdot D(\mathbf{u})\nabla c = \hat{c}q, \qquad (x,y,t) \in \Omega \times (0,T),\\
&c(x,y,0) = c_0(x,y), \qquad (x,y) \in \Omega,\\
&\mathbf{u} \cdot \mathbf{n} = D(\mathbf{u})\nabla c \cdot \mathbf{n}, \qquad (x,y,t) \in \partial\Omega \times (0,T).
\end{aligned} \tag{15}$$

In this formulation p is pressure, K is permeability, φ is porosity, c is the concentration of the injected fluid, q is the well function, $\hat{c}$ is the fluid injected or produced, and D is a tensor that can be used to model physical diffusion and dispersion. We take $D = 0$ since we are assuming that our numerical first order diffusive errors are much larger than any physical diffusion or dispersion at the scales of interest. For the quarter five-spot problem, we use no-flow boundary conditions for the pressure, whereas for the radial displacement problem Dirichlet boundary conditions are used. Details on these boundary conditions are given below. We inject $\hat{c} = 1$ and produce whatever concentration is present at the production well. For the viscosity, $\mu(c)$, we use the common quarter power mixing rule,

$$\mu(c) = (1 - c + M^{1/4} c)^{-4} \mu_{res}, \tag{16}$$

where $M = \mu_{res} / \mu_{inj}$ is the mobility ratio. Equation (15) contains an elliptic pressure equation describing incompressible flow. It is derived using Darcy's law, $\mathbf{u} = -\dfrac{K(x, y)}{\mu(c)} \nabla p$, which relates the velocity field to the pressure gradient. With $D = 0$, Equation (15) also contains a linear hyperbolic transport equation. When the mobility ratio $M > 1$ the problem is said to have an adverse mobility ratio and is theoretically unstable.

We solve the coupled system Equations (15) with an IMPEC strategy (implicit pressure, explicit concentration). At each time step, a two-point flux approximation is used to solve the pressure equation. In the x-direction, we set

$$U_{i+1/2,j} = \frac{hK_{i+1/2,j}}{\mu(C^*_{i+1/2,j})} \frac{(P_{i+1,j} - P_{i,j})}{h}, \tag{17}$$

where $K_{i+1/2,j} = 2/(1/K_{i+1,j} + 1/K_{i,j})$ is the harmonic average of the permeability on either side of the face. The fluid mobility is upwinded in a manner consistent with the transport discretization using the previous flux field and current concentrations, which gives $C^*_{i+1/2,j} = \frac{1}{2}\left(\overline{C}_3^{(2),n} + \overline{C}_1^{(3),n}\right)$. We assume that the fluxes do not change too rapidly between time steps. If this were indeed the case, the IMPEC procedure could instead be iterated until $C^*_{i+1/2,j}$

was consistent with fluxes at the new time level. The fluxes in the y-direction $U_{i,j+1/2}$ are defined in an analogous manner. With these definitions, the linear system of equations

$$U_{i+1/2,j} - U_{i-1/2,j} + U_{i,j+1/2} - U_{i,j-1/2} = \frac{q_{i,j}}{h^2}, \tag{18}$$

is solved first for pressure and then the fluxes are then calculated locally from Equation (18). Here, $q_{i,j} < 0$ for a production well, and $q_{i,j} > 0$ for an injection well. After solving for pressure and the flux field, we advance conentrations to the next time level using one of the transport methods described above

The time step size is constrained to ensure that we do not extract more fluid from a production well block than is available, that is, $\Delta t \leq 1/(h^2 q_{\max})$ where $q_{\max} = \max_{\{i,j\}}(-q_{i,j})$, i.e. the largest production well rate. The time step size is also constrained by the CFL condition for the hyperbolic transport solve.

As mentioned above we first consider the quarter five-spot problem which was presented in Figure 2. There are two principle domains of symmetry for this problem: the diagonal domain in which the mean flow (between the wells) is diagonal to the grid lines, and the parallel domain in which the mean flow is parallel to the grid lines. By symmetry, the domain boundary, indicated by the dotted line in Figure 2, is a no-flow boundary. Care must be taken when handling this boundary condition as improper handling can mask the effects of the numerics. For the diagonal domain the wells are in cells $(1,1)$ and (N,N) and for the parallel domain the wells are in cells $(1,1)$, $(1,N)$, $(N,1)$, and (N,N), as shown in Figure 2. With this placement of the wells the problem is symmetric around the center of the border cells. Therefore, the no-flow boundary is not in fact the outer edge of the domain but rather the line passing through the cell center. This implies that the proper way to handle the boundary is with ghost cells $C_{0,j} = C_{2,j}$,

$C_{N+1,j} = C_{N-1,j}$, $C_{i,0} = C_{i,2}$, and $C_{i,N+1} = C_{i,N-1}$. The same ghost cell framework is used for solving the pressure equation.

In Figure 3 the results for a 50×50 diagonal grid are compared with those for a 71×71 parallel grid with a mobility ratio of $M = 30$. The recovery curves are shown for 1 PVI (pore volume injected) for SPU and the Flat scheme. The displacement fronts are compared at 0.4 PVI for all four schemes. The displacement fronts show that both SPU and Koren's scheme have strongly different solutions on the two grids. On the diagonal grid, the fingers in the SPU solution correspond to the grid directions (0 and $\pi / 2$), which are also the zero points of the transverse diffusion. The zero points for Koren's scheme are also at 0, $\pi / 2$, and as expected we see fingers in these directions. We also notice a finger at $\pi / 4$ which is expected based on the boundary and well conditions. The level of transverse diffusion at $\pi / 4$ is about half that of SPU. The N-scheme and the Flat scheme have more similar solutions on the two grids, with both exhibiting a three-finger structure. These three fingers correspond to the three zero points of the N-scheme at 0, $\pi / 4$, and $\pi / 2$. The fingers in the displacement fronts for the Flat scheme are slightly smaller than for the N-scheme, though finger growth still occurs along the same directions. This is likely due to the fact that the Flat scheme is a modification to the N-scheme and the diffusion added is purposely kept low so as not to diffuse into no-flow grid blocks. The breakthrough times are much closer for the N-scheme and the Flat scheme than for SPU and Koren's scheme. Overall, the Flat scheme shows the least grid dependency.

Common Name	κ	Interpolation	Time Step
Single Point Upwinding (SPU)	$u+v$	Linear interpolation on $\{C^n_{i,j}, C^n_{i-1,j}, C^n_{i,j-1}\}$	$\Delta t \leq h/(u+v)$
Narrow Scheme (N-Scheme)	$\lvert u-v \rvert$	Linear interpolation on $\{C^n_{i,j}, C^n_{i-1,j}, C^n_{i-1,j-1}\}$ if $u > v$ and $\{C^n_{i,j}, C^n_{i,j-1}, C^n_{i-1,j-1}\}$ if $v > u$	$\Delta t \leq h/\max(u,v)$
Corner Transport Upwinding (CTU)	$u+v-2uv\Delta t/h$	Bilinear interpolation on $\{C^n_{i,j}, C^n_{i-1,j}, C^n_{i,j-1}, C^n_{i-1,j-1}\}$	$\Delta t \leq h/\max(u,v)$
Koren's Scheme	$(u+v)/(u^2+uv+v^2)$	Other first order interpolation $\{C^n_{i,j}, C^n_{i-1,j}, C^n_{i,j-1}, C^n_{i-1,j-1}\}$	$\Delta t \leq h(u+v)/(u^2+uv+v^2)$

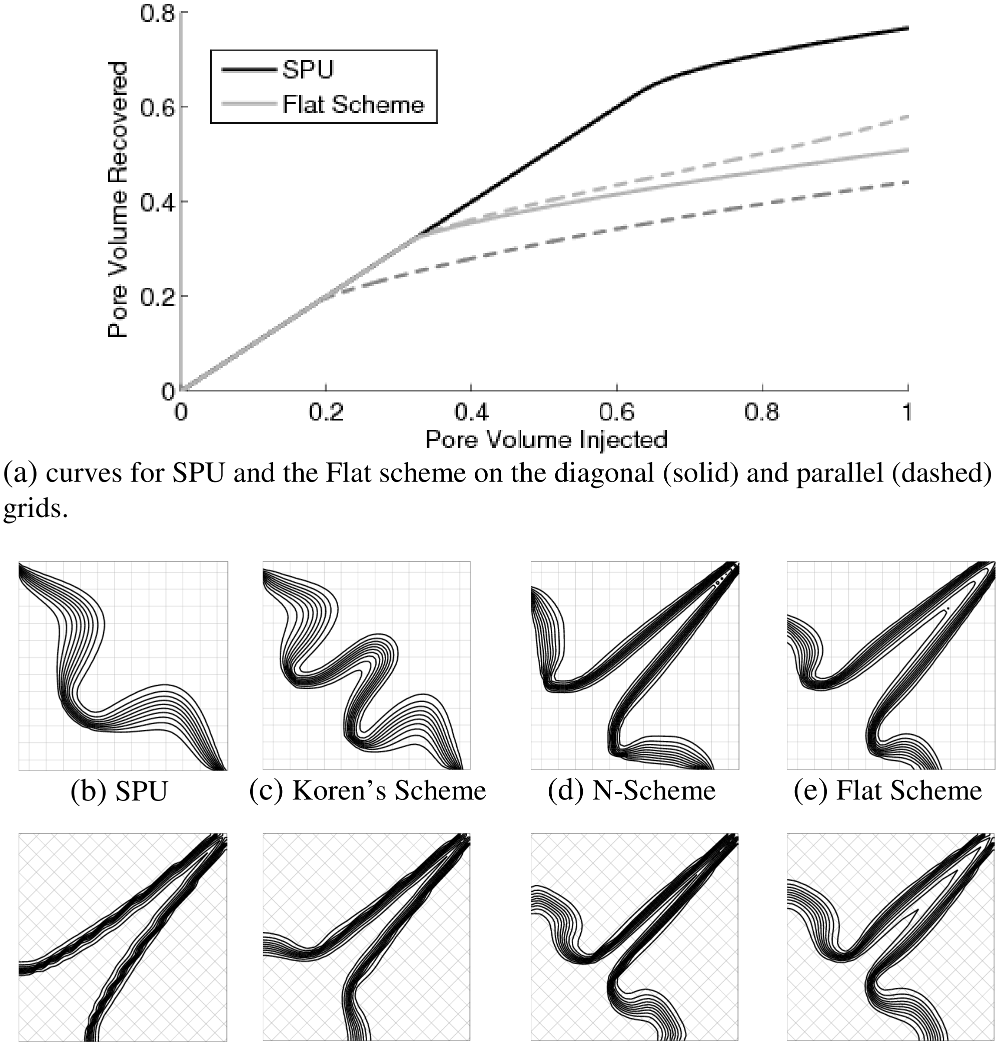
Table 1: Comparison of single parameter κ for four well known numerical methods: the N-scheme (Roe & Sidilkover, 1992), the CTU scheme (Collela, 1990) and Koren’s scheme (Koren, 1991)

Common Name	ϖ_p
Single Point Upwinding (SPU)	$\varpi_p = 0$
The Narrow Scheme (N-Scheme)	$\varpi_p = \min(1, \varpi_p^*)$
Koren's Scheme	$\varpi_p = \varpi_p^* / (1 + \varpi_p^*)$

Table 2: The parameter ϖ_p for SPU, the N-scheme and Koren's scheme, with $\varpi_p^* = \max(0, U_{uw} / U_p)$

Range of $\tilde{\theta}$	$\tilde{\theta}$ Pair of (a_p, b_p) values
$0 \le \tilde{\theta} \le \theta^*$	$(\frac{\tilde{\gamma}^3 \psi - \tilde{U}\tilde{V} \,\vert\, \tilde{V} - \tilde{U} \,\vert}{\tilde{U}(\tilde{U} - \tilde{V})}, 0)$
$\theta^* \le \tilde{\theta} \le \frac{\pi}{2} - \theta^*$	$(0, \frac{\tilde{\gamma}^3 \psi - \tilde{U}\tilde{V} \,\vert\, \tilde{U} - \tilde{V} \,\vert}{2\tilde{U}\tilde{V}})$
$\frac{\pi}{2} - \theta^* \le \tilde{\theta} \le \frac{\pi}{2}$	$(0,0)$

Table 3: (a_p, b_p) pairs for half-face p. If $U_{p-1} < 0$ or then $(a_p, b_p) = (0,0)$. Parameters for the coefficients calculation are defined as $\tilde{\theta} = \arctan(U_{uw} / U_p)$, $\tilde{U} = 2 / (1/U_{p-1} - 1/U_{p+1})$, $\tilde{V} = \tilde{U} \tan \tilde{\theta}$, and $\tilde{\gamma} = \sqrt{\tilde{U}^2 + \tilde{V}^2}$.



(a) curves for SPU and the Flat scheme on the diagonal (solid) and parallel (dashed) grids.

(b) SPU (c) Koren’s Scheme (d) N-Scheme (e) Flat Scheme

(f) SPU (g) Koren’s Scheme (h) N-Scheme (i) Flat Scheme

Figure 3: A miscible adverse mobility ratio, M = 30, quarter-five spot displacement simulated on a 50 X 50 diagonal grid and 71 X 71 parallel grid. SPU, Koren's scheme, the N-scheme, and the Flat scheme are compared for displacement fronts shown at 0.4 PVI. Solutions (b)-(e) were computed on the diagonal grid and the contours in (f)-(i) used the parallel grid

5. Parallel Streamline Simulation

5.1 Parallel programming on shared memory systems

The success of streamlines is based on the physical observation that in heterogeneous reservoir the time scale at which fluids flow along streamlines is often much faster than the time scale at which the streamline locations change significantly. This allows decoupling of the costly three-dimensional transport problem into a set of one-dimensional advection problems along streamlines. Streamline simulation uses a dual grid approach: the equations governing pressure are solved in an Eulerian approach on a fixed three-dimensional grid, which we will refer to as the pressure grid. The equations governing transport are solved along the individual streamlines of the streamline grid.

It has frequently been mentioned that a strong computational advantage of streamline methods is that the decoupled one-dimensional transport systems are naturally parallelizable. In many applications, and in particular in compositional processes where costly thermodynamic equilibrium calculations are performed to model the transport, the 1D streamlines contain the bulk of the computational work. Although this inherent parallelism of streamline simulation is often given as one of its main attractions, to date no work has been published that has convincingly demonstrated this feature. The first motivation for our work is therefore to discuss parallelization and the associated challenges. Although it is true that the decoupled PDE problems along streamlines are independent and therefore naturally parallelizable, parallel performance may be affected by load balancing, computational overheads introduced by the parallelization, memory locality issues as well as costs associated with communication. We address these issues in this work.

We are also motivated by the recent developments in computer technology. Parallel computers are now widely available and affordable. Moreover, all the major manufacturers of microprocessors are replacing conventional single-core processors with multi-core processors, or chip-multiprocessors (CMP) (Olukotun & Hammond, 2005). CMP is an architecture where several processors called cores are placed together within a single microchip, essentially creating a shared memory system on a chip. This shift in technology to a parallel computing environment will likely have a strong impact on the performance of computational algorithms. Non-parallel software may in fact run slower on a new multi-core chip than on a traditional single-core chip, because the individual cores on CMPs typically exhibit simpler in-order designs and lower clock frequencies compared to contemporary state-of-the-art single-core processors. Hence the second motivation for studying parallel streamline solvers is to be able to sustain peak performance on future chip architectures.

Because of the move to CMPs, we deemed it most important to analyze streamline parallelization using a shared memory model. If several CMPs are connected together to form a distributed parallel cluster, we hope to be able to reuse a shared memory implementation within each node of the cluster. We present results for three systems: a Sun Fire ultraSPARC-IV server, a Sun v40z AMD Opteron server, as well as the new CMP just recently released by Sun, the UltraSPARC T2. In this work, we focus primarily on the parallelization of the streamline solves and the communication between streamlines and the pressure grid.

We test on two 3D domains with varying levels of heterogeneity and different well spacings. The physical model we use---single-phase flow in an incompressible system---is kept simple for two reasons. First, this model gives the most conservative scenario: the contributions of the tracing and mapping stages are significant compared to the solution of the transport problems and parallel efficiency will be reduced as a result. Second, using this linear physics allows us to accurately predetermine the workload per streamline so we can employ and test explicit load balancing algorithms. However, when modeling real field-cases, transport-physics will be significantly more complex, shifting more computation time onto the 1D solve and thus leading to a higher parallel efficiency. The numbers we find in this work, therefore, can be seen as lower bounds.

Parallel methods and parallelization in reservoir simulation

Parallel computing is a widely used and powerful technique to increase computational performance. In theory, if a simulation takes T_serial time on a single processor, a parallel computer system can solve the same problem in T_parallel=T_serial/p time, where p is the number of processors used. However, it is generally not possible to achieve such linear scalability because of overheads introduced by the parallelization, such as the time spent in communication between processors, and an imbalance in the computational load carried by the processors. Additionally, not all parts of complex codes can be parallelized with the same efficiency. For the vast majority of computational codes, some serial section will always remain leading to sub-linear scalability, according to Amdahl's Law. If good scalability can be achieved, parallel computing allows larger problems to be solved, and/or shorten turn-around times for simulation studies. In reservoir simulation the latter is very attractive, as traditional engineering work flows such as history matching and field optimization problems typically require many forward simulations.

In most computer systems, the memory address of the next instruction to be executed in a process is stored in a so-called program counter. Modern systems also support threads, that is, within each process we can have several program counters active, pointing to different locations in the same program. If the hardware supports parallel execution, the threads can be mapped to different hardware program counters for parallel execution. Threads share the address space of a single process, and communicate by reading and writing to this shared address space. Processes themselves communicate by explicitly transferring data. On the software side, a parallel program can conceptually be thought of as a collection of program counters working on the same program in parallel, using some combination of processes and threads. One option is, for example, to create a set of processes each carrying an identical copy of the program. This is the distributed memory model. Another option is to create a single process and start multiple threads within this process. This, we refer to as the shared memory model, which is the model that we consider in this work. For shared memory, common tools and languages include OpenMP (OpenMP, 2005) or various public domain libraries, such as POSIX (POSIX, 1996).

5.2 Streamline solvers and parallelization

The general streamline simulation process consists of a 5-step cycle, which is repeated until the final simulation time is reached. We briefly describe each of the steps and the computational approaches used in a generic streamline simulator:

1. Given boundary conditions (well conditions) and initial conditions, the pressure equation is solved on the three-dimensional (Eulerian) pressure grid. With the pressure known, the velocity is explicitly computed using Darcy's law. Finite difference or finite volume methods are generally used to discretize the equation.
2. Given the velocity field, tracing is generally performed with Pollock's analytical tracing method (Pollock, 1988). During tracing of a streamline, the time of flight of the entry and exit points of each grid cell crossed by the streamline are recorded. The resulting time of flight discretization is often highly irregular since streamlines can pass arbitrarily close to vertices of the background grid and/or local flow velocities can vary across orders of magnitudes. The 1D grid along each streamline is therefore generally post-processed in some way to improve the quality and efficiency of the transport solves.
3. Solution variables are mapped from the pressure grid onto the streamlines. The first order mapping used in most streamline simulators assumes that saturations are piecewise

constant on the cells of the pressure grid. These constant values are taken from each background grid cell and assigned to the 1D streamline segments that cross those cells.

4. The transport equations are solved along the streamlines using an appropriate numerical method and for a predefined global time step. During this global time step, the pressure field, and hence the velocity field and streamlines are held fixed. A pressure update and thus a change in streamline geometries happens because new wells come online, production/injection rates change and/or the total mobility field changes.
5. At the end of the global time step, the newly computed solution variables are mapped back from the streamlines to the pressure grid. The 1D profiles must be remapped to the original time of flight.

The process is now restarted from step 1.

The streamline method is similar to an IMPES (IMplicit Pressure, Explicit Saturations) approach for oil-water or black-oil systems, or an IMPEC (IMplicit Pressure, Explicit Compositions) approach for compositional problems with the difference that the transport equations are solved on the specialized transport grid formed by the streamlines (Gerritsen et al., 2007). There are two distinct time step sizes in this process. The first is the time step between pressure updates (the global time step). The second is the time-step used in the numerical transport solve along streamlines to move saturations or compositions between pressure updates (the local time step). The local time steps are determined by stability criteria when explicit schemes are used, and by the desired temporal accuracy in case of implicit methods. The local time step sizes may vary from streamline to streamline, which is attractive in explicit methods where the stability restrictions can be locally severe.

We use the common coverage method in which a coarse streamline grid, for a fixed number of streamlines per face of the well blocks, is traced first. This is followed by a search for cells in the pressure grid that are not traversed, and a forward and backward trace from these cells to the producer and injector, respectively.

Parallelization of streamline solvers

Parallelism can be extracted in an obvious way by simply assigning streamlines to threads. Since the streamlines are fully decoupled from each other, linear scalability would be expected. However, load imbalance and parallelization overhead may cause the scalability to be lower. Furthermore, communication with the pressure grid through mappings, tracing of streamlines and streamline grid creation may also lower efficiency. We use the term bundle to refer to a set of streamlines that are part of the queue ready to be distributed among the available threads. Load balancing is easier to achieve for bundle sizes that are large relative to the number of threads as there is more flexibility when assigning streamlines to threads. We keep all streamlines that make-up the bundle in memory, and thus the size of the bundle will determine the amount of memory allocated for the streamline grid. Theoretically, this approach could lead to cache misses and therefore efficiency reductions if the bundles are very large. This is not an issue on the shared memory architectures tested, since the number of threads that can be used effectively on these systems is low and the bundles can be kept relatively small without affecting load balancing. An alternative is to only use the launch points to define a bundle, in which case there would be no memory issue. In other words, a bundle would simply be made up of a list of launch points, and as the points are distributed to free threads, the streamline are traced, solved, and mapped back.

Pollock's tracing algorithm can be parallelized by computing the exit points in all directions of a cell simultaneously. Obviously this parallelization can only scale to 4 threads in 2D and 6 threads in 3D. To extract more parallelism we can trace multiple streamlines concurrently in one of two ways. In a static assignment, each thread is assigned a launch point and traces the entire streamline from start to finish, for example, from injector to producer. The computational tasks are fairly coarse-grained as they consist of tracing an entire streamline. It is not feasible to perfectly load balance a static assignment, because we do not a-priori know the number of cells a streamline will cross. But, as long as we have many more streamlines than threads, as will the case in industrial applications, we expect any thread idle time to be low on average. In a dynamic assignment the individual segments (grid cells) of all streamlines in the bundle are assigned to threads on a first-come-first-serve basis. Although this generally results in a better load balance it leads to strongly increased communication and overhead, and is not pursued here.

In our specific implementation, all streamlines in a bundle are traced and the 1D problem solved along them before the solution variable are mapped back to the pressure grid. In this mapping step we have the same choice of assigning segments to threads as in the case of tracing. Extra care must be taken in this step in the presence of pressure grid cells that are crossed by a large number of streamlines, e.g., cells around wells and in other high flow areas. When several threads are writing to the same cell, we must enforce mutual exclusion to the variables which will introduce a sequential bottleneck and affect the parallel efficiency. The problem can in some cases be mitigated by careful selection of the order in which segments are mapped back. As an example, if we use a static assignment and map each streamline starting from its first segment adjacent to the injection well, the cells around the injector will suffer greatly from memory contention. To alleviate this contention, we can do a circular shift of the streamline segments and loop though them in order. In this way only one thread will start mapping from the injector well provided that there are more segments than threads for each streamline. If a dynamic assignment is used the cell contention will be more randomized depending on in which order the segments are taken from the work-queue. An alternative is to map back to the underlying pressure grid as soon as the solution variables are known along a streamline. Although mutual exclusion would still have to be enforced, the probability that two or more streamlines would try to map to the same static grid cell would be reduced.

After tracing a bundle we have all data needed for creating the 1D streamline grids, and for solving these 1D problems along the streamlines. We now have two options for parallelization: we either assign sets of complete streamlines in the active bundle to each thread in some manner, which we refer to as the owner approach, or we assign multiple threads to each streamline. We call this the distributed approach. The owner approach is similar to the static approach used for the parallelization of the tracing. The load imbalance will now depend on the computational streamline loads, as well as the number of streamlines per thread. If the streamline loads vary widely, but the ratio of streamlines to thread is high, a random assignment of streamlines to threads may very well balance the total workload per thread. In other words, for high streamline to thread ratios, the load distribution will be less insensitive to variations in individual streamline work loads and the random assignment, which can be left to the built-in scheduler in the shared memory architecture, will likely work well. For completeness, we also investigated the use of an explicit load balancing algorithm, such as the aforementioned CCP solve, that may lead to improved performance for lower ratios. In such a case, however, we must be able to accurately estimate individual streamline loads. The estimates will depend on the physical model simulated. In water flooding or black-oil type simulations, for example, the work will not vary much between segments along any streamline and can be reasonably estimated a-priori. Total costs will depend on required local time step sizes, which may vary from streamline to streamline, the number of segments along the streamlines, and the chosen numerical methods. In compositional problems, the estimates are more complicated to obtain when nonlinear equations of state must be solved (flash calculations) in multi-phase regions. We then need to estimate both the extent of multi-phase regions along each streamline and understand the expected rate of convergence of the flash calculations. Explicit load balancing introduces parallel overhead. If this overhead cannot be amortized the advantage of explicit load balancing may disappear.

In the distributed case, threads are not assigned complete streamlines, but partial streamlines, or streamline sub-grids. Streamline sub-grids may be as small as one grid cell. A small amount of communication is introduced at the boundaries between the sub-grids. If the 1D streamline grid is small, the ratio of computation to communication may be low and the overhead significant depending on the hardware used. Also, if an implicit or adaptive-implicit method is used the upwind data dependencies must be honored which may limit the amount of parallelism available. From a load balancing point of view the distributed case has an obvious advantage as we can evenly distribute the sub-grids to the threads. Various hybrid approaches can also be used. One option would be to first split the active streamline bundle in sub-bundles, each assigned to its own set of threads. These threads can then be assigned to streamline sub-grids as in the distributed approach. However, as discussed above, it is not expected that load balancing is a critical problem for field-scale applications with a high streamline to thread ratio, and for this reason we have selected the owner approach in this work. This has the additional advantage that code from an existing sequential implementation can be reused.

5.3 Parallelization strategy

We have implemented a single-phase streamline simulator using a Cartesian Cell-based Anisotropically Refined (CCAR) grid as the pressure grid (Gerritsen et al., 2007, and Keats & Lien, 2004). Because of their adaptivity, CCAR grids are unstructured in nature and require indirect addressing to find neighboring cells.

Here, we apply uniform CCAR grids only with a standard 7-point stencil. We solve the linear system using a GMRES solver preconditioned by an algebraic multigrid method from the HYPRE package (Chow et al, 2006).

The tracer equation is discretized using the standard single-point upwinding (SPU) scheme. The tracer solves along the streamlines are much cheaper to solve than more complex multi-phase problems generally encountered in streamline simulation of real reservoirs. As a consequence, the tracing and mapping stages in our experiments will receive a disproportionate weight compared to the actual transport solve. In more realistic settings the tracing and mapping stages will not be as heavily weighted as here, meaning that parallel efficiencies in practice will be higher than those measured in this study.

To trace streamlines we use the dual trace method, which is the method used in the commercial package 3DSL, a widely used commercial streamline simulator. This approach is attractive because it allows us to pick up load balancing information during the first trace, which is needed in case an explicit load balancing algorithm. In the first stage of tracing, streamlines are started at injector wells. We select a total of N candidate launch points from the injector grid block faces to build an active bundle of N streamlines. We repeat this process until sufficient streamlines are traced from all injector wells. Next, we start the cover mode in which we search for and select at most N empty interior grid cells, and launch new streamlines from these cells to ensure proper coverage. We repeat the search for empty cells and the tracing from these cells until all grid cells are covered. The building of an empty grid cells list is not parallelized in the current implementation. However, since N is usually not large, we expect the efficiency loss to be negligible.

As the run-time costs of tracing an individual streamline cannot be estimated a-priori, each thread is simply assigned N/P streamlines to trace, with N the size of the active bundle and P the number of threads. In OpenMP this can be accomplished by adding a work-sharing directive with a static schedule. In the first trace of a streamline, the thread counts the number of segments and computes the total time-of-flight. It then allocates memory to store the now known number of segments (irregular grid cells) along the streamlines, and starts the second tracing step reusing the same launch point. In this second step, it stores each segment, picks up the transport variables from the pressure grid, and stores them in the streamline data structures.

We parallelize the 1D solvers using the owner model, that is, by assigning whole, not partial, streamlines to threads. Our primary reason for choosing the owner model is to reduce communication and increase data reuse by matching the partitioning from the tracing when possible. Another important reason is that the owner model allows us to reuse code from an existing sequential implementation. In the distributed model the 1D solves would have to be rewritten for implicit or adaptive-implicit formulations.

After the tracing, each thread regularizes the irregular 1D streamline grids defined by the streamline segments picked up by the tracing algorithm. In the regularization we use a first-order interpolation scheme. For the tracer flow problem modeled here, we arbitrarily chose the number of regularized grid cells to be twice that of the original number of segments, but in more complex simulations, the 1D TOF grid design is generally optimized for accuracy. After the regularization, we call the actual transport solve and step through time, using a time step size determined by the local stability criteria, until we reach the end of the global time step, at which time the solution variables are mapped back to the pressure grid, a new global time step is chosen, the pressure field is updated, and new streamlines are traced.

We have three options to distribute the N streamlines in the active bundle to the P threads available. We can simply assign N/P streamlines to the threads without any attempt at load balancing. We will refer to this as the unbalanced case. The other option is to assign the streamlines in the active bundle to the threads using the built-in OpenMP schedulers on the shared memory architectures used. We will refer to this option as the dynamic load balancing option. The third option is to use an explicit load-balancing algorithm, such as the CCP solve discussed previously. The effectiveness of either of these three approaches depends strongly on the average number of streamlines per thread and the work load variations amongst streamlines, and we will test these dependencies. The dynamic approach, for example, which effectively assigns streamlines to threads whenever the threads complete a 1D streamline solve, is not expected to be very sensitive to streamline work load variations if the ratio of streamlines to threads is high.

In the explicit load balancing case, we need to define and compute the weights to be used in the load-balancing algorithm. These weights should be good approximations of the actual run-time cost of each 1D solve. The values of the weights will necessarily depend both on the physics simulated and the computer system used. In the case of single-phase flow, we can accurately define the required flops per streamline solve needed since the problem is linear; the workload is directly proportional to the product of the number of grid cells and the total number of local time steps until the next pressure update. The computer system dependent component is much harder to estimate as it depends on the memory system (data locality, communication) and instruction scheduling effects of the processor. In this work, we have left this component out. It is important to note here that as we do not model the system dependent effects we cannot expect this load balancing to be perfect although we think we have captured the most important components in our weights.

When load balancing is used, there will be a reassignment of streamlines after the tracing step. How many streamlines are reassigned depends on how close the optimal partitioning is to the simple N/P partitioning used in the tracing stage. This reassignment will generate communication and may affect the parallel efficiency somewhat.

The three versions also exhibit different data locality properties. In the unbalanced case, the threads trace and solve the same set of streamlines. This can have a positive effect on data locality as the streamline segments, which are created and stored in the tracing might still be available in some cache memory when we start the solve stage. In the load balanced cases data touched in the tracing is likely evicted from the cache memories when we get to the solve stage, as we need to trace all streamlines before performing the load balancing.

5.4 Shared Memory Systems

For all numerical experiments we used three architectures. The first is a large Sun Fire 6900 shared memory server (Charlesworth, 2001) running Solaris 10 equipped with 192 GB DRAM and 24 dual-core UltraSPARC-IV processors running at 1350MHz. The UltraSPARC-IV has a 16 MB external L2 cache which is blocked into two private 8 MB partitions. The SPARC system is the most conventional SMP architecture we test with a more or less uniform access time to memory. This system was originally designed to host a single-core UltraSPARC-III processor. Adding another core to the CPU will stress the interconnect.

The second system is a medium-sized Sun v40z shared memory server running CentOS Linux using the 2.6.9 version of the Linux kernel. This system has 4 dual-core Opteron processors running at 2200 MHz and 16 Gb of DRAM. On the Opteron processor each core has a private 1MB on-chip L2 cache. With the on-chip L2 and higher clock frequencies we expect this system to be significantly faster than the SPARC system. The Opteron system is cache-coherent non-uniform memory access architecture (cc-NUMA). The four dual-core CPUs are connected using a HyperTransport network.

The third system is a Sun SPARC Enterprise T5520 with a single UltraSPARC-T2, also known as the Niagara 2, which runs Solaris 10. We will refer to this machine as the T2 system. The T2 is an

8-core chip capable of executing 64 threads in parallel, 8 per core. The T2 cores are clocked at 1.4 GHz and can retire 2 instructions per cycle where only one can be a floating point instruction. Hence, the Opteron has better per core throughput and more cache memory per thread whereas the T2 can execute more threads concurrently while sharing the cache memory. The T2 features a 3MB on-chip L2 cache that is shared, in contrast to the Opteron system. The T2 system has only very recently been released. At the time of this study we had limited access to a prototype machine, which allowed us to run preliminary comparisons. We include them because we believe that Core Multi-Processor (CMP) architectures like the T2 are representative of the future trend in chip manufacturing.

The code is written in Fortran 95 and compiled using Sun STUDIO 12 compilers on all platforms. On the SPARC machine we used only one core per CPU to maximize bandwidth and the amount of cache memory available to us.

5.5 Experimentation

To evaluate the proposed implementation we designed three flow experiments. The experiments were constructed to explicitly create heterogeneous workloads so that the need for load balancing could be assessed. The first one, which we call SPE10 is taken from the 10th SPE Comparitative Solution project (Christie and Blunt, 2001). As our CCAR pressure grid is designed for grid sizes of powers of 2, we select, in the horizontal, a centered 32x128 subset of the original 60x220 data, and, in the vertical, the bottom 32 fluvial layers of the original 80 layer system. Figure 1 shows a representative layer with well locations. The reservoir is channelized and some of the wells are connected to the main channels. The second case, which we refer to as LOWCORR, is a synthetic dataset of size 128x256x32 with low permeability correlation length. Figure 2 shows the horizontal permeability field for one of the vertical layers and the irregular location of the wells. The third case is a realistic 4 million cell (256x256x64) heterogeneous reservoir with low permeability correlation length. The reservoir contains 5 injector wells and 12 production wells. Figure 3 shows 2000 streamlines projected onto one horizontal slice of the pressure field. Table I gives simulation data for the three test cases.

Table I. Statistics for the experimental setup

	SPE10	LOWCORR	Field Case
Cells in pressure grid	131,072	1,138,688	4,194,304
Number of streamlines	26,896	57,992	383,343
Time step	2000 days	5000 days	5000 days
Number of injectors	1	2	5
Number of producers	4	5	12

The number of streamlines per bundle, N, is taken small enough for the bundle to fit into the cache, yet large enough to ensure a reasonable number of streamlines per thread. It varies between N=200 and N=600. In this range, we have found parallel efficiency measurements not to be sensitive to the actual value of N for the relatively small number of threads available on the shared memory architectures. The total number of flux evaluations per streamline varies widely in all three test cases because of varying streamline length as a result of the reservoir heterogeneity and well placement.

We quantify the effects of load balancing and the overall performance of the code by comparing run times for the three shared memory architectures, the three load balancing algorithms and the three reservoirs. All CPU times are measured using the OpenMP omp_wtime() function. We report the minimum run times out of five simulations in each case, and took care that each time the share memory system was otherwise unloaded. The timings are taken for a single global time step excluding the pressure solver. Hence, all overhead from tracing, mapping and coverage analysis is included in these timings.

We first study if the streamline load variability, present in all three test cases and in any industrial application, requires a specialized load-balancing algorithm for large bundle sizes. We select the SPARC and Opteron systems and start with comparing the unbalanced case, where we simply give each thread N/P streamlines without looking at streamline loads, to the explicitly load balanced case, which assigns streamlines to the P threads using the CCP solve, on the SPE10 and LOWCORR test cases. Table II and III show the timings and speed-up for the SPE10 and LOWCORR cases, respectively. Speed-ups are obtained with respect to the single- thread runtime of the unbalanced case. This single-thread runtime was found to be identical (within 1-2%) to a sequential version compiled without the OpenMP directives.

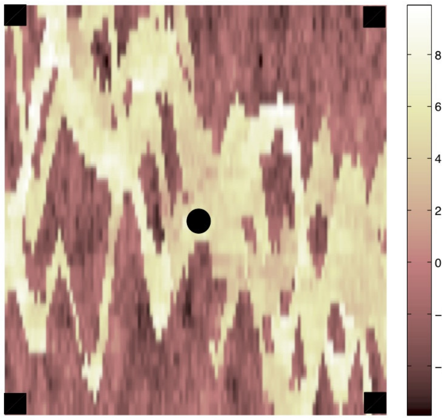


Figure 1. Rendering of the permeability in the x-direction for representative layers for the SPE10 case. The images are colored according to the natural logarithm of the permeability in milli-Darcy. The circles show the location of the injector wells and the squares show the location of the production wells.

Table II. Execution times and speedup for SPE10

Threads	Unbalanced		Explicit load balance	
	Execution time	Speedup	Execution time	Speedup
SPARC				
1	38.98	1.00	39.26	1.00
2	21.05	1.85	20.48	1.92
4	12.03	3.24	11.16	3.52
8	6.98	5.58	6.22	6.31
16	4.75	8.21	4.01	9.79
OPTERON				
1	11.95	1.00	12.06	1.00
2	6.48	1.84	6.45	1.87
4	3.62	3.30	3.50	3.45
8	2.14	5.58	2.07	5.83

Table III. Execution times and speedup for LOWCORR

Threads	Unbalanced		Explicit load balance	
	Execution time	Speedup	Execution time	Speedup
SPARC				
1	356.05	1.00	357.76	1.00
2	182.51	1.95	184.62	1.94
4	97.82	3.64	100.45	3.56
8	52.81	6.74	55.86	6.40
16	29.05	12.26	32.51	11.00
OPTERON				
1	104.21	1.00	104.72	1.00
2	49.98	2.09	51.90	2.02
4	26.68	3.91	27.04	3.87
8	14.37	7.25	14.61	7.12

In the SPE10 case, load balancing has a small effect on the overall performance. In the LOWCORR case, explicit load balancing does not lead to any improvements in the parallel performance at all. This is not because the streamline solves are not well balanced. Rather, the actual load of each thread is affected by communication, instruction scheduling and memory system effects, and clearly explicit load balancing introduces penalties that negate gains made by load balanced transport solves.

The above experiments were conducted for an average bundle size of N=200 leading to at least 25

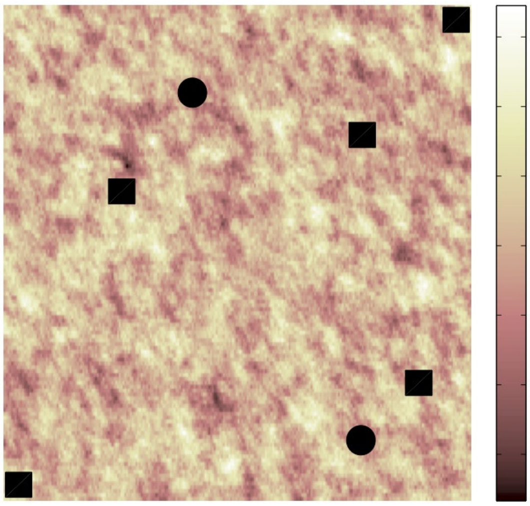


Figure 2. Rendering of the permeability in the x-direction for representative layers for the LOWCORR case. The images are colored according to the natural logarithm of the permeability in milli-Darcy. The circles show the location of the injector wells and the squares show the location of the production wells.

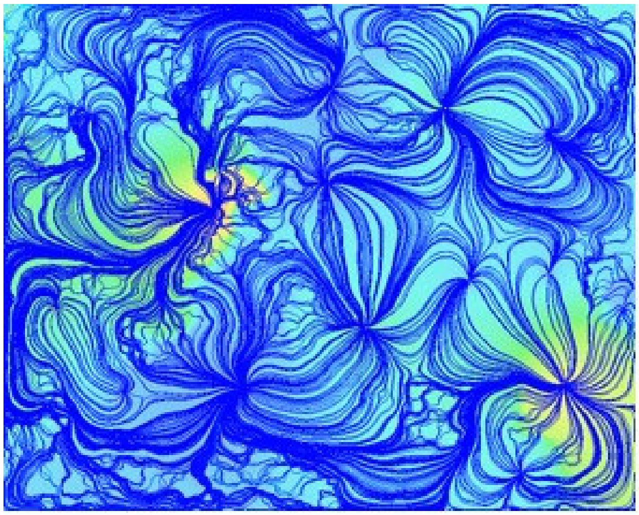


Figure 3. Visualization of streamline paths for the field case. The image shows the projection of 2000 streamlines on a horizontal slice of the pressure field.

streamlines per thread on the Opteron machine. We also conducted experiments for smaller N. As N decreases, explicit load balancing becomes attractive. However, in realistic settings the number of streamlines per thread will be high, which reduces the sensitivity to streamline workload variation, and explicit load balancing is likely unnecessary.

The scalability for both shared memory systems is more favorable in the LOWCORR case. For the Opteron, this case even gives near-linear speed-up. To understand the observed differences better, we measured the relative run-time costs for a single thread of the primary components of the simulator, excluding the pressure solve, when the explicit load balancing algorithm is used, and displayed the measurements in table IV. The category miscellaneous contains the mappings, overhead from the coverage algorithm itself, as well as the CCP solve used for load balancing. These operations contribute little to the overall runtimes. Tracing seems to be fairly expensive on the Opteron architecture, which exhibit smaller cache memories compared to the SPARC CPUs.

Table IV. Percentage run-time costs on a single thread.

	SPE10		LOWCORR	
	SPARC	Opteron	SPARC	Opteron
Tracing	41	54	20	31
Transport solves	53	37	76	63
Miscellaneous	6	9	4	6

With these data and the performance data for the explicit load balancing cases we can estimate the speed-up of the combined tracing and miscellaneous operations, because we know that the transport solves themselves are load balanced well. The numbers are displayed in table V. Both SPE10 and LOWCORR data lead to these approximate speedups, and so we can use them with a reasonable degree of confidence.

Table V. Estimated speedup for tracing, coverage, mapping and CCP solve.

Threads	SPARC	Opteron
2	1.7	1.9
4	2.8	3.4
8	4.3	5.4
16	6.3	

These estimates allow us to approximately predict streamline solver speed-up for cases with more complex physics. For example, when comparing transport solves for a 10-component compositional problem to the three tracer flow solves used here, we find that the computational load of one streamline solve is two or three orders of magnitude higher. This is because of an increased number of flux evaluations, increased cost per flux evaluations, stricter stability constraints on the time step size, and costly equation of state solves along the streamlines. Even taking a low factor of 20, we get near-linear speedup for both architectures, with estimated speedups of around 7.8 on both architectures for 8 processors. For a factor of 100, the predicted speedup is linear.

In a second comparison, we compare the parallel efficiencies for explicit and dynamic load balancing on the Opteron system for the SPE10 case and the field case. Here, we use a bundle size of N=600, which explains the slight discrepancies with the explicit load balancing numbers reported in tables II and III. The results in table VI show favorable parallel performances, with the dynamic scheduling approach consistently outperforming explicit load balancing. Although the streamline workloads are not uniformly distributed, the actual load balance is mostly unaffected because of the high number of streamlines per thread. A strong advantage of dynamic load balancing explicit load balancing is that the former is relatively straightforward to implement using the built-in OpenMP scheduler. Again, the speedup decreases as the number of threads grows because of low speedup of the costly tracing and miscellaneous components. As shown in the previous subsection, the speedup will be more favorable or applications with more complex physics, also in the case of dynamic load balancing. As before, much smaller bundle sizes leads to reduced performance of the dynamic scheduler as the number of streamlines per thread becomes sufficiently low for load imbalances to start playing a role.

Table VI. Execution times and speedup for Opteron.

Threads	Explicit load balance		Dynamic load balance	
	Execution time	Speedup	Execution time	Speedup
SPE10				
1	12.08	1.00	12.01	1.00
2	6.73	1.79	6.64	1.82
4	3.84	3.15	3.74	3.23
8	2.28	5.30	2.21	5.47
Field case				
1	307.94	1.00	307.77	1.00
2	161.93	1.90	161.97	1.90
4	94.35	3.26	92.05	3.35
8	55.61	5.54	50.06	6.15

Table VII shows the timings for the SPE10 case on the T2 system using dynamic load balancing. The number of streamlines in an active bundle is N=600. Note that this system exhibits 64 hardware threads in total but only 8 of these can execute floating-point instructions concurrently. The scalability on this system is almost perfectly linear for up to 8 threads, and higher than on the Opteron system. In terms of absolute performance the Opterons are still faster because of the higher throughput cores of this architecture. Even though the T2 has only 8 floating point units we still see improved performance using more than 8 threads. This is probably due to the fact that the other hardware threads can execute integer related instruction while the floating point units are waiting for data from the memory system, but the speedup is no longer linear.

Table VII. Execution times and speedup for SPE10 on the T2 (dynamic load balancing).

Threads	Execution time	Speedup
1	84.40	1.00
2	42.44	1.99
4	21.36	3.95
8	10.91	7.73
16	6.27	13.47

Because the solver workloads per thread remain identical when moving from the Opteron system to the T2 system, the differences in observed scalability must be explained by system characteristics, data locality, memory access costs and/or communication costs when moving data to and from the pressure grid in tracing and mapping steps. Indeed, communication costs are expected to be lower on CMP machines like the T2 system than on architectures like the Opteron system because CMP machines feature shared caches. We believe that it is indeed the decreased communication costs associated with a shared cache that is the main reason for the improved observed scalability.

5.6 Discussion and conclusions

We have shown in this work that streamline solvers for real-field applications are very suitable for parallelization on shared memory systems. High parallel efficiency can be achieved using built-in OpenMP dynamic schedulers to assign streamlines to threads if the number of streamlines per thread is high, as is the case in real-field applications. In such cases, the parallel efficiency is not sensitive to variations in workload between streamlines. We also implemented and tested a specialized static load-balancing algorithm and compared the overall parallel performance with that of the built-in schedulers. Using static load balancing does not lead to any noticeable improvement. This is good news as it simplifies parallelization significantly.

We employed the owner approach in which threads are assigned whole streamlines. This approach is more attractive than a distributed model, in which streamline segments are assigned to threads, because it allows re-use of existing sequential code for the 1D streamline solves, also for implicit time-stepping algorithms. We found any load balance advantage of the distributed model to be negligible for real-field applications where the streamline to thread ratio is generally high.

We carefully analyzed dynamic vs. static schedulers, owner vs. distributed assignment, as well as the effects of bundle size to understand whether or not data locality issues were important. We found data locality to be much less of an issue that we originally had expected. Although we saw some degradation due to data locality issues for low streamline-to-thread counts, for larger values we could not record any noticeable differences.

The tracing and mapping stages of the streamline solves are not as easily, nor as well parallelizable as the 1D transport solves. For the simplified single-phase test problem used here, the 1D transport solves were relatively inexpensive. Even so, the measured parallel efficiency was high (around 70%) for all test cases. Using the measured parallel speed-ups for the simplified problem, it is not hard to estimate parallel speedup for problems with more complex physics. With estimated 1D computational costs being a factor of 100 to 1000 more expensive than single-phase in compositional-type simulations, we expect near-linear scalability on the shared memory machines tested.

We are particularly encouraged by the high parallel efficiencies observed on the new multi-core processor architecture, because this is the direction in which future computer systems are heading. We believe that it is the decreased communication cost associated with a shared cache (not to be confused with a shared memory) that is the main reason for the improved scalability.

We did not look at parallelization of the pressure solver, which is well documented in the literature. The overall speedup of a streamline simulator will be affected by the parallel efficiency of the pressure solver. However, with increasing physics, the cost per simulation time step is shifted towards the 1D transport solve along the streamlines. The cost for solving the scalar equation for pressure is relatively insensitive to physics and instead much more related to the size (number of active cells) and spatial distribution of geological parameters such as permeability, and grid geometry. Thus, as problems become larger and more complex, the key component will be an efficient parallelization of the 1D transport solves and standard parallelization of the implicit pressure solver should work well.

We have not investigated parallelization on distributed architectures, but rather focused on shared memory systems because of the strong trend in this direction with the advent of multi-core processor (CMP) chips. While distributed computing will remain an important part of reservoir simulation, we see current trends moving towards solving many concurrent simulations on large clusters, where each node in the cluster might be a CMP architecture. However, distributed models do raise a number of challenging questions associated with domain decomposition. Tracing, mapping and solving across domains will be more involved, and communication overheads more critical. More research is needed to confirm this however.

6 Project conclusions and specific outlines

To summarize, we present the major conclusions and outcomes of this project.

6.1 Compositional streamline simulation framework

We built a complete framework that allowed streamline simulation to be extended to fully compositional processes, such as those encountered in gas injection. The major conclusions and outcomes of this work are

- An adaptive streamline coverage algorithm is developed. Adding streamlines locally can reduce computational costs by concentrating computational efforts where needed, and reduce mapping errors. Adapting streamline coverage effectively controls mass balance errors that mostly result from the mapping from streamlines to pressure grid.
- It is now possible to add partial streamlines, that is, streamlines that do not necessarily start and/or end at wells. This allows more efficient coverage and avoids the redundant work generally done in the near-well regions.
- A higher order mapping from pressure grid to streamlines is designed that significantly reduces smoothing errors.
- A Kriging algorithm is used to map from the streamlines to the background grid. The higher accuracy of the Kriging mapping means that it is not essential for grid blocks to be crossed by one or more streamlines. The higher accuracy comes at the price of increased computational costs.
- To reduce errors associated with fixing the pressure field between pressure updates, a higher order global time-stepping method is developed that allows the use of larger global time steps.
- Third-order ENO schemes are suggested to propagate components along streamlines. Both in the two-phase and three-phase experiments these ENO schemes outperform other (higher

order) upwind schemes. Application of the third order ENO scheme leads to overall computational savings because the computational grid used can be coarsened.

- Grid adaptivity along streamlines is implemented to allow sharp but efficient resolution of solution fronts at reduced computational costs when displacement fronts are sufficiently separated.
- A correction for Volume Change On Mixing (VCOM) is implemented that is very effective at handling this effect.
- A specialized gravity operator splitting method is proposed for use in compositional streamline methods that gives an effective correction of gravity segregation.

6.2 Parallel streamline solvers

We developed a parallelization strategy for streamline solvers on the next generation shared memory machines. The main conclusions and outcomes of this part of the project are:

- The built-in dynamic scheduling strategies of OpenMP lead to parallel efficiencies that are comparable to optimal schedules obtained with customized explicit load balancing strategies as long as the ratio of number of streamlines to number of threads is sufficiently high, which is the case in real-field applications. This is an important result, as it eases the transition of serial to parallel streamline codes.
- The parallel speedup depends on the relative contribution of the tracing and mapping stages as compared to the solution of the transport equations along streamlines. As the physical complexity of the simulated 1D transport process increases, the contribution of the less efficient tracing and mapping stages is reduced and near-linear scalabilities can be obtained.
- The owner approach, in which threads are assigned whole streamlines, is more attractive than a distributed model, in which streamline segments are assigned to threads, because it allows re-use of existing sequential code for the 1D streamline solves, also for implicit time-stepping

algorithms. Any load balance advantage of the distributed model is negligible for real-field applications where the streamline to thread ratio is generally high.

- Initial concerns about data locality and cash misses never materialized in our work, and little or no performance degradation could be attributed to data locality.
- Parallel speedup on the new generation Core Multi Processor (CMP) architectures is particularly high: we achieved linear speedup in all tests. We attribute this also to the particularly efficient shared cache design of modern CMP architectures.
- This work confirms what many researchers have assumed about streamline simulation: that it is trivially parallelizable on shared-memory machines and that near-linear scalability can be achieved.

6.3 New compositional solvers

We developed two new types of transport solvers specifically designed for gas injection processes: relaxation methods and iso-diffusive multi-D schemes. The relaxation schemes can be applied to the one-dimensional streamline transport equations. The multi-D schemes are specifically designed for those stages in the gas injection process where Eulerian methods are desirable. For example, in the early stages of injection, gravity segregation is very strong and streamline methods are not capable of predicting the process with sufficient accuracy. A Eulerian approach can then be used to such time that gravity segregation is more or less established and streamline methods can take over.

For relaxation methods, key outcomes of these developments are:

- The constant sub-characteristic Jin-Xin scheme is attractive for two-phase multicomponent systems in that it removes the dependency of the numerical solver on the eigenstructure of the system and nonlinear Riemann solutions. But, the necessary restriction on the sub-characteristic speed results in excessive numerical diffusion that can significantly reduce solution accuracy in strongly nonlinear compositional problems.

- The new constant $\alpha - \beta$ system does not offer sufficient improvement and behaves like the Jin-Xin scheme for eigenvalues with equal magnitude, or the corresponding upwind schemes when small negative eigenvalues are present.
- The variable version of the Jin-Xin relaxation maintains all the advantages of standard the Jin-Xin scheme, while reducing numerical diffusion considerably by locally imposing the sub-characteristic condition. For ternary gas-oil displacement system, our extensive testing showed that in many cases the second order variable relaxation scheme is competitive with component-wise ENO-RK3 and that it is always as least as good as component-wise TVD-RK2.
- Variable $\alpha - \beta$ relaxation, where all eigenvalues are chosen to closely follow the physical speeds of the system, will result in an ill-conditioned problem. A better alternative is to vary only the positive eigenvalues of the $\alpha - \beta$ relaxation system, while keeping the negative eigenvalues close to $-f'_{max}$. This results in a scheme with performance similar to variable Jin-Xin relaxation.

For the truly multi-D scheme developments, key outcomes are:

- Modified equations analysis can be used in a predictive manner for determining preferential flow angles for numerical methods on structured grids.
- A general framework for multi-D schemes with local positivity constraints can be formulated using interaction regions. Specifically, a subset of purely upwind advective schemes is capable of representing several common multi-D schemes.
- It is possible to significantly reduce grid orientation effects and numerical biasing through the

use of truly multi-D schemes and the introduction of numerical diffusion corrections. This was demonstrated for the new Flat scheme, which has near constant transverse diffusion, and gave favorable results on both homogeneous and heterogeneous displacements.

6.4 Project publications

Journal publications

H. Loef, M.G. Gerritsen and M. Thiele, Parallel Implementations of Streamline Simulators, *Computational Geosciences*, 2008.

J. Kozdon, B.T. Mallison and M.G. Gerritsen, Robust Multi-D Transport Schemes with Reduced Grid Orientation Effects, *Transport in Porous Media*, 2008a.

J. Kozdon, M.G. Gerritsen and M. Christie, The Grid Orientation Revisited: Near-well, Early-time Effects and Solution Coupling Methods, *Transport in Porous Media*, 73(3), 255-277, 2008b.

K. Jessen, M.G. Gerritsen, and B.T. Mallison, High-Resolution Prediction of Enhanced Condensate Recovery Processes, *Society of Petroleum Engineers Journal*, Volume 13, number 2, June 2008.

J. V. Lambers, M.G. Gerritsen and B.T. Mallison, Accurate Local Upscaling with Compact Multi-Point Transmissibility Calculations, *Computational Geosciences,* Special Issue on Multiscale Methods for Flow and Transport in Heterogeneous Porous Media, 2008.

M.G. Gerritsen and J.V. Lambers, Integration of Local-Global Upscaling and Grid Adaptivity for Simulation of Subsurface Flow in Heterogeneous Formations, *Computational Geosciences* 12, 193-218, 2008.

S. Krishnamurthy and M.G. Gerritsen, A Variable Relaxation Scheme for Multiphase, Multicomponent Flow, *Transport in Porous Media*, 71(3):345-377, 2008.

B.T. Mallison, M.G. Gerritsen, S. Matringe, Improved Mappings for Streamline Based Simulation, *Society of Petroleum Engineers Journal*, Volume 11, Number 3, 2006.

B.T. Mallison, M.G. Gerritsen, K. Jessen, F.M. Orr Jr., High-order Upwind Schemes for Two-Phase Multi-Component Flow, *Society of Petroleum Engineers Journal*, Volume 10, Number 3, 297-311, 2005.

J. Kozdon, B.T. Mallison and M.G. Gerritsen, Multi-Dimensional Upstream Weighting for Multi-Phase Transport in Porous Media, *SIAM Journal on Numerical Analysis*, 2008.

J.V. Lambers and M.G. Gerritsen, Variable Compact Multi-point Transmissibility Upscaling for 3D Adapted Grids, *Computational Geosciences,* 2008.

S. Krishnamurthy and M.G. Gerritsen, Variable Relaxed Schemes for Multi-dimensional Hyperbolic Conservation Laws, *SIAM Journal on Numerical Analysis,* 2008.

Conference papers

J.V. Lambers, M.G. Gerritsen, D. Fragola, Multiphase, 3-D Flow Simulation with Integrated Upscaling, MPFA Discretization, and Adaptivity, SPE118983, *Society of Petroleum Engineers Reservoir Simulation Symposium*, 2009.

J. Kozdon, B. Mallison, and M. Gerritsen, Multi-D Upwinding for Multi-Phase Transport in Porous Media, SPE-119190, *Society of Petroleum Engineers Reservoir Simulation Symposium*, 2009.

B.T. Mallison, M.G. Gerritsen and G. Kreiss, Asynchronous Time Integration of Flux-conservative Transport,*Proceedings of the 11th European Conference on the Mathematics of Oil Recovery*, Bergen, Norway, 2008.

H. Loef, M.G. Gerritsen and M. Thiele, Parallel Streamline Simulation, SPE113543, *Europec/EAGE Conference and Exhibition*, Rome, Italy, 2008.

J.V. Lambers and M.G. Gerritsen, Spatially-varying Compact Multi-point Flux Approximations for 3-D Adapted Grids with Guaranteed Monotonicity, *Proceedings of the 11th European Conference on the Mathematics of Oil Recovery*, Bergen, Norway, 2008.

K. Jessen, M.G. Gerritsen, and B.T. Mallison, High-Resolution Prediction of Enhanced Condensate Recovery Processes, SPE99619, *SPE/DOE Symposium on Improved Oil Recovery*, Tulsa, 2006.

M.G. Gerritsen, K. Jessen, B.T. Mallison and J.V. Lambers, A Fully Adaptive Streamline Framework for the Challenging Simulation of Gas Injection Processes, SPE 97270, *Society of Petroleum Engineers Annual Technical Conference and Exhibition*, Dallas, Texas, 2005.

J.V. Lambers and M.G. Gerritsen, An Integration of Multi-Level Local-Global Upscaling and Adaptivity, SPE 97250, *Society of Petroleum Engineers Annual Technical Conference and Exhibition*, Dallas, Texas, 2005.

J. Nilsson, M.G. Gerritsen, R. Younis, A novel Adaptive Anisotropic Grid Framework for Efficient Reservoir Simulation, SPE 93243, *Society of Petroleum Engineers Reservoir Simulation Symposium*, The Woodlands, Texas, 2005.

S. Matringe, M.G. Gerritsen and B.T. Mallison, Optimizing Streamline Coverage for Efficiency and Accuracy, SPE 93306, *Society of Petroleum Engineers Reservoir Simulation Symposium*, The Woodlands, Texas, 2005.

J. Nilsson, M.G. Gerritsen, R. Younis, An Adaptive High-Resolution Simulation for Steam-Injection Processes, SPE 93881, *Society of Petroleum Western Regional Meeting*, Irvine, California, 2005.

G. Valenti, B.T. Mallison, K. Jessen and M.G. Gerritsen, High-Order Upwind Schemes for Three-Phase Multicomponent Flows, A Preliminary Investigation, SPE 90594, *Society of Petroleum Engineers Annual Technical Conference and Exhibition*, Houston, Texas, 2004.

References

Aavatsmark, I.: 2002, An introduction to multipoint flux approximations for quadrilateral grids, *Computational Geosciences* 6, 405-32.

Abate, J., Wang, P. and Sepehrnoori, K.:2001, Parallel compositional reservoir simulation on a cluster of PCs, *International Journal of High Performance Computing Applications*, Spring 2001.

Aregba-Driollet D, Natalini R. 1996. ”Convergence of relaxation schemes for conservation laws.” *Appl Anal.* 61:163-193.

Aziz, K, and Settari, A.: 1979, *Petroleum Reservoir Simulation*, Applied Science Publisher, Essex, U.K.

Banda MK. 2005. Variants of relaxed schemes and two-dimensional gas dynamics. *J. Comp. Appl. Math.* 175:41-62.

Batycky, R. P.: 1997, A Three-Dimensional Two-phase Field Scale Streamline Simulator, PhD thesis, Department of Petroleum Engineering, Stanford University.

Batycky RP, Blunt M, Thiele MR. 1997. A 3D field-scale streamline based reservoir simulator *Soc. Petrol. Eng. Res. Eng.* 12:246-254.

Bell, J.B. and Trangenstein, J.: 1989, Mathematical Structure of Compositional Reservoir Simulation*, SIAM Journal of Scientific and Statistical Computing*, 10, 817-845 .

Berenblyum, R.A., Shapiro, A.A., Jessen, K., Stenby, E.H., Orr Jr., F.M.: 2003, Black Oil Streamline Simulator with Capillary Effects, *SPE 84037*, presented at the SPE Annual Technical Conference and Exhibition held in Denver, CO.

Berger, M.J. and Oliger, J.: 1984, Adaptive mesh refinement for hyperbolic partial differential equations, *Journal of Computational Physics*, 53, 484.

Bratvedt, F., Gimse, T. and Tegnander, C.: 1996, Streamline computations for porous media flow including gravity, *Transport in Porous Media* 25, 1, 63-78.

Brezzi, F.; Hughes, TJR; Marini, LD; Masud, A.: 2005, Mixed discontinuous Galerkin methods for Darcy flow, *Journal of Scientific Computing* 22-23, 119-45.

Cao, H.: 2002, *Development of Techniques for General Purpose Simulators*, PhD Thesis, Department of Petroleum Engineering, Stanford University.

Celia, M.A., Russell, T.F., Herrera, I. and Ewing, R.E.: 1990, An Eulerian-Lagrangian Localized Adjoint Method for the Advection Diffusion Equation, *Advances in Water Resources* 13, 187-206.

Chalabi A. 1999. Convergence of relaxation schemes for hyperbolic conservation laws with stiff source terms. *Math. Comput*. 68:955-970.

Chapman S, Cowling TG. 1970. *The Mathematical theory of nonuniform gases*. Cambridge Univ. Press.

Charlesworth, A., The Sun Fireplane System Interconnect, in Proceedings of the 2001 ACM/IEEE conference on Supercomputing, Denver, 2001.

Chen, Y., Durlofsky, L., Gerritsen, M. and Wen, X. H.: 2003, A coupled local-global upscaling approach for simulating flow in highly heterogeneous formations, *Advances in Water Resources* 26, 1041-60.

Cheng, H, Osako, I. and Datta-Gupta, A.: 2005, A Rigorous Compressible Streamline Formulation for Two and Three-Phase Black Oil Simulation, SPE 96866, presented at the SPE Annual Technical Conference and Exhibition, Dallas, Texas.

Chow, E., Falgout, R.D., Hu, J.J, Tuminaro, R.S. and Yang, U.M., A Survey of Parallelization Techniques for Multigrid Solvers, SIAM Series on Software, Environments, and Tools, 2006.

Christie, M. A. and M. J. Blunt.: 2001, Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques, SPE 66599, presented at the SPE Reservoir Simulation Symposium, Houston, Texas.

Cinar, Y., Jessen, K., Berenbluym, R., Juanes, R. and Orr, Jr. F.M.: 2004, An Experimental and Numerical Investigation of Crossflow Effects in Two-Phase Displacements, SPE 90568, presented at the SPE Annual Technical Conference and Exhibition, Houston, Texas.

Colella, P., Multidimensional upwind methods for hyperbolic conservation laws, *Journal of Computational Physics,* 1990*, 87*, 171-200

Crane, M., Bratvedt, F., Bratvedt, K., Childs, P. and Olufsen, R.: 2000, A fully compositional streamline simulator, SPE 63156, presented at the SPE Annual Technical Conference and Exhibition.

Delis AI, Katsounis Th. 2005. Numerical solution of the two-dimensional shallow water equations by the application of relaxation methods. *Appl. Math. Mod.* 29(8): *754-783.*

Deutsch, C.V., Journel, A.G.: 1992, *GSLIB: Geostatistical Software Library and User's Guide*, Oxford University Press.

Di Donato, G. and Blunt, M. J.: 2003, Streamline-based dual-porosity simulation of reactive transport and flow in fractured reservoirs, Water Resources Research 40.

Dindoruk, B.: 1992, Analytical Theory of Multiphase Multicomponent Displacement in Porous Media, Ph.D. Thesis, Department of Petroleum Engineering, University of Stanford.

Douglas Jr., J. and Russell, T. F.:1982, Numerical methods for convection dominated diffusion problems based on combining the method of Characteristics with Finite Element or Finite Difference Procedures, *SIAM Journal of Numerical Ana*lysis, 19, 871-885.

Edwards, M. G.: 2000, M-matrix flux splitting for general full tensor discretization operators on structured and unstructured grids, *Journal of Computational Physics* 160, 1-28.

Gerritsen, M., Mallison, B. and Jessen, K., A Compositional Streamline Framework for Gas Injection Processes, 2007.

Gerritsen M, Durlofsky LJ. 2005. Modeling fluid flow in oil reservoirs. *Annu. Rev. Fluid Mech.* 37:211-238.

Gerritsen, M. and Lambers, J.: 2005, A Specialized Upscaling Method for Adaptive Grids, in review, *Computational Geosciences*.

Gmelig Meyling, R.H.J, A characteristic finite element method for solving non-linear convection-diffusion equations on locally refined grids, in Proceedings 2nd ECMOR conference, 255-262, 1990.

R.H.J. Gmelig Meyling: 1991, Numerical methods for solving the nonlinear hyperbolic equations of porous media flow, in Proceedings Third International Conference on Hyperbolic Problems, Vol. I, II, Uppsala, 503-517.

Guzman, R.E.: 1994, Three phase flow in field-scale simulations of gas and WAG, SPE 28897, presented at the European Petroleum Conference, London, 25-27 October.

Harten A, Engquist B, Osher S, Chakravarthy S. 1987. Uniformly high order accurate essentially non-oscillatory schemes III. *J. Comp. Phys*. 71:231–303

Haukås, J., Aavatsmark, I. and Espedal, M.: 2004, A black-oil and compositional IMPSAT simulator with improved compositional convergence, presented at the 8th European Conference on the Mathematics on Oil Recovery, Cannes, France.

Hornung, R.D., Khan, S. A. and Trangenstein, J. A.: 1996, Adaptive mesh refinement and upscaling for multi-component flow in porous media, *Computational Geosciences*, Feb. 1996.

Hundsdorfer W, Verwer JG. 2003. *Numerical solution of time-dependent advection-diffusion-reaction equations*. Springer Series in Comp. Math.

Hurtado, F.; Maliska, C.; da Silva, A. & Cordazzo, J. A Quadrilateral Element-Based Finite-Volume Formulation for the Simulation of Complex Reservoirs, *SPE Paper 107444-MS presented at the SPE Latin American and Caribbean Petroleum Engineering Conference held in Buenos Aires, Argentina,* 2007.

Ichiro, O., Datta-Gupta, A. and King, M. J.: 2003, Time step selection during streamline simulation via transverse flux correction, SPE 79688 presented at the SPE Reservoir Simulation Symposium, Houston, Texas.

Ingebrigtsen, L., Bratvedt, F., Berge, J.: 1999, A streamline based approach to the solution of three-phase flow, SPE 51904, presented at the SPE Reservoir Simulation Symposium.

Jenny, P., Lee, S. H., and Tchelepi, H. A.: 2005, Adaptive fully implicit multi-scale finite volume method for multi-phase flow and transport in heterogeneous porous media, *Journal of Computational Physics,* in press.

Jessen K, Wang Y, Ermakov P, Zhu J, Orr FM Jr. 2001. Fast, approximate solutions for 1D multicomponent gas injection problems. *Soc. Petrol. Eng. J.* 6(4):442–451.

Jessen, K. and Orr, F.M. Jr.: 2002, Compositional streamline simulation, SPE 77379, presented at the SPE Annual Technical Conference and Exhibition, San Antonio, Texas.

Jessen, K. and Orr, F.M. Jr.: 2004, Gravity Segregation and Compositional Streamline Simulation, SPE 89448, Presented at the 2004 SPE/DOE Fourteenth Symposium on Improved Oil Recovery held in Tulsa, Oklahoma.

Jessen K, Stenby EH, Orr FM Jr. 2004. Interplay of phase behavior and numerical dispersion in finite difference compositional simulation. *Soc. Pet. Eng. J.* 9(2):193-201

Jin S. 1995. Runge-Kutta methods for hyperbolic conservation laws with stiff relaxation terms. *J. Comp. Phys.*122:51-67.

Jin S, Xin ZP. 1995. The relaxation schemes for systems of conservation laws in arbitrary space dimensions. *Comm. Pure Appl. Math*. 48:235-277.

Jin S, Katsoulakis MA. 1997. Relaxation approximations to front propagation. *J. Diff. Equations.* 138:380-387.

Jin S, Xin ZP. 1998. Numerical passage from systems of conservation laws to Hamilton-Jacobi equation, and a relaxation scheme. *SIAM J. Num. Anal*. 35:2385-2404.

Juanes R, Patzek TW. 2004. Relative permeabilities for strictly hyperbolic models of three-phase flow in porous media. *Trans. Porous Med.* 57:125–152

Keats, W.A. and Lien, F., Two-dimensional anisotropic cartesian mesh adaption for the compressible Euler equations, International Journal for Numerical Methods in Fluids, v.46 n.11, 2004.

Koren, B. Low-diffusion rotated upwind schemes, multigrid and defect corrections for steady, multi-dimensional Euler flows *International Series of Numerical Mathematics,* 1991*, 98*, 265-276

Kozdon, J.; Gerritsen, M. & Christie, M. Grid Orientation Revisited: Near-well, Early-time Effects and Solution Coupling Methods *Transport in Porous Media,* 2008

Kurganov A, Tadmor E. 2000. New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations. *J. Comp. Phys.*160:241-282.

Lacroix, S., Vassilevski, Yu., Wheeler, J., and Wheeler, M.: 2003, Iterative solution methods for modeling multiphase flow in porous media fully implicitly, *SIAM Journal on Scientific Computing* 25 (3), 905-26.

Lake LW. 1989. *Enhanced Oil Recovery*. Englewood Cliffs, NJ: Prentice-Hall.

Lambers, J., Gerritsen, M. G., and Mallison, B.: 2007, Accurate local upscaling with compact multi-point transmissibility calculations, *Computational Geosciences*, Special Issue on Multi-Scale Methods, in press.

LeVeque RJ. 2002. *Finite volume methods for hyperbolic problems*. Cambridge texts Appl. Math.

LeVeque RJ, Pelanti M. 2001. A class of approximate Riemann solvers and their relation to relaxation schemes. *J. Comp. Phys.* 172:572-591.

Liu HL, Warnecke G. 2000. Convergence rates for relaxation schemes approximating conservation laws. *SIAM J. Num. Anal*. 37:1316-1337.

Liu TP. 1987. Hyperbolic conservation laws with relaxation. *Comm. Math. Phys.*108:153:175.

Lohrenz, J., B.G. Bray and C.R. Clark.: 1964, Calculating viscosities of reservoir fluids from their compositions, *Journal of Petroleum Technology*, 1171.

Mallison BT. 2004. *Streamline-based simulation of two-phase, multicomponent flow in porous media.* PhD thesis. Stanford Univ.

Mallison, B., Gerritsen, M., Jessen, K. and Orr, F.M. Jr: 2005a, High-order upwind schemes for two-phase, multi-component flow, *SPE Journal*, September 2005.

Mallison, B.T., Gerritsen, M.G., Matringe, S.F.: 2005b, Improved mappings for streamline-based simulation, SPE 89352, *SPE Journal*, in press.

Matringe, S.F., Gerritsen, M.G.,.: 2004, On accurate tracing of streamlines, SPE 89920, presented at the SPE Annual Technical Conference and Exhibition held in Houston, Texas, U.S.A., 26–29 September 2004.

Matringe, S.F., Gerritsen, M.G, Mallison, B.T.: 2005, Optimizing streamline coverage for efficiency and accuracy in a new streamline framework, SPE 93306, presented at the SPE Reservoir Simulation Symposium, Houston, Texas.

Michelsen, M.L.: 1998, Speeding up the two-phase PT-flash, with application for calculation of miscible displacements, *Fluid Phase Equilibria* 1, 143.

Mittal, R., Iaccarino, G.: 2005, Immersed Boundary Methods, *Annual Review of Fluid Mechanics* 37, 239-261.

Mortise, G.: 1998, 1998 Worldwide EOR Survey, *Oil and Gas Journal*, April 20, 49-97.

Mortise, G.: 2000, EOR Weathers low oil prices, *Oil and Gas Journal*, Mar. 20, 39-61.

Nessyahu H, Tadmor E. 1990. Non-oscillatory central differencing for hyperbolic conservation laws *J. Comp. Phys.*87(2):408-463.

Nilsson, J., Gerritsen, M., Younis, R.: 2005, A novel adaptive anisotropic grid framework for efficient reservoir simulation, SPE 93243, presented at the SPE Reservoir Simulation Symposium, Houston, Texas.

Olukotun, K. and Hammond, L., The Future of Microprocessors, ACM Queue v.3 n.7, 26-29, 2005.

OpenMP Fortran Specification v2.5, OpenMP Architecture Review Board, May 2005.

Orr FM Jr., Dindoruk B, Johns RT. 1995. Theory of multicomponent gas/oil displacements. Ind. Eng. Chem. Res. 34:2661–2669.

Orr FM Jr. 2005. *Theory of Gas Injection Processes*. Stanford Univ.

Orr, F.M., Jr.: 2007, *Theory of Gas Injection Processes*, Tie-Line Publications, Copenhagen, Denmark.

Peng, D. Y. and Robinson, D. B.: 1976, A new two-constant equation of state, *Ind. Eng. Chem. Fund.* 15, pp. 59-64.

Pollock, D.W.: 1988, Semi-analytical computation of path lines for finite-difference models, *Ground Water* 26, 6*,* 743-750.

Portable Operating System Interface (POSIX)--Part1: System Application, IEEE Portable Applications Standards Committee, 1996.

Qiu J, Shu CW. 2002. On the construction, comparison, and local characteristic decomposition for high order central WENO schemes. *J. Comp. Phys.* 183(1):187–209

Riaz, A. & Meiburg, E. Linear stability of radial displacements in porous media: Influence of velocity-induced dispersion and concentration-dependent diffusion *Phys. Fluids,* 2004*, 16*, 3592

Riviere, B., Wheeler, M. F. and Banas, K.: 2000, Discontinuous Galerkin methods applied to a single phase flow in porous media, *Computational Geosciences* 4, 337-49.

Roe, P. & Sidilkover, D. Optimum Positive Linear Schemes for Advection in Two and Three Dimensions *SIAM Journal of Numerical* Analysis, 1992, 29, 1542-1568

Russell, T. F. and Celia, M. A.: 2002, An overview of research on ELLAM, *Advances in Water Resources* 25, 1215-31.

Sammon, P.H.: 2003, Dynamic grid refinement and amalgamation for compositional simulation, SPE 79683, presented at the SPE Reservoir Simulation Symposium, Houston, Texas.

Schneider, G. & Raw, M. A Skewed, Positive Influence Coefficient Upwinding Procedure For Control-Volume-Based Finite-Element Convection-Diffusion Computation *Numerical Heat Transfer, Part A: Applications,* 1986*, 9*, 1-26

Shubin, G. & Bell, J. An analysis of the grid orientation effect in numerical simulation of miscible displacement *Computer methods in applied mechanics and engineering,* 1984*, 47*, 47-71
Sweby PK. 1984. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Num. Anal.* 21:995–1011

Tadmor E, Tang T. 2001. Pointwise error estimates for relaxation approximations to conservation laws. *SIAM J. Math. Anal.* 32:870-886.

Thiele, M.R., Batycky, R.P., and Blunt, M.J.: 1997, A Streamline Based 3D Field-Scale Compositional Reservoir Simulator, SPE 38889, presented at the SPE Annual Technical Conference, San Antonio, TX.

Thiele, M. and Batycky, R. P.: 2006, Using streamline-derived injection efficiencies for improved waterflood management, SPE 84080, *SPE REE*, in press.

Thomas, G.W. and Thurnau, D.H.: 1983, Reservoir simulation using an adaptive implicit method, *SPE Journal*, October 1983, 759-768.

Todd, M.; O'Dell, P. & Hiraski, G. Methods for increased accuracy in numerical reservoir simulators *SPE Journal,* 1972*, 12*, 515-530

Valenti, G, Jessen, K. Mallison, B.T. and Gerritsen, M.G.: 2004, High-order upwind schemes for three-phase multi-component flows: A preliminary investigation, SPE 90594, presented at the SPE Annual Technical Conference and Exhibition, Houston, Texas.

Valenti, G.: 2005, Streamline-based simulation of three-phase multi-component flows in porous media, MSc thesis, Department of Petroleum Engineering, Stanford University.

Van Leer, B.: 1979, Towards the Ultimate Conservative Finite Difference Scheme, V: A Second Order Sequel to Godunov's Method, *Journal of Computational Physics* 32, 101.

Van Ransbeeck, P. & Hirsch, C. Deconinck, H. & Koren, B. *(ed.)* A General Analysis of 2D/3D Multidimensional Upwind Convection Schemes *Vieweg,* 1997

Wang, H., Zhao, W. and Ewing, R. E.: 2005, A numerical modeling of multi-component compressible flows in porous media with multiple wells by an Eulerian-Lagrangian method, *Computing and Visualization in Science* 8 (2), 69-81.

Wang, P., Balay, S., Sepehrnoori, K., Wheeler, J., Abate, J., Smith, B. and Pope, G. A.: 1999, A fully implicit parallel EOS compositional simulator for large scale reservoir simulation, SPE 5188, presented at the SPE Reservoir Simulation Symposium, Houston, Texas.

Younis R, Gerritsen M. 2006. *Multiscale process coupling by adaptive fractional stepping: An in-situ combustion model*. Presented at the Soc. Petrol. Eng. Symp. on Improved Oil Recovery, Tulsa, Oklahoma.

National Energy Technology Laboratory

626 Cochrans Mill Road
P.O. Box 10940
Pittsburgh, PA 15236-0940

3610 Collins Ferry Road
P.O. Box 880
Morgantown, WV 26507-0880

One West Third Street, Suite 1400
Tulsa, OK 74103-3519

1450 Queen Avenue SW
Albany, OR 97321-2198

539 Duckering Bldg./UAF Campus
P.O. Box 750172
Fairbanks, AK 99775-0172

Visit the NETL website at:
www.netl.doe.gov

Customer Service:
1-800-553-7681