Fully-Implicit Higher-Order Schemes Applied to Polymer Flooding

T. S. Mykkeltvedt* (IRIS) K.-A. Lie (SINTEF) X. Raynaud (SINTEF)

June 16, 2016

Abstract

In water-based EOR methods, active chemical or biological substances are added to modify the physical properties of the fluids or/and the porous media at the interface between oil and water. The resulting displacement processes are governed by complex interplays between the transport of chemical substances, which is largely linear and highly affected by numerical diffusion, and how these substances affect the flow by changing the properties of the fluids and the surrounding rock. These effects are highly nonlinear and highly sensitive to threshold parameters that determine sharp transitions between regions of very different behavior. Unresolved simulation can therefore lead to misleading predictions of injectivity and recovery profiles.

Use of higher-order spatial discretization schemes have been proposed by many authors as a means to reduce numerical diffusion and grid-orientation effects. Most higher-order simulators reported in the literature are based on explicit time stepping, and only a few are implicit. One reason that fully-implicit formulations are not widely used might be that it becomes quite involved to compute the necessary linearizations for modern high-resolution discretizations of TVD and WENO type. Herein, we solve this problem by using automatic differentiation.

As an example of EOR, we consider polymer flooding, which involves complex and adverse phenomena like adsorption in the rock, degradation and in-situ chemical reactions, shear thinning/thickening, dead pore space, etc. Using a few idealized test cases, we compare and contrast explicit and fully-implicit time stepping for a variety of high and low-resolution spatial discretizations.

Introduction

In many oil recovery processes, water is injected to maintain reservoir pressure and to force the oil towards production wells. Water is less viscous than oil, thus the water front can develop viscous instabilities and finger through the reservoir. As a consequence, much of the oil can be left behind as residual or non-recoverable oil (Gao, 2011). Enhanced oil recovery (EOR) is essential to improve oil recovery and to increase a field's potential. In water-based EOR methods, the physical properties of the fluids and the surrounding rock can be modified through active chemical or biological substances (Lake, 1989). The transport of these substances is largely linear and therefore highly affected by numerical diffusion. Furthermore, the subsequent effects on the fluids and the surrounding rock are highly nonlinear and sensitive to threshold parameters that determine transitions between regions of very different behavior. Thus, the displacement process is complex and challenging to simulate. Unresolved simulation can lead to misleading predictions of injectivity and recovery profiles.

As an example of water-based EOR, we consider polymer flooding, which improves both the local displacement efficiency and the areal sweep (Lake, 1989; Sorbie, 1991). Injecting dissolved polymer increases the water viscosity and enhances its ability to push oil through the rock because of a more favorable mobility ratio between the injected and displaced fluids. This effect is most pronounced when mobility ratios are unfavorable, e.g., in reservoirs with heavy oil. Polymer molecules dissolved in water can also improve the areal sweep efficiency by reducing channeling through high flow zones and through viscous cross-flow between layers of different permeability. Water viscosity is strongly affected by the polymer concentration, and it is therefore crucial to capture polymer fronts sharply to resolve the nonlinear displacement mechanisms correctly. Polymer fronts will in the worst case be linear waves and generally have significantly less self-sharpening effects than water fronts. This poses a challenge when using standard low-order methods, whose large numerical diffusion tend to smear the polymer bank and hence fail to accurately resolve the EOR effect. To overcome this challenge the use of higher-resolution spatial discretization schemes have been discussed by many authors, e.g., Bell et al. (1989); Holing et al. (1990); Khan et al. (1995); Kaibara and Gomes (2001); Chertock et al. (2009); Trangenstein (2009). These discretizations are developed to maintain high-order accuracy on smooth parts of the solution and at the same time minimize the creation of spurious oscillations around discontinuities.

The predominant approach to industry-standard reservoir simulation is to use a fully-implicit discretization. On one side, this takes care of the coupling between fluid pressure and transport of phases and components. More important, it presents an efficient means to treat short time constants coming from high local flow rates (e.g., in near-well regions), cells with small pore volumes, etc. With a few exceptions, see e.g., Blunt and Rubin (1992); Liu et al. (1995), most higher-order simulators reported in the literature are based on explicit time stepping. One reason why fully-implicit formulations are not widely used together with high-resolution discretizations such as total variation diminishing (TVD) and weighted essentially non-oscillatory (WENO) schemes, is that these introduce strong nonlinearities that can be difficult to linearize correctly. To overcome this challenge, we herein propose to use automatic differentiation, e.g., as implemented in the open-source MRST software (Krogstad et al., 2015; Lie, 2015; MRST, 2015b). This ensures that no analytical derivatives have to be programmed explicitly, and this tool is important for implicit methods since it both reduces implementation time and risk of errors.

In this work, we first review several high-resolution spatial discretizations and discuss how these can be combined with sequential explicit and fully-implicit time stepping. Through a series of idealized test cases we then compare and contrast the resulting high-order schemes with standard first-order schemes based on upstream mobility weighting. We also argue why implicit discretization is beneficial and generally required. In particular, we show that using a second-order reconstruction (and improved spatial quadrature) will counteract the numerical dissipation imposed by the temporal discretization in a fully-implicit setting and ensure that we can maintain displacement profiles that are significantly sharper than what can be computed with a first order scheme. When the dissipation introduced by the numerical scheme is larger than the physical diffusion, grid-orientation errors will occur. This has been discussed by several authors, see e.g., Chen et al. (1993). Using a high-resolution spatial discretization significantly diminish these errors.

Mathematical Model

Polymer flooding is described by an immiscible, two-phase model with three fluid components (oil, water and polymer) that incorporates many of the fluid effects found in contemporary commercial simulators. This includes adsorption of polymer onto the reservoir rock, reduction in permeability, mixing of polymer in water, and compressibility of fluids and rock. Inaccessible pore space is left out for brevity, but could have easily been included. Moreover, we do not model pseudoplastic effects of the diluted polymer solution which may cause shear-thinning or shear-thickening behavior.

The conservation equations for oil, water, and polymer are given by

$$\frac{\partial}{\partial t}(\rho_\alpha \phi s_\alpha) + \nabla\cdot(\rho_\alpha \mathbf{u}_\alpha) = 0, \qquad \alpha = o,w \tag{1a}$$

$$\frac{\partial}{\partial t}\left[\rho_w \phi s_w c + \rho_r(1-\phi_{\text{ref}})\bar{a}\right] + \nabla\cdot(c\rho_w \mathbf{u}_{wp}) = 0. \tag{1b}$$

Here, ρ_α, s_α, and $\mathbf{u}_\alpha$ denote density, saturation, and velocity of phase α, whereas c is the polymer concentration. For simplicity, we assume that capillary pressure can be neglected so that we henceforth can write $p_o = p_w = p$ and use p as a primary unknown along with $s = s_w$ and c. The function $\bar{a}$ models the amount of polymer adsorbed onto the rock, ϕ is the porosity, ϕ_{ref} and ρ_r are the reference porosity and reference density of the rock. Simple PVT behavior is commonly modeled through inverse formation-volume factors $b_\alpha(p) = b_\alpha \rho_\alpha^S$, where ρ_α^S is the surface density of phase α. To avoid overloading the notations, we will not insert those in the equations.

