

# On Front Tracking for Compressible Flow

Halvor Møll Nilsen

Knut-Andreas Lie

October 6, 2008

## **Abstract**

Streamline simulation is particularly efficient for incompressible two-phase flow, for which one can use front tracking to solve the 1-D transport problems along streamlines. Here we investigate the extension of this method to compressible (and immiscible) flow and discuss some of the difficulties involved, and in particular the choices one has in writing the 1-D transport equation(s). Our study is motivated by the simulation of CO<sub>2</sub> injection, and we therefore also develop methods that are particularly suited for solving compressible flow where one phase is incompressible. Altogether, we present four front-tracking methods that are based on a combination of solving ordinary Riemann problems and Riemann problems with discontinuous flux.

## Introduction

Fast and robust methods are crucial for modelling and simulating of flow in porous media, in particular because many of the physical and geological parameters are uncertain or unknown. One therefore often needs to simulate a large set of possible realisations. Streamline methods (Datta-Gupta and King, 2007) are gaining in popularity and provide fast desktop reservoir simulation of large reservoir models or multiple realisations to a wide range of users. Traditionally, streamline simulation has been based upon simplified physics, but recent advances have demonstrated its potential as a fully-fledged alternative to conventional simulators for compressible three-phase or component flows (Crane et al., 2000; Thiele et al., 1997). A key step in all streamline simulators is the solution of 1-D transport equations along streamlines. For incompressible flow, one may utilise an unconditionally stable and highly efficient front-tracking method that operates independent of the underlying irregular time-of-flight (tof) grid. For compressible flow, on the other hand, previous efforts have all been based on the use of standard finite-volume methods, possibly in combination with a regularisation of the tof-grid. In the following, we will (for the first time) present front-tracking methods that can be used also for compressible flow. As their incompressible counterpart, the new front-tracking methods are Lagrangian and represent piecewise constant saturations in terms of a set of evolving discontinuities. However, unlike in the incompressible case, the methods need to keep the underlying tof-grid to represent the divergence of the velocity, which is typically nonzero.

## Mathematical Models

We consider immiscible two-phase flow with no capillary forces, which can be modelled by two continuity equations and Darcy's laws. To simplify the presentation, we assume that the rock is incompressible and that there are no gravity forces. We pick the fluid saturations  $S_i$  and the pressure  $p$  as our primary unknown, introduce the total Darcy velocity  $v = -K\lambda\nabla p$ , and write the governing equations in the so-called fractional-flow form

$$\phi \frac{\partial S_i}{\partial t} + v \cdot \nabla f_i(S_i) + \phi c_i S_i \frac{\partial p}{\partial t} + \frac{1}{\rho_i} f_i(S_i) \nabla \cdot (\rho_i v) = q_i, \quad i = 1, 2, \quad (1)$$

$$\phi(c_1 S_1 + c_2 S_2) \frac{\partial p}{\partial t} + \nabla \cdot v - (c_1 \lambda_1 + c_2 \lambda_2) \frac{v K^{-1} v}{\lambda^2} = q_1 + q_2. \quad (2)$$

Here  $\phi$  is porosity and  $K$  is the permeability. For phase  $i$ ,  $\rho_i$  is the density,  $\lambda_i$  is the mobility,  $c_i = \frac{1}{\rho_i} \frac{\partial \rho_i}{\partial p}$  is the compressibility, and  $q_i$  represents fluid sources. Finally,  $\lambda = \lambda_1 + \lambda_2$  is the total mobility and  $f_i = \lambda_i/\lambda$  the fractional flow of phase  $i$ . In the case when one phase (say phase 1) is incompressible, the corresponding saturation equation simplifies to

$$\phi \frac{\partial S_1}{\partial t} + v \cdot \nabla f_1(S_1) + f_1(S_1) \nabla \cdot v = q_1 \quad (3)$$

In streamline simulation (Datta-Gupta and King, 2007), one uses a sequential splitting to decouple the system (1)–(2), in which one first fixes  $S_i$  and solves (2) on an Eulerian grid to obtain  $p$  and  $v$ , which are in turn held fixed as the saturations  $S_i$  are advanced a time-step  $\Delta t$  along a discrete set of streamlines (Lagrangian grid). In the sequential splitting, one should note that the pressure equation (2) represents conservation of volume. If this is not taken into account when setting up the saturation step (e.g., as in Cheng et al. (2006)), one cannot guarantee that  $0 \leq S_i \leq 1$ . For instance, with one incompressible phase, a naïve splitting that keeps  $\nabla \cdot v$  constant will immediately give saturations outside the unit interval because we do not impose the physical restriction that the divergence of  $v$  is identically zero when only the incompressible phase is present. To make sure that the transport equations always conserves volume, we can use the pressure equation (2) to eliminate  $\nabla \cdot v$  from (1). The resulting equation can be written on non-conservative form as

$$\phi \frac{\partial S_i}{\partial t} + v \cdot \nabla f_i(S_i) + g_i(S, x) = q_i, \quad (4)$$

where the new source terms

$$g_i(S, x) = \phi \frac{\partial p}{\partial t} (c_i S_i - [c_1 S_1 + c_2 S_2] f_i(S_i)) + \frac{v K^{-1} v}{\lambda^2} (-c_i \lambda_i + [c_1 \lambda_1 + c_2 \lambda_2] f_i(S_i)) \quad (5)$$

sum to zero. In addition,  $g_i(0, x) = g_i(1, x) \equiv 0$  and this, together with  $f_i(0) \equiv 0$ , guarantees that (4) has the property that  $0 \leq S_i \leq 1$ .

### Streamline Formulations

Streamlines are a family of curves that are everywhere tangential to the instantaneous flow field. In the absence of gravity and capillarity forces, there is no mass transfer between individual streamlines, meaning that each streamline can be viewed as an isolated flow system. By introducing the time-of-flight  $\tau$ , given by  $v \cdot \nabla \tau = \phi$ , and the operator identity  $v \cdot \nabla = \phi \partial_\tau$ , one can reduce the multidimensional transport equations (1), (3), or (4) to a family of 1-D transport equations along streamlines. For instance, (3) reduces to<sup>1</sup>

$$\frac{\partial S}{\partial t} + \frac{\partial f(S)}{\partial \tau} = -f(S) \frac{\nabla \cdot v}{\phi}(\tau). \quad (6)$$

Cheng et al. (2006) used this form for two compressible phases and neglected implicitly the time-derivative of the density,  $c_i S_i \partial_t p$ . Unfortunately, (6) is not conservative for compressible flow because  $v$  is generally not divergence free.

It may therefore be convenient, at least for the case with one incompressible phase, to introduce a volume factor  $\sigma$  to make a divergence-free field, that is,  $\nabla \cdot (v/\sigma) = 0$ . If we now introduce a new spatial coordinate  $\eta$  along each streamline given by  $(v/\sigma) \nabla = \phi \partial_\eta$ , and apply this to (3), we get a form that is particularly good if one is to use a finite-volume scheme,

$$\frac{\partial S}{\partial t} + \frac{\partial}{\partial \eta}(\sigma f(S)) = 0. \quad (7)$$

If we instead use the standard time-of-flight coordinate, the transport equation will not be explicitly conservative, but take the form

$$\frac{\partial S}{\partial t} + \frac{1}{\sigma} \frac{\partial}{\partial \tau}(\sigma f(S)) = 0. \quad (8)$$

### Front Tracking for One Incompressible Phase

To make a very fast numerical method it is usually necessary to exploit special properties of the specific problem. As one such example, we consider the case when one phase is incompressible. Then, it is natural to use (3) rather than (1) as the transport equation since the former is simpler. Notice, however, that this means that the second phase will not be mass conservative.

First, we present a front-tracking method based on (8), which is in non-conservative form and therefore will not be volume conservative. On the other hand, this formulation is the one that is closest to the streamline (and front-tracking) formulation used for incompressible flow. The front-tracking method is based on solving the equation (8) exactly in the case where  $f$  is a piecewise linear function of  $S$  and  $\sigma$  and the initial data  $S_0$  are piecewise constant functions of  $\tau$  on the underlying time-of-flight grid. This Cauchy problem consists of a series of local Riemann problems

$$\partial_t S + \frac{1}{\sigma(\tau)} \partial_\tau(\sigma(\tau) f(S)) = 0, \quad S(\tau, 0) = \begin{cases} S_i, & \tau < \tau_i, \\ S_{i+1}, & \tau > \tau_i, \end{cases} \quad (9)$$

The solution of these Riemann problems is a similarity solution consisting of constant states separated by discontinuities. Assume first that  $\sigma$  is constant. Introduce a local convexification  $f^c$

<sup>1</sup>In streamline simulation, the fluid sources/sinks are modelled as inflow/outflow boundaries rather than source terms. Therefore  $q_i \equiv 0$  henceforth. We also drop the subscripts on  $S$  and  $f$  to simplify notation.

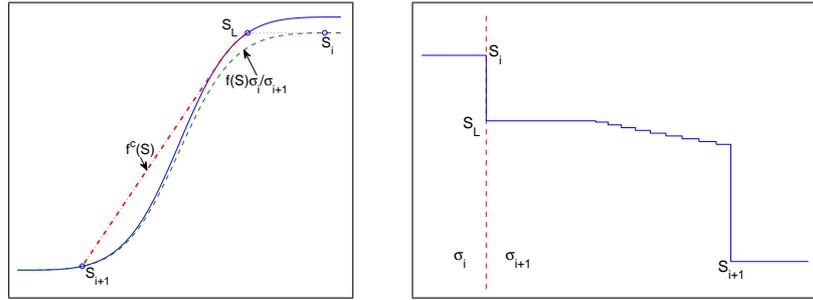


Figure 1: Solution to a discontinuous Riemann problem constructed: the left plot shows the construction and the right plot the resulting self-similar solution consisting of standard Buckley–Leverett profile with left state  $S_L$  and right state  $S_{i+1}$  followed by a stationary discontinuity  $(S_i, S_L)$ .

of  $f$  on the interval  $[S_i, S_{i+1}]$  (see Figure 1), and let  $\{v^0, \dots, v^N\}$  be the nodes of the piecewise linear  $f^c$ . Then the solution of (9) takes the form

$$S(\tau, t) = \begin{cases} v^0 = S_i, & \tau < r_1(t), \\ v^j, & r_j(t) < \tau < r_{j+1}(t), \quad j = 1, \dots, N-1, \\ v^N = S_{i+1}, & \tau > r_N(t), \end{cases} \quad (10)$$

where the space-time rays  $r_j(t)$  satisfy the Rankine–Hugoniot condition

$$\frac{dr_j}{dt} = \frac{f^c(v^{j+1}) - f^c(v^j)}{v^{j+1} - v^j}, \quad r_j(0) = \tau_i. \quad (11)$$

This is the standard front-tracking method used for incompressible flow, where one tracks the rays (henceforth called fronts) until they collide, then solve a new Riemann problem (9), track new fronts, and so on. For compressible flow, we also need to introduce stationary discontinuities where  $\sigma$  is discontinuous. Assume now that  $\sigma$  is discontinuous at  $\tau = \tau_i$  in (9). Then the solution will consist of a stationary discontinuity with left state  $S_i$  and right state  $S_L$  given by  $\sigma_i f(S_i) = \sigma_{i+1} f(S_L)$ , followed by a Riemann fan (10) with left state  $S_L$  (instead of  $S_i$ ) and right state  $S_{i+1}$ . In setting up (8), we have incorrectly assumed that  $\nabla \cdot v$  is constant in time in each cell, see the discussion above. This means that in some cases it may be impossible to find  $S_L$ , e.g., if  $\sigma_i f(S_i)/\sigma_{i+1} > 1$ . We then introduce a regularisation by setting  $S_L = 1$ , which corresponds to introducing a more general solution based on a modified entropy condition (Adimurthi et al., 2007). The overall method will henceforth be referred to as FT1.

The front-tracking method (FT2) for (7) follows the exact same construction, except that we now need to scale the speed of the fronts (11) by  $\sigma$ . Notice that both FT1 and FT2 need no regularisation if  $\nabla \cdot v$  is positive, which will typically be the case for e.g.,  $\text{CO}_2$  injection when the  $\partial_t p$ -term does not dominate the flow equation (2).

### Front Tracking for Two Compressible Phases

Here we will use the large–time-step method introduced by Karlsen et al. (2008), in which the front-tracking method is an essential ingredient. To this end, we rewrite (1) using the streamline coordinate, introduce an internal time-step  $k$ , and evaluate the source term using saturation values from the previous internal time-step,

$$\partial_t S^{n+1} + \partial_\tau f(S^{n+1}) = -\tilde{g}(\tau), \quad \tilde{g}(\tau) = c_1 S^n \partial_t p + (\rho_i \phi)^{-1} f(S^n) \nabla \cdot (\rho_i v). \quad (12)$$

If we now introduce the function  $G(\tau) = \int_0^\tau \tilde{g}(x) dx$ , we may rewrite the above equation as

$$\partial_t (S + G(\tau)) + \partial_\tau (f(S) + G(\tau)) = 0. \quad (13)$$

If we now make a piecewise constant approximation to  $G$ , we can apply a front-tracking scheme that is similar to FT1 and FT2 within each time-step. The only difference is the discontinuous

Table 1: Runtime statistics for the approximate solutions in Figure 2. Runtime for the standard upwind scheme was 85 milliseconds.

	FT1	FT3	FT4
Fronts	2523	2188	3425
Collisions	1051	531	1033
Finding $S_L$	1092	718	1288
Riemann problems	1109	851	1352
Times called	3	12	12
Cpu time(ms)	4.9	3.4	4.6

Riemann problem, for we need to solve the equation  $f(S_i) + G_i = f(S_L) + G_{i+1}$  to determine the stationary jump. If  $G_i > G_{i+1}$  we may have to regularise the solution at the end-point  $S = 1$  and similarly at  $S = 0$  if  $G_i < G_{i+1}$ . The overall scheme, consisting of front-tracking and projection, will be referred to as FT3.

Alternatively, we may base our method on (4), which in streamline coordinates reads

$$\partial_t S + \partial_\tau f(S) = -g(S, \tau), \quad (14)$$

where

$$g(S, \tau) = \left( c_1 S - [c_1 S + c_2(1 - S)]f(S) \right) \partial_t p + \frac{vK^{-1}v}{\phi\lambda^2(S)} \left( -c_1\lambda_1(S) + [c_1\lambda_1(S) + c_2\lambda_2(S)]f(S) \right). \quad (15)$$

The front-tracking for this equation is as in FT3, except that we now update the source term at the end of each internal time-step. The resulting method will be referred to as FT4.

### Numerical Examples

In this section we present three examples to illustrate the qualitative properties of our front-tracking and large-time-step methods. In the first example, we consider injection of a compressible liquid into a more viscous incompressible liquid. This case is motivated by the injection of supercritical CO<sub>2</sub> into an aquifer. In the last two examples, we test the opposite case, where we inject an incompressible phase into a compressible phase that is ten times less viscous (Case 2) and ten times more viscous (Case 3). These cases are representative of water injection into a compressible oil-phase.

The specific features of the three examples are discussed in more detail below. In all cases, the cpu-time of the front-tracking methods are between one and two orders less than the cpu-time of an explicit, upstream-weighted, finite-volume method. The complexity of our front tracking method is now limited by the number of collisions.

**Case 1:** We consider injection of a compressible liquid (phase 2) with viscosity 0.1 into a medium with length 50 that is initially filled with an incompressible liquid (phase 1) with unit viscosity. The flow is driven by a pressure difference given by the Dirichlet boundary conditions of  $p_L = 350$  and  $p_R = 300$ . We use quadratic relative permeability curves and assume a simplified equation-of-state,

$$\rho_i(p) = \rho_i^0 \exp(c_i(p - p_r)), \quad (16)$$

where  $\rho_1^0 = 1$ ,  $\rho_2^0 = 0.6$ ,  $c_1 = 0$ , and  $c_2 = 0.08$ . The pressure equation (2) is discretised using a standard finite-volume method on a grid with 100 uniform cells. To reach time  $t = 12$ , we use three splitting steps. In the front-tracking methods we approximate the flux function(s) using 50 uniform intervals, and for FT3 and FT4 we use four internal time-steps between each pressure update.

Figure 2 shows the saturation profiles and the fronts in the  $(x, t)$ -plane for FT1, FT3, and FT4. All three methods show the ability for large time-steps (the CFL number is around 5 for

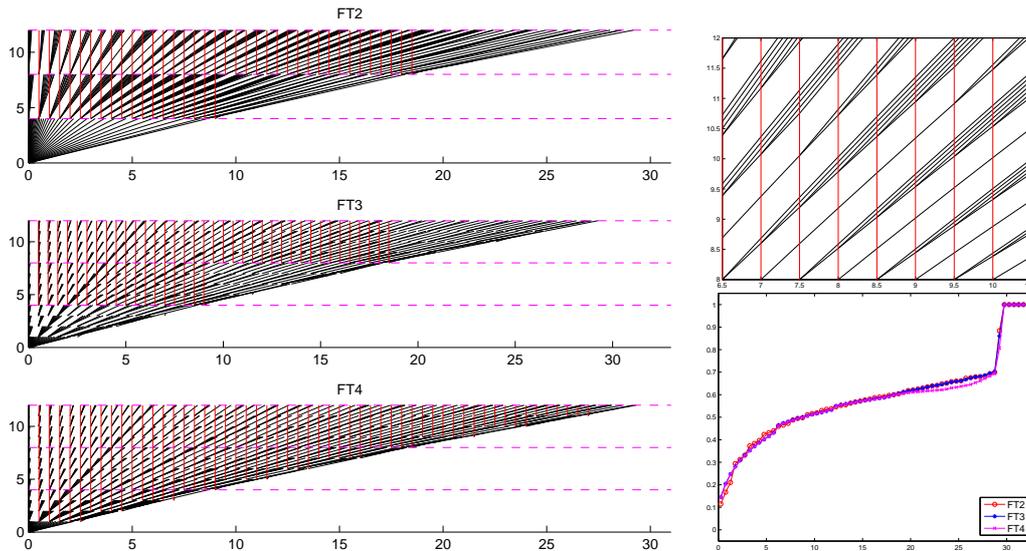


Figure 2: Injection of a compressible liquid into an incompressible liquid (Case 1). The left plot shows fronts in the  $(x, t)$ -plane for FT1, FT3, and FT4, where the global time-steps are indicated by horizontal dashed lines. The upper-right plot shows a zoom of the fronts for FT1. The lower-right plot shows the saturation profiles  $S_1(x, t)$  for all three methods shown on the underlying grid in physical space.

FT3/FT4 and around 20 for FT1) and give reasonable accuracy of the saturation profile. From Table 1, we see that FT3 has the lowest number of fronts, front collisions, Riemann problems, and computations of left states  $S_L$  and hence is the fastest method. On the other hand, this is also the method that has the crudest approximation of compressibility effects. Somewhat surprisingly, FT1 has higher runtime than FT4 despite having a lower number of fronts, Riemann problems, and computations of  $S_L$ . A closer examination of our simple research code, revealed that the cpu-time was dominated by a suboptimal implementation of the priority queue used to keep track of front collisions. This will be rectified in upcoming work.

Let us now consider the fronts in Figure 2 in more detail. Since the time-of-flight parametrisation changes from one time-step to the next, we plot fronts and solution profiles as a function of  $x$  rather than of  $\tau$ . The left plot indicates the difference in which the three methods approximate compressibility effects. For FT1/FT3, compressibility effects are only updated in each pressure step by updating  $\sigma$  and  $G(\tau)$ , respectively, whereas for FT4, the source term  $G(\tau, t)$  is updated after each internal time-step. This is reflected in a larger number of stationary discontinuities (static fronts) and somewhat lower saturation values in the cells near the injection front. Similarly, compared with FT3, we see that FT1 has a larger number of interactions between static fronts and weak fronts representing the refraction wave.

The upper-right plot shows another interesting behaviour, which can be seen for all methods. When a front collides with a static discontinuity, the strength of the dynamic wave increases, and because of the piecewise linear flux function, the wave is split in two (or more) fronts.

**Case 2:** Consider now the opposite situation, where we inject an incompressible phase into a ten times less viscous compressible phase. The saturation of the incompressible phase is still the primary variable together with pressure, but now we use five global pressure steps.

The lower-right plot in Figure 3 shows that FT1 and the standard upwind scheme give almost identical results, except for the numerical diffusion of the upwind scheme. Similarly, we see the importance of the internal steps in FT3 and FT4: with only four internal steps, the front speeds are clearly overestimated. For FT3 and FT4 to give approximately the same accuracy as FT1, we must increase the number of local steps to 20. This will increase the runtime, but both methods are still faster than FT1. In this process, we introduce more projections, which gives a diffusive

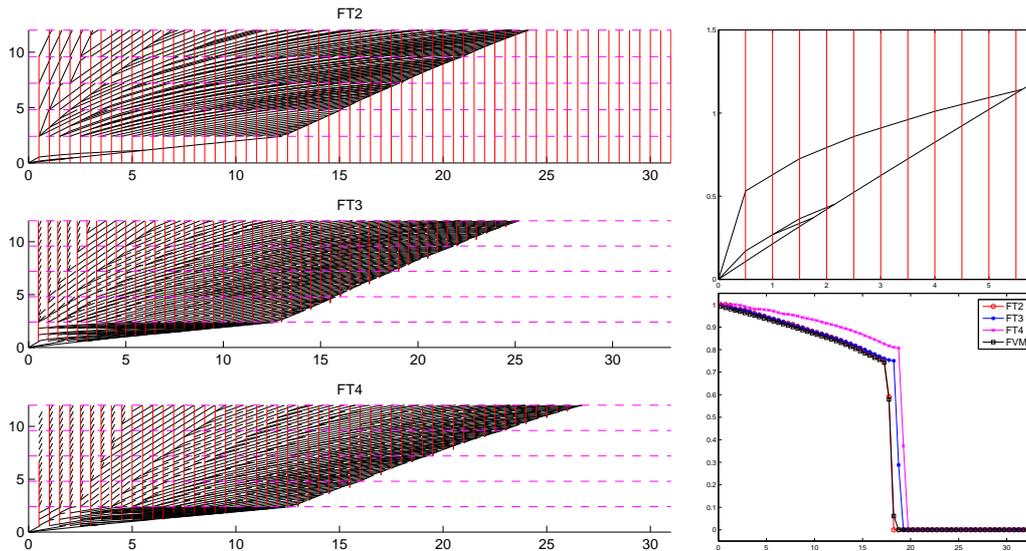


Figure 3: Injection of an incompressible phase into a ten times less viscous compressible phase (Case 2). The left plot shows fronts in the  $(x, t)$ -plane, the upper-right plot shows a zoom around the inlet for FT1, and the lower-left plot compares the front-tracking solutions with the standard upwind scheme on the same grid.

Table 2: Runtime statistics for the approximate solutions in Figure 3. Runtime for the upwind scheme was 3163 milliseconds.

	FT1	FT3		FT4	
Fronts	5576	5571	9210	4996	8569
Collisions	2554	2170	1148	1828	663
Finding $S_L$	2850	2636	4329	2325	3971
Riemann problems	2581	2754	4577	2447	4223
Times called	5	20	100	20	100
Cpu time(ms)	9.6	8.1	8.8	7.5	8.3

effect and to obtain an equally accurate solution as FT1, we would have had to also increase the resolution of the tof-grid.

From the left plot, we see that FT1 has static fronts in the whole domain to represent changes in  $\nabla v$ , while FT3 and FT4 only need to represent changes in  $g(S, \tau)$  appearing behind the injection front. Because fronts are allowed to propagate longer for FT1 than for FT3/FT4, this method has the highest number of fronts, collisions, and Riemann problems, despite having been called fewer times, see Table 2. Observe also that the front-tracking methods are 300–400 times faster than the upwind method. Unlike for the front-tracking methods, the time-step of the explicit upwind method is severely restricted by the high volume-fluxes arising in the unswept zone between the outflow boundary and the injection front, where the liquid has small density and viscosity. This means that the upwind scheme spends a considerable number of operations computing a constant state in the unswept zone, whereas the front front-tracking methods focus their computational effort in the swept zone between the injection front and the inflow boundary.

The upper-right plot in Figure 3 is taken from the first transport step of FT1, and shows a zoom around the inlet and illustrates most of the collisions types that can appear in the front tracking. When a dynamic front collides with a static front it will modify the static front and change speed and strength. The change in strength may cause the front to split if the front is in a region where it represents a refraction wave. When two dynamic fronts collide, we solve an ordinary Riemann problem that results in a new set of dynamic fronts.

For this specific case were the velocity increases in the  $x$ -direction, we clearly see that the fronts go faster on the right-hand side of the domain. Actually, this is not the case when we track

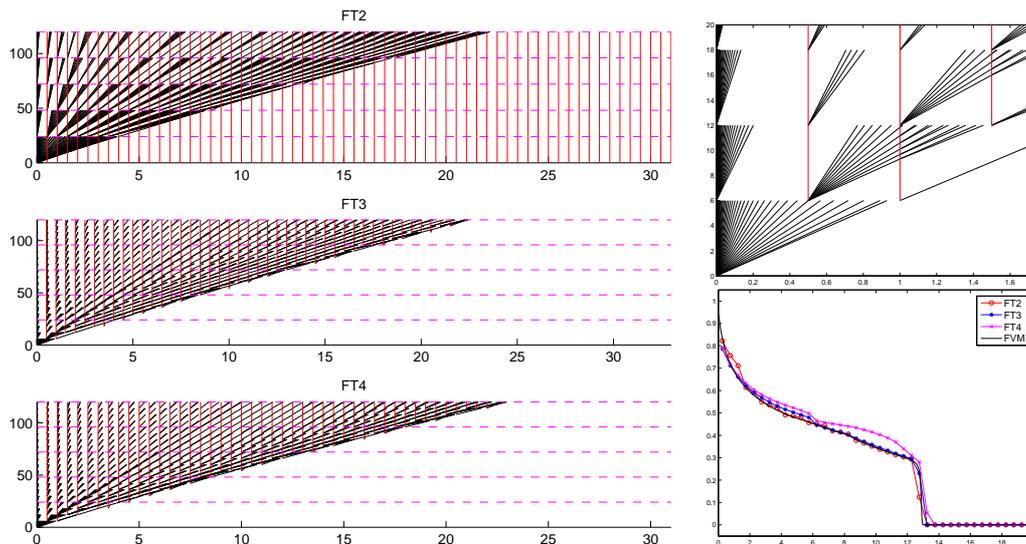


Figure 4: Injection of an incompressible fluid into a ten times more viscous compressible fluid (Case 3). The left plot shows fronts in the  $(x, t)$  plane, the upper-right plot shows a zoom of the fronts for FT3, and the lower-right plot compares the front-tracking solutions with the upwind scheme on the same grid.

the fronts in the tof-coordinate, but the effect appears when the fronts are transformed back to physical space. In principle, the fronts are not straight lines in physical space, but very weakly curved due to the coordinate transformation between time-of-flight and physical space.

**Case 3:** The last example is motivated by enhanced recovery of a heavy oil. We therefore change the viscosity of the second phase to ten, which means that we also must increase the final simulation time by a factor ten.

FT3 and FT4 both project the front-tracking solution onto the underlying tof-grid between the internal steps, which is needed to capture the dynamics of the source term. The upper-right plot of Figure 4 is taken from the first global step of FT3 and clearly shows this effect: First, the front-tracking solution is projected to a piecewise constant function on the tof-grid. Then, at the start of the new internal step the discontinuities at the grid interfaces are found (i.e., between  $S_i$  and  $S_L$ ), before we solve the Riemann problem needed to approximate the evolving wave by a finite number of fronts within each cell. We observe that the different fans correspond to different segment of the refractive wave, and that the piecewise constant approximation of the refraction wave is more uneven compared to FT1 due to the projections. Finally, we clearly see that the slowest speed of each fan increases for each new start point, since the saturation at the point decreases.

The lower-right plot of Figure 4 shows that all methods except FT4 give the same result. FT4 differs a bit because the source term is modified in each local step. A comparison with a more accurate solution shows that the modified source term makes the solution worse for this case. The biggest error appears because of the IMPES splitting that underestimates the velocity and not because we neglect the time-dependence in the source term.

### Summary

We have constructed four front-tracking methods suitable for solving compressible two-phase flow and showed that they work very well compared with the standard upwind method even for regular grids. Given that the front-tracking methods are unconditionally stable, this advantage is expected to increase when the methods are integrated into a 3D simulator, for which the tof-grids tend to be both highly irregular and graded toward fluid sources and sinks.

In addition, we have found a method, FT1, that is simple, efficient, and accurate for the case of one incompressible phase. We believe this method will prove very useful for streamline

Table 3: Runtime statistics for the approximate solutions in Figure 4. Runtime for the upwind scheme is 223 milliseconds.

	FT1	FT3	FT4
Fronts	4561	3119	3202
Collisions	1824	635	619
Finding $S_L$	2133	992	986
Riemann problems	1805	1068	1059
Times called	5	20	20
Cpu time(ms)	8.5	5.4	5.5

simulation of CO<sub>2</sub> injection.

### Acknowledgement

The research was funded by the Research Council of Norway under grant no. 178013. We also thank Dr. Mishra for providing insight into the regularisation of the Riemann problems that are not solvable.

### References

- Adimurthi, Mishra, S., and Gowda, G.D.V. [2007] Conservation law with the flux function discontinuous in the space variable-ii. *J. Comput. Appl. Math.*, **203**(2), 310–344.
- Cheng, H., Osako, I., Datta-Gupta, A., and King, M.J. [2006] A rigorous compressible streamline formulation for two and three phase black oil simulation. *SPE J.*, **11**(4), 407–417.
- Crane, M., Bratvedt, F., Bratvedt, K., Childs, P., and Olufsen, R. [2000] A fully compositional streamline simulator. In: *SPE Annual Technical Conference and Exhibition*. Dallas, Texas, SPE 63156.
- Datta-Gupta, A. and King, M.J. [2007] *Streamline Simulation: Theory and Practice*, volume 11 of *SPE Textbook Series*. Society of Petroleum Engineers.
- Karlsen, K.H., Mishra, S., and Risebro, N.H. [2008] A large time-stepping scheme for balance equations. *J. Engrg. Math.*, **60**(3–4), 351–363.
- Thiele, M.R., Batycky, R.P., and Blunt, M.J. [1997] A streamline-based 3D field scale compositional reservoir simulator. In: *SPE Annual Technical Conference and Exhibition*. San Antonio, Texas, SPE 38889.