To form a complete model, we also assume that oil and water occupy the entire pore space, $s_o + s_w = 1$, and use Darcy's law to relate the phase velocities $\mathbf{u}_\alpha$ of oil and water, and the velocity $\mathbf{u}_{wp}$ of water containing polymer, to gradients of pressure. For oil, this reads

$$\mathbf{u}_o = -\mathbf{K}\frac{k_{ro}(s_o)}{\mu_o}(\nabla p - \rho_o g \nabla z), \tag{2}$$

where $\mathbf{K}$ is the absolute rock permeability, k_{ro} and μ_o are the relative permeability and the viscosity of oil, g is the gravitational constant, and z is the coordinate in the vertical direction. The adsorption of polymer onto the rock will introduce a resistance to flow that reduces the effective permeability experienced by water containing diluted polymer. This is modeled by dividing the absolute permeability by a non-decreasing function $R_k(c)$. Pure water and water containing diluted polymer are in general not fully miscible but we will not model in detail this behavior and rather use the Todd and Longstaff (1972) model that upscales the complex patterns of viscous fingers and models the viscosity change of the mixture in terms of effective viscosities. Whereas the viscosity of oil is assumed to be constant, the viscosity of the other two fluid components is assumed to depend upon the polymer concentration. This gives us Darcy equations of the form,

$$\mathbf{u}_w = -\underbrace{\frac{k_{rw}(s_w)}{\mu_{w,\text{eff}}(c)R_k(c)}}_{\lambda_w(s,c)}\underbrace{\mathbf{K}(\nabla p - \rho_w g \nabla z)}_{\mathbf{v}_w(p)}, \qquad \mathbf{u}_{wp} = -\underbrace{\frac{k_{rw}(s_w)}{\mu_{p,\text{eff}}(c)R_k(c)}}_{\lambda_{wp}(s,c)}\underbrace{\mathbf{K}(\nabla p - \rho_w g \nabla z)}_{\mathbf{v}_w(p)}, \tag{3}$$

where we assume that the pressure and density are independent of polymer, and that the relative permeability does not depend on mixing.

The degree of mixing of polymer into water comes in through the mixing parameter $w \in [0,1]$ that generally depends on the heterogeneity of the porous medium, the displacement scenario, etc. Let $\mu_m = \mu_m(c)$ be the viscosity of a *fully mixed* polymer solution, then the effective polymer viscosity is defined as

$$\mu_{p,\text{eff}} = \mu_m(c)^w \mu_p^{1-w}, \tag{4}$$

where $\mu_p = \mu_m(c_{\max})$. Furthermore, the viscosity of the *partially mixed* water is given by

$$\mu_{w,\text{e}} = \mu_m(c)^w \mu_w^{1-w}.$$

Finally, the effective water viscosity is found by interpolating linearly between the inverse of the effective polymer viscosity and the partially mixed water viscosity

$$\frac{1}{\mu_{w,\text{eff}}} = \frac{1-c/c_{\max}}{\mu_{w,\text{e}}} + \frac{c/c_{\max}}{\mu_{p,\text{eff}}}. \tag{5}$$

Numerical Discretization

To find a numerical approximation of s, c and p, the spatial domain is subdivided into a finite number of finite volumes. In the following, we only consider regular Cartesian grids in two spatial dimensions, and use the integral form of the conservation equations (1) to compute approximations to the *discrete cell averages*,

$$q_i(t) = \frac{1}{|\Omega_i|}\iint\limits_{\Omega_i} q(x_1,x_2,t)\,\mathrm{d}\mathbf{x}. \tag{6}$$

Here, q denotes one of the primary variables p, s, or c and $\Omega_i = [(i_1-\frac{1}{2})\Delta x_1,(i_1+\frac{1}{2})\Delta x_1]\times[(i_2-\frac{1}{2})\Delta x_2,(i_2+\frac{1}{2})\Delta x_2]$ is the grid cell number $i=(i_1,i_2)$. Let Γ_{ij} denote the interface between cells i and j having normal vector $\mathbf{n}_{i,j}$ pointing from Ω_i to Ω_j. If we pick the water equation, for instance, the discretized equation can be written in compact form as

$$\big[\rho_w(p_i)\phi(p_i)s_i\big]^{n+1} = \big[\rho_w(p_i)\phi(p_i)s_i\big]^{n} - \frac{\Delta t}{|\Omega_i|}\sum_{|i-j|=1}\int_{\Gamma_{ij}}\big[\rho_w(p)\,\lambda_w(s,c)\big]_{ij}^{m}\big(\mathbf{v}_w^m\cdot\mathbf{n}\big)_{i,j}\mathrm{d}s. \tag{7}$$

Here, using $|i-j| = |i_1-j_1|+|i_2-j_2| = 1$ ensures that the sum only runs over neighbors that share a common face. The temporal discretization is specified by setting $m=n$ for an explicit scheme and $m=n+1$ for an implicit scheme. The integrand denotes the mass flux *evaluated at the interface* and how to compute this term will be the focus for most of the following discussion. For brevity, we henceforth drop the subscript w and superscript m.

First-order schemes

In our discretization of the mass flux, we treat pressure differently from saturation and concentration. The pressure is assumed to be constant within each grid cell and to find the density at the interface Γ_{ij}, we use a simple average, so that $\rho_{ij} = \frac{1}{2}(\rho(p_i)+\rho(p_j))$. Likewise, for the flux $v_{i,j} = \mathbf{v}_{ij}\cdot\mathbf{n}_{i,j}$, we use a standard two-point approximation; that is, we write

$$v_{i,j} \approx \Big[T_{i,j}^{-1}+T_{j,i}^{-1}\Big]^{-1}\big(p_i-p_k\big), \qquad T_{i,j} = \frac{\mathbf{K}_i(\mathbf{x}_{ij}-\mathbf{x}_i)\cdot\mathbf{n}_{i,j}}{|\mathbf{x}_{ij}-\mathbf{x}_i|^2}, \tag{8}$$

where $\mathbf{x}_i = (x_{i_1},x_{i_2})$ denotes the centroid of cell Ω_i and $\mathbf{x}_{ij}$ is the centroid on Γ_{ij}. Summarizing, we have introduced the following approximation of the pressure-dependence

$$\int_{\Gamma_{ij}}\rho_{ij}(p)\lambda_{i,j}(s,c)\,\big(\mathbf{v}\cdot\mathbf{n}\big)_{i,j}\mathrm{d}s \approx \tfrac{1}{2}\big[\rho(p_i)+\rho(p_j)\big]v_{i,j}\int_{\Gamma_{ij}}\lambda(s,c)\,\mathrm{d}s. \tag{9}$$

To evaluate the remaining integral, we need to make three different choices that will determine our numerical scheme: (i) which quadrature rule to use for the integral in (9), (ii) how to *reconstruct* the necessary point values at the quadrature points from the cell-averages s_i and c_i, and (iii) how to approximate the mobility given point values that generally are different on opposite sides of the interface Γ_{ij}.

To get a first-order scheme, it is sufficient to use the midpoint rule for the integral and assume a constant reconstruction so that we get one-sided point values $s_{i,j}^{-} = s_i$ and $s_{j,i}^{+} = s_j$, and similarly for the concentration. Given these values, there are several ways we can evaluate the integrand. Herein, we use simple upstream evaluation for all our schemes, i.e.,

$$\lambda_{i,j}(s,c) = \begin{cases}\lambda(s_i,c_i), & \text{if } v_{i,j}\geq 0,\\ \lambda(s_j,c_j), & \text{otherwise.}\end{cases} \tag{10}$$

Solving the discretized systems

If we now plug (9) and (10) into (7), and repeat the same derivation with the obvious modifications for the oil and the polymer equations, we end up with a system that can be written on residual form as

$$\mathbf{F}(\mathbf{y}) = \mathbf{0}, \tag{11}$$

where $\mathbf{y} = [\mathbf{p}, \mathbf{s}, \mathbf{c}]$ contains all the unknown cell-averaged values for pressure, saturation, and concentration for the new time-step. Even if we use an explicit time integration, this system is nonlinear because of the fluid and rock compressibilities. To solve the system, we use a standard Newton–Raphson method: Assume that we have an initial guess $\mathbf{y}_0$, we write $\mathbf{y} = \mathbf{y}_0 + \delta\mathbf{y}$ and solve

$$\mathbf{0} \approx \mathbf{F}(\mathbf{y}_0) + \mathbf{J}\delta\mathbf{y}, \tag{12}$$

where $\mathbf{J} = d\mathbf{F}/d\mathbf{y}$, to determine the increment $\delta\mathbf{y}$. This process is repeated until the residual $\mathbf{F}$ or the increment $\delta\mathbf{y}$ is sufficiently small in some suitable norm. This process exhibits quadratic convergence under certain requirements on the smoothness and differentiability of $\mathbf{F}$. These requirements cannot be satisfied for an arbitrary time step Δt. For the explicit time discretization, the scheme is only stable if the time step satisfies a standard CFL condition. Likewise, it is common to introduce some mechanism in the implicit scheme that cuts the time step until a satisfactory convergence is achieved; we will return to this discussion in the numerical experiments. Another practical challenge with (12) is how to compute the Jacobian matrix $\mathbf{J}$, which may contain quite intricate nonlinear dependencies, in particular when the scheme is extended to higher order. Analytical derivation and subsequent coding of the Jacobian can be very time-consuming and prone to human errors. To alleviate this problem, we propose to use automatic differentiation as implemented in the open-source MRST software (Krogstad et al., 2015; Lie, 2015). The key idea of automatic differentiation is that the computation of the residual $\mathbf{F}$ can be broken down to a (nested) sequence of elementary function evaluations. Each elementary function evaluation is simple to differentiate analytically and can easily be coded into a software using operator overloading so that the overloaded function evaluation computes the function value *and its derivative(s)*. Nested function evaluations are taken care of by using the chain rule. As a result, all we need to do is to code the evaluation of the residual, and then the software calculates $\mathbf{J}$ automatically. We will therefore not discuss the structure and computation of $\mathbf{J}$ in more detail for any of the schemes.

Second-order slope-limiter schemes

We start by introducing the following fourth-order Gaussian quadrature rule to evaluate the edge integral in (9),

$$\int_{-1/2}^{1/2} \lambda(x)\mathrm{d}x = \frac{1}{2}\left[\lambda\left(\frac{-1}{2\sqrt{3}}\right) + \lambda\left(\frac{1}{2\sqrt{3}}\right)\right]. \tag{13}$$

Then, the only remaining part is to describe how to reconstruct the necessary one-sided point values. Using a piecewise linear reconstruction gives second-order formal accuracy on smooth solutions, quadratic gives third-order, and so on. Most reconstructions found in the literature are introduced for a scalar entity, and it is tacitly assumed that they can be extended in a component-wise manner to vector-valued entities. Another question is which quantities to reconstruct. Herein, we have chosen to reconstruct the primary physical variables s and c in the integration points and then use these to evaluate the mobility at the interface.

For a second-order scheme, a linear reconstruction is given by

$$\tilde{q}(\mathbf{x}) = q_i + \sigma_i^1(x_1 - x_{i_1}) + \sigma_i^2(x_2 - x_{i_2}), \tag{14}$$

where the slopes σ_i^1 and σ_i^2 can be estimated from the discrete difference in cell-average values between neighboring cells. To avoid creating spurious oscillations near discontinuities, the slopes σ_i must be limited through a nonlinear dissipation mechanism that adds stabilizing numerical diffusion near discontinuities while preserving formal order on smooth parts. This is achieved through so-called slope-limiters that were first introduced in a series of papers by van Leer (1974, 1977). Let Φ be a nonlinear averaging function that is capable of choosing the slope. This function is called a limiter and is applied independently in each spatial direction. Given Φ, the slope σ_1^1 is set to

$$\Delta x_1 \sigma_i^1 = \Phi\left(q_i - q_{i-(1,0)}, q_{i+(1,0)} - q_i\right). \tag{15}$$

Herein, we will consider three different limiters,

$$\Phi^{mm}(a,b) = \frac{1}{2}\big(\mathrm{sgn}(a) + \mathrm{sgn}(b)\big)\ \min\big(|a|,|b|\big), \tag{16}$$

$$\Phi^{sb}(a,b) = \frac{1}{2}\big(\mathrm{sgn}(a) + \mathrm{sgn}(b)\big)\ \max\big(\min(\theta|a|,|b|), \min(|a|,\theta|b|)\big), \tag{17}$$

$$\Phi^{vL}(a,b) = \big(b + a\frac{|b|}{|a|}\big)/\big(1 + \frac{|b|}{|a|}\big). \tag{18}$$

The *minmod limiter* Φ^{mm} compares the upwind and downwind slopes and chooses the one that is smaller in magnitude. If the slopes have different signs, the slope is set to zero. This gives a robust limiter that adds as much numerical dissipation as possible, while keeping second-order accuracy (Sweby, 1984). The *superbee* limiter Φ^{sb} (Roe, 1986) represents the other extreme end in the sense that it chooses the slope to be as steep as possible and hence introduces very little numerical dissipation. When used as part of an *implicit* discretization, both limiters have the disadvantage that they are not smooth functions of their arguments: Φ^{mm} has one kink, whereas Φ^{sb} has three. These kinks will adversely affect the Newton–Raphson solver used to solve the system of discrete equations. As a third alternative, we have therefore included the smooth *van Leer* limiter Φ^{vL}, which reconstructs steeper slopes than the minmod limiter but gentler slopes than the superbee limiter.

WENO schemes

The main design principle behind the slope-limiter technology is to ensure that the resulting schemes give approximate solutions having diminishing total variation when applied to a scalar equation. This principle cannot be extended beyond second order and instead one tries to construct an essentially non-oscillatory (ENO) solution, i.e., a solution in which oscillations do not grow significantly with time. To understand the key idea, we will look at a simplified example. Assume that we want to reconstruct a function $q(x)$ inside a cell i based on cell averages q_{i-1}, q_i and q_{i+1}. To this end, we define two linear polynomials $q^{\pm}(x)$ based on q_i and $q_{i\pm 1}$ and a quadratic polynomial $q^c(x)$ based on all three cell averages. The classical ENO idea (Harten et al., 1987) is to choose the one of the three polynomials that gives the least oscillatory solution. To aid this choice, the method uses a measure of local smoothness of the given data that is based on divided differences. Whereas this switching of stencils seems to work well for explicit schemes, it introduces discontinuity in the Jacobian matrix for implicit schemes and may cause severe flip-flopping and general lack of convergence in the nonlinear iterations (Knudsen, 2014). In the so-called weighted ENO (WENO) schemes (Liu et al., 1994), the key idea is to instead use a convex combinations of the three polynomials $w^- q^-(x) + w^c q^c(x) + w^+ q^+(x)$, where $w^- + w^c + w^+ = 1$. The weights w are designed such that they reproduce the optimal polygonal approximation if the solution is smooth inside the overall stencil, but tends to zero if the corresponding local stencil contains a discontinuity.

Herein, we use a simplified version of the WENO idea based on four linear reconstructions, which gives a scheme that is formally only second-order accurate. To this end, let us define a polynomial

$$q^{NE}(\mathbf{x}) = q_i + \sigma_i^E(x_1 - x_{i_1}) + \sigma_i^N(x_2 - x_{i_2}), \qquad \sigma_i^E = \frac{q_{i+(1,0)} - q_i}{\Delta x_1}, \qquad \sigma_i^N = \frac{q_{i+(0,1)} - q_i}{\Delta x_2}, \tag{19}$$

and a corresponding smoothness indicator,

$$\beta_i^{NE} = \tfrac{1}{4}\left[\left(\Delta x_1 \sigma_i^E\right)^p + \left(\Delta x_2 \sigma_i^N\right)^p + \varepsilon\right]^{-\ell}, \tag{20}$$

where ε is a suitable small parameter. Here, we set $\varepsilon = 10^{-7}$ and $\ell = 2$. Similar polynomials and smoothness indicators are defined for NW, SE, and SW. Then we can define the weights as $w^{NE} = \beta^{NE}/(\beta^{NE}+\beta^{NW}+\beta^{SE}+\beta^{SW})$, and use the following polynomial to reconstruct point values at the integration points,

$$q_i(\mathbf{x}) = \sum_{\nu=NE,NW,SE,SW} w^{\nu} q^{\nu}(\mathbf{x}). \tag{21}$$

In our implicit schemes, we have chosen to define the nonlinear weights based on the unknown solution at time $n+1$. This contributes to enhance the nonlinearity of the resulting scheme. An alternative, and possibly more efficient, approach would be to define the nonlinear weights based on the previous iterate or the values from the previous time step; see e.g., Gottlieb et al. (2006).

Numerical Experiments

In this section, some numerical experiments are performed to compare and contrast the various schemes described above. Unless stated otherwise, we assume a homogeneous, isotropic permeability, which has been quite arbitrarily set to 100 md and the porosity is taken to be 0.2. In all test cases below, we use the fluid data described in Figure 1. The densities of oil and water are taken to be 962 and 1080 kg/m^3, the oil and water viscosity are 5 and 0.48 cP, and a Todd–Longstaff mixing parameter of $w = 1$ corresponding to full mixing is used in all experiments. Reduction in permeability proved to have a small effect on our numerical experiments, and without lack of generality, we have therefore set $R_k(c) = 1$ for simplicity in (3). In some of the examples, we report computational times to indicate the computational cost of the various schemes. These numbers are likely afflicted by many artifacts resulting from our relatively simple implementation in MATLAB and should therefore merely be interpreted as rough estimates of relative performance.

Viscosity multiplier	
c	$\mu_m(c)/\mu_w$
0	1
0.5	3
1.0	6
1.5	12
2.0	24
3.0	48

Adsorption	
c	$\bar{a}(c)$
0	0
0.25	0.000012
0.50	0.000016
0.75	0.000019
1.00	0.000020
1.25	0.000021
1.50	0.000023
1.75	0.000025
3.00	0.000025

Compressibility	
Phase	Value [bar^{-1}]
rock	$3.00 \cdot 10^{-5}$
water	$4.28 \cdot 10^{-5}$
oil	$6.65 \cdot 10^{-5}$

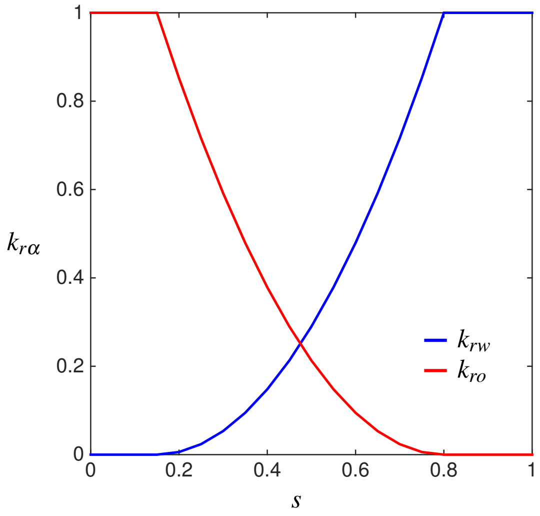

Figure 1 *Fluid parameter used for the numerical experiments. The relative permeabilities $k_{r\alpha}$ are formed by linear interpolation of tabulated data.*

1D example with high porosity contrast

We start by considering a simple one-dimensional case to illustrate key ideas and motivate the use of implicit discretization. Water with polymer is injected at a given rate from the left ($x = 0$) and pressure is imposed on the right ($x = L$). The grid has 100 equally spaced cells. By using a 1D model and neglecting compressibility, we can rewrite the flow equations in Buckeley–Leverett form, which somewhat simplified reads

$$\frac{\partial}{\partial t}\left[\begin{matrix} s \\ sc + a(c) \end{matrix}\right] + \frac{u}{\phi}\frac{\partial}{\partial x}\left[\begin{matrix} f(s,c) \\ m(c)f(s,c) \end{matrix}\right] = 0, \qquad f(s,c) = \frac{\lambda_w(s,c)}{\lambda_w(s,c)+\lambda_o(s)},$$

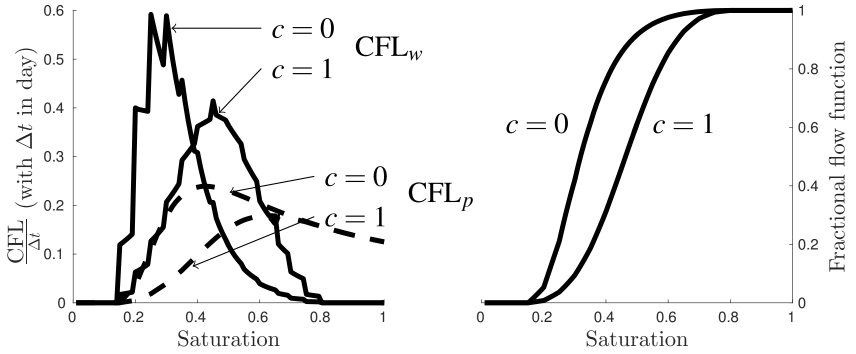

Figure 2 *On the left, illustration of the CFL number for the s-waves (solid lines) and c-waves (dashed lines) as functions of saturation for two polymer concentrations, $c = 0$ and $c = 1$. The apparent piecewise linearity of the CFL curve comes from the fact that our data parameters are tabulated. We observe that there are several local extrema, meaning that we have a fractional flux function with several inflection points. The left plot shows the fractional flow curve, where a higher polymer concentration results in a translation of the curve to the right, so that water break-through is delayed and recovery improved.*

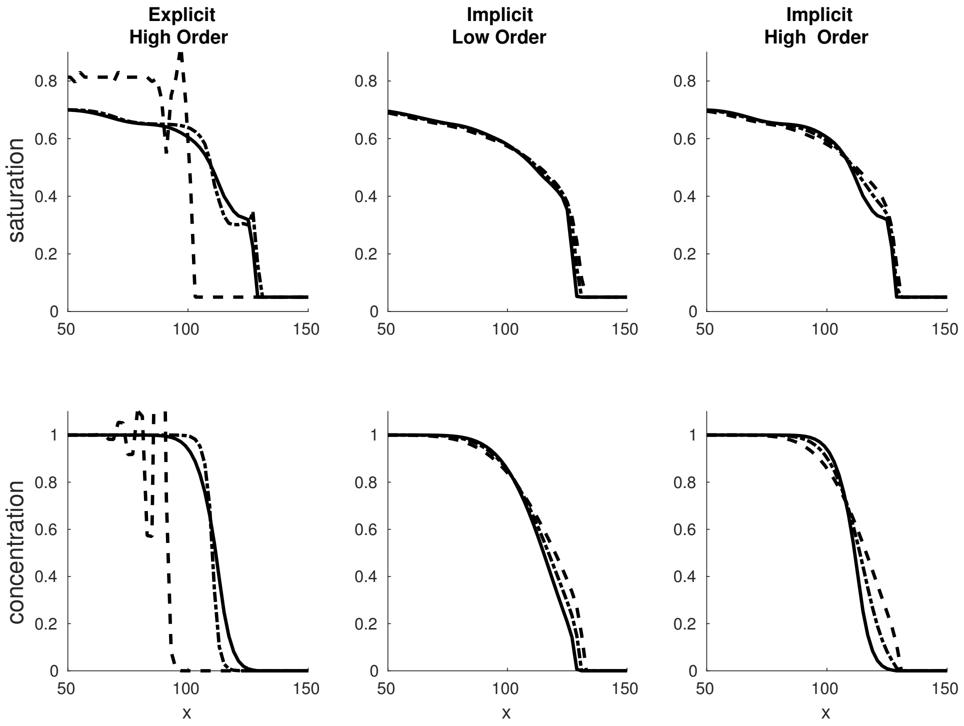

Figure 3 *Plot the of saturation and concentration at the last time step, zoom on the front region, which has reached half of the simulation domain. The values are plotted for three different time discretization step, Δt equal to 0.01 days (solid line), 1 day (dash-dot), and 3 days (dashed)*

where the total flux u is constant and $m(c)$ denotes the ratio $\mu_{w,\text{eff}}/\mu_{p,\text{eff}}$. This hyperbolic system, $Q_t + \frac{u}{\phi}F(Q)_x = 0$, has two families of waves, an s-wave in which only the water saturation changes, and a c-wave in which both water saturation and polymer concentration vary. Let r_s and r_c denote the corresponding eigenvalues of $F'(Q)$. Then, we can estimate the CFL number of the wave families as (see Figure 2)

$$,\text{CFL}_s = \frac{u\Delta t}{\phi \Delta x} r_s(s,c), \quad \text{CFL}_c = \frac{u\Delta t}{\phi \Delta x} r_c(s,c).$$

Figure 3 reports approximate solutions computed by the explicit and implicit schemes with different time steps. For the explicit scheme, a time step of three days exceeds the CFL limit by far and hence we get an unstable, oscillatory solution. A time step of one day is slightly above the CFL limit and hence we get a slight instability near the water front but very sharp resolution of the polymer front. By decreasing the time step further, we get rid of the instability but smear the polymer front. As expected, no instabilities are observed for the implicit scheme. However, only the second-order accurate solver captures the trailing polymer front.

Problems with uniform petrophysical variables are seldom encountered in real life, and to illustrate why implicit discretization is more useful in practice, we consider a conceptual heterogeneous problem in which the porosity is reduced by a factor M in the interval $[0.1L, 0.2L]$. To analyse this case, we transform to time-of-flight coordinates (τ, t), where $\tau = x\phi/u$, and henceforth assume that $u = 1$ without loss of generality. To compare how the low-porosity region affects the explicit and implicit discretizations, we consider a simple advective wave, $q_t + q_\tau = 0$, for which the effective numerical equations for the first-order schemes are given as $q_t + q_\tau = \frac{1}{2}(\Delta\tau \pm \Delta t)q_{\tau\tau}$, with positive sign for the implicit scheme and negative sign for the explicit scheme. This means that a discontinuity propagating over a period t in time will be smeared to a width $\mathscr{O}\big(\sqrt{t(\Delta\tau \pm \Delta t)}\big)$. Using a hand-waving argument, we can say that the total numerical smearing experienced by a linear discontinuity as it propagates through the domain is proportional to

$$\underbrace{\frac{9\phi}{10}\big(\Delta x\phi \pm \Delta t\big)}_{\text{high-porosity region}} + \underbrace{\frac{\phi}{10M}\Big(\frac{\Delta x\phi}{M} \pm \Delta t\Big)}_{\text{low-porosity region}}.$$

The numerical smearing is clearly dominated by the high-porosity region for both schemes. For the explicit scheme, the time-step is restricted by the fast flow in the low-porosity region, i.e., $\Delta t \propto \phi\Delta x/M$. Reducing the porosity by a factor M means that we not only are forced to take M times as many time steps, but also end up with significantly more numerical smearing. With implicit temporal discretization, the numerical dissipation decreases with decreasing time step. However, since the overall smearing is dominated by the high-porosity region, we can safely use a large CFL number in the low-porosity region and instead choose time step so that we achieve acceptable smearing in the high-porosity region. This will drastically reduce the number of time steps, and this gain in computational effort can enable us to introduce high-resolution spatial discretization and/or improved spatial resolution.

In Figure 4, we consider a case with pure water flooding simulated with a stable time step and a larger time step that is stable in the high-porosity region but exceeds the CFL condition in the low-porosity region. As expected, oscillations appear in the latter case but disappear as time evolves, in part because of the self-sharpening inherent in the non-convex flux function and in part because of the much stronger numerical smearing experienced once the oscillations propagate into the high-porosity region. We hasten to emphasize that a CFL number above the stability limit was used for illustration purposes only, and that we by no means try to recommend this sort of dangerous practice. With the implicit scheme, we get stable solutions for both time steps, as expected. Figure 4 also shows that the displacement profile is somewhat sharper for smaller time step, but the improved resolution is hardly significant.

Increasing the order of the discretization naturally improves the solution. This is shown for the implicit scheme in Figure 5 for a case of polymer flooding. In Table 1, we also report a comparison of the explicit and implicit schemes in terms of performance. Since the explicit schemes need small time steps to satisfy the CFL condition in the low-porosity region, they introduce a large amount of numerical dissipation in the high-porosity regions. Hence, both implicit schemes give almost as sharp displacement fronts as the explicit scheme, whereas the gain in computational cost is a factor five for the standard upwind method and a factor two for the high-resolution method.

Quarter five-spot: time stepping

In the next example, we consider the classical quarter five-spot test problem, which consists of an injector and a producer placed diagonally opposite of each other in a square domain with no-flow conditions set on all boundaries. Injection is modeled as a source term with constant injection rate and production is modelled as a boundary condition with fixed pressure. Figure 6 shows solutions just after water breakthrough computed with the first-order scheme (constant reconstruction) and with minmod, van Leer, su-

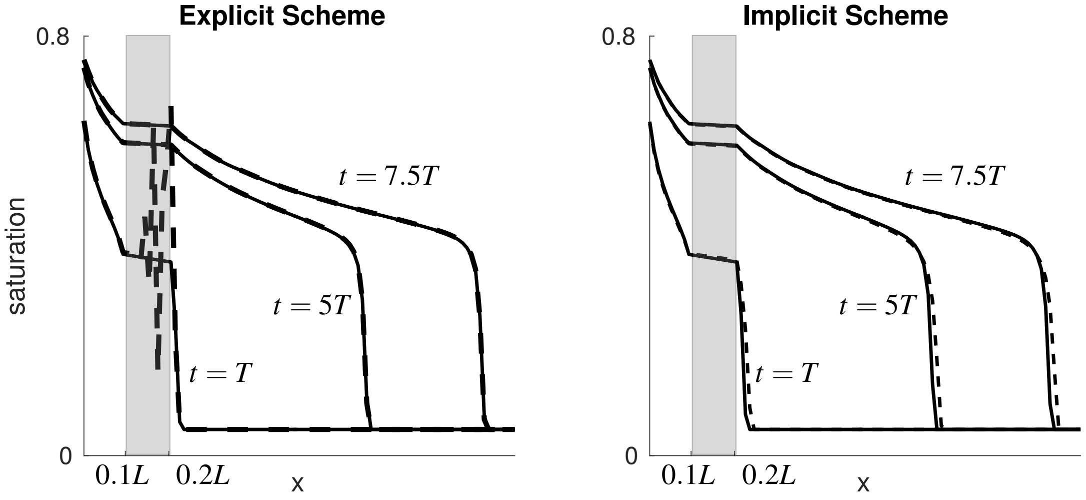

Figure 4 *Comparison plots for the case with two region with different porosity ($\phi = 0.01$ for $x \in [0.1L, 0.2L]$ and $\phi = 0.2$ otherwise). Two different time steps are used, $\Delta t = 0.01$ day (solid lines) and $\Delta t = 0.1$ (dashed lines). For simplicity, no polymer is considered in this case*

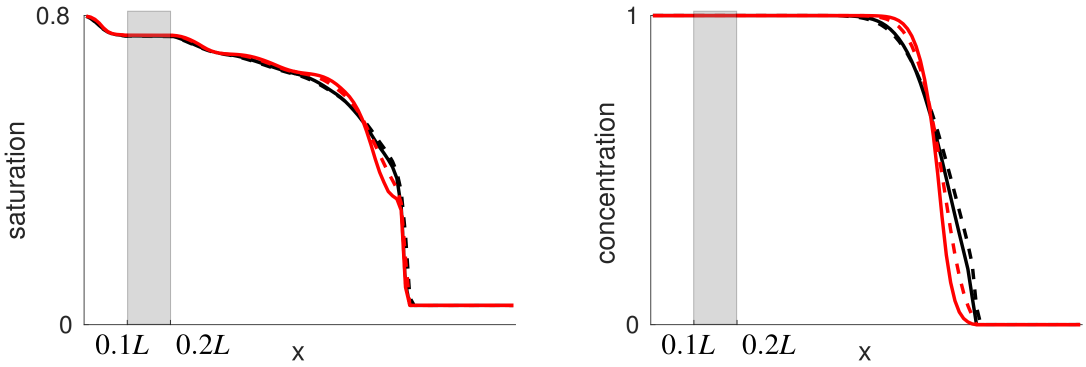

Figure 5 *Comparison between implicit high-order (red) and low-order (black) schemes for time steps equal to $\Delta t = 0.1$ day (solid line) and $\Delta t = 1$ day (dashed line).*

Table 1 *Comparison between implicit and explicit methods for the case with two different porosities. We give the total number of linear solves required for each simulation as well as the total simulation time.*

Simulation case	Linear solves	CPU time
$\Delta t = 0.1$ (day), explicit, 1st order	15840	513s
$\Delta t = 0.1$ (day), explicit, 2nd order	16308	688s
$\Delta t =$ 1 (day), implicit, 1st order	3018	117s
$\Delta t =$ 1 (day), implicit, 2nd order	2529	326s

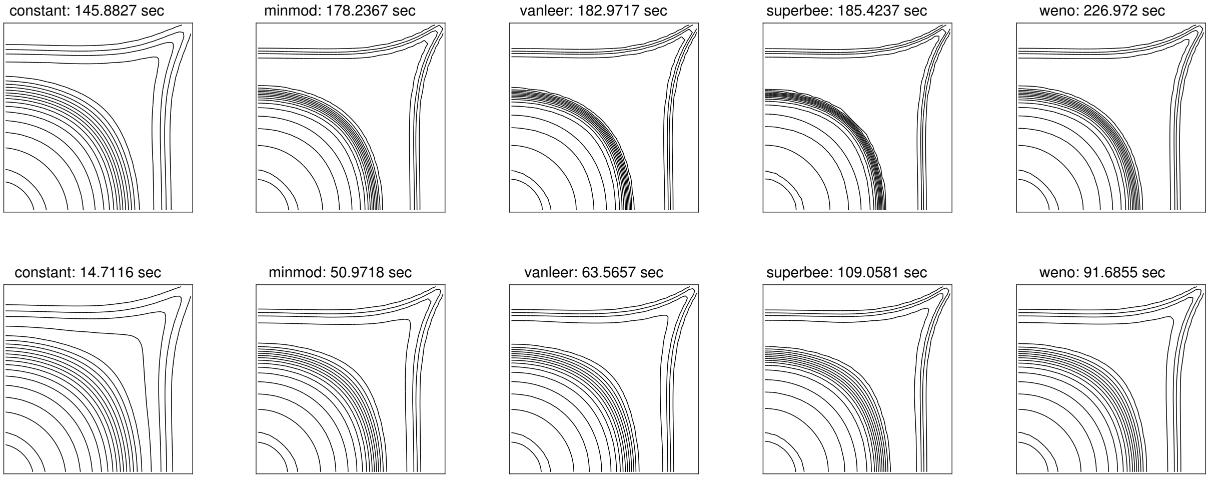

Figure 6 *Comparison of saturation profiles on a quarter five-spot problem computed by explicit schemes with 1600 time steps and implicit schemes with 50 time steps on a uniform Cartesian grid with* 50×50 *cells. Notice how the leading water front is sharper resolved than the trailing polymer front.*

perbee, and WENO reconstructions. The explicit version uses 1600 equal time steps, which corresponds roughly to a unit CFL number. In this scheme, the phase fluxes are computed based on saturation and concentration values reconstructed at the beginning of the time step, whereas the pressure-dependence of the total Darcy flux is treated implicitly. Our computational setup includes fluid compressibility and thus we generally end up with nonlinear residual equations. In IMPES and other similar sequential solution strategies, the pressure (and total Darcy fluxes) and the saturation/concentrations are computed in sequence in separate steps. Here, we solve the fully coupled residual system directly using a Newton method with a residual tolerance set to 10^{-3} times the maximum strength of the source sink. As a result, the explicit schemes typically need to perform more than one nonlinear iteration in the first time steps to account for the initial pressure transient. Figure 7 reports the number of iterations required by all five schemes. The first-order scheme requires three iterations in the first step, two iterations in the next 28 steps, and a single iteration thereon. When second-order reconstructions are added, the overall system becomes more nonlinear and coupled, and two iterations are required for a longer initial period. The plots indicate that the sharper a scheme resolves discontinuities, the longer the period will be before numerical diffusion makes the displacement profiles so smooth that the Newton iterations converge in one iteration. The superbee limiter gives steeper slopes than van Leer, which in turn gives steeper slopes than minmod. WENO will generally not construct very steep slopes at discontinuities unless we increase the exponent ℓ in the smoothness indicator. (Notice that WENO $\to$ ENO as $\ell \to \infty$.)

For the implicit schemes, we use a time-step control built into MRST. In its simplest form, this time-step controller takes a set of time step targets (control steps) and desired number of nonlinear iterations as input. Here, we have used five iterations as our target and a step target that consists of 50 equally spaced time steps. To avoid a large initial error, the first implicit time step is replaced by a $(\frac{1}{32}, \frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2})$ subdivision to form a gradual ramp-up. To stay within the upper limit of five nonlinear iterations, the controller may reduce the time step so that any control step is chopped into several substeps. If several substeps have been computed with less than five iterations, the controller will try to increase the time step so that the iterations increase towards five again. Figure 8 reports the substeps and the corresponding number of iterations for each of the five implicit schemes, whereas Figure 9 shows the corresponding cumulative number of iterations. For the first-order scheme, the first step of the ramp-up sequence requires five iterations, whereas the remaining steps only need three iterations or less. The scheme is therefore able to complete all controls steps without any chopping. With the minmod limiter, we get the same behavior for the first three ramp-up steps. In the fourth step, the number of iterations increases to four and hence the fifth step is chopped slightly, giving a tiny substep that is hardly visible on the figure. In the sixth and last ramp-up step we therefore get a succession of gradually larger substeps. However, the scheme is not yet able to take the full control step and keeps chopping the

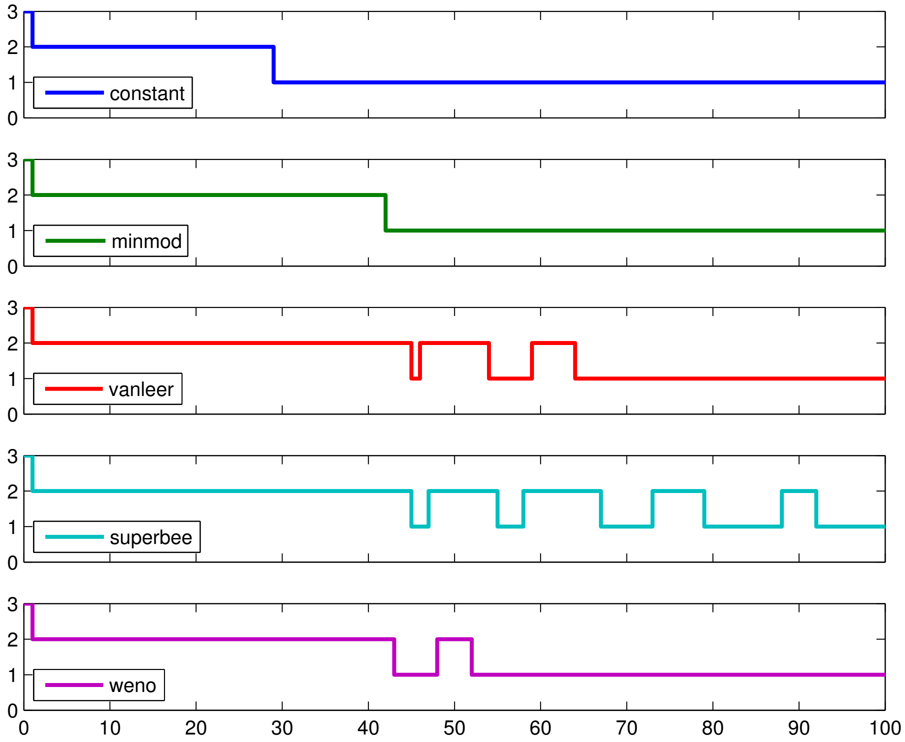

***Figure 7** Number of iterations during the first 100 time steps for the explicit schemes in Figure 6.*

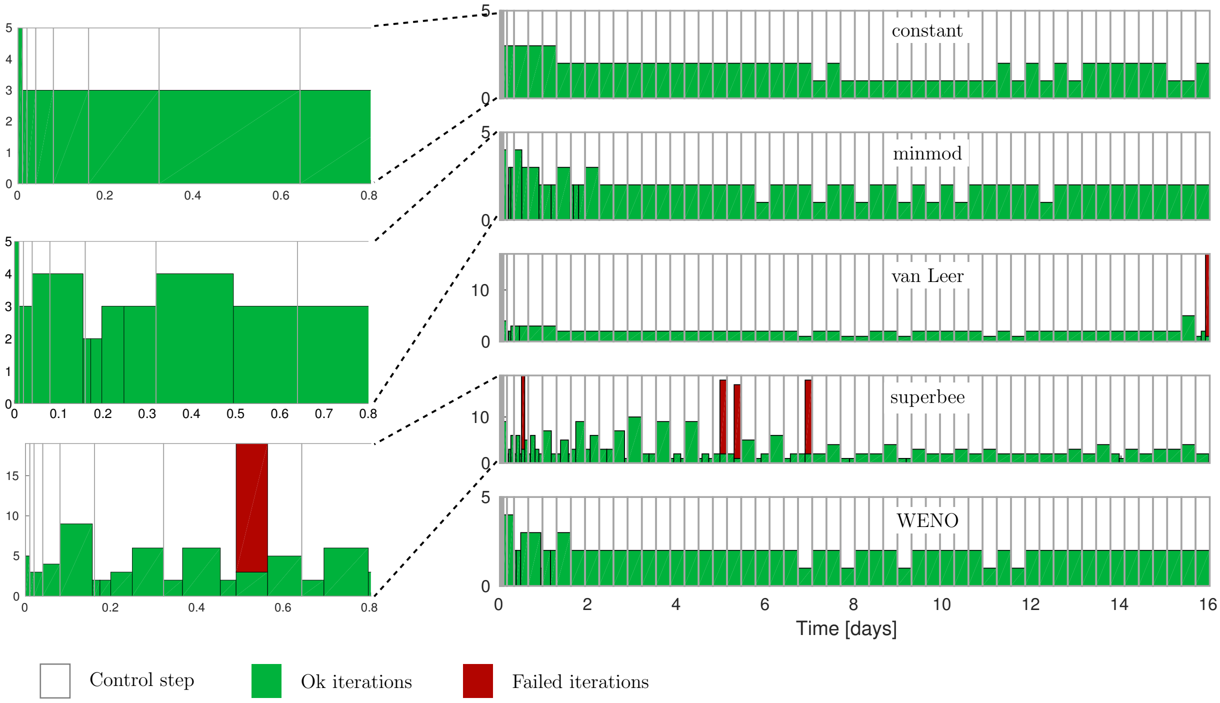

***Figure 8** Number of nonlinear iterations for the five implicit schemes from Figure 6. The bar graph to the right shows the number of iterations per targeted time step (control step) and how these have been divided into substeps to ensure that the number of iterations per step (green bars) do not exceed five. Steps that have not converged within 15 iterations are considered to fail and illustrated by a red bar.*

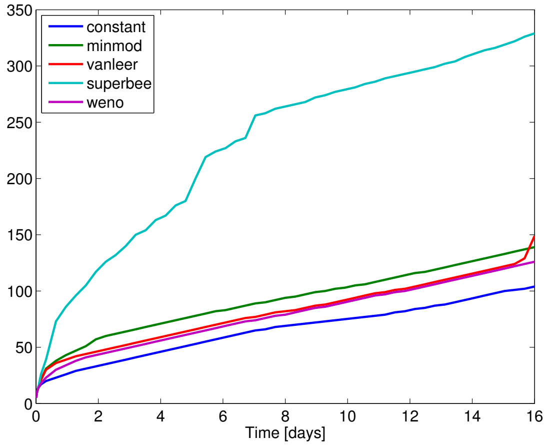

Figure 9 *Cumulative number of nonlinear iterations for the five implicit schemes from Figure 6.*

control steps in two until we reach approximately two days. The WENO and van Leer schemes have similar behavior; both are smoother than the minmod limiter (which has one kink) and hence require less iterations throughout the simulation period, except in the last time step, where the van Leer scheme fails to converge as water breaks through in the producer. For the highly compressive superbee limiter, which has three kinks, the Newton solver struggles more and altogether requires 2.4 as many iterations in total compared with the minmod limiter. The number of iterations, and the fact that computing a second-order reconstruction is more expensive than the constant reconstruction, explains the difference in runtimes for the first and second-order schemes. The high runtime for the WENO scheme is explained by an inefficient implementation that contains a large number of redundant function evaluations.

Quarter five-spot: spatial and temporal convergence

Figure 6 confirms that using second-order reconstruction and improved spatial quadrature gives more accurate solution profiles for the explicit schemes, as expected. For the implicit schemes, the improved resolution is somewhat masked by increased numerical dissipation introduced by the large time steps, but also in this case the leading water front and the following chemical front are resolved more sharply by the second-order schemes. To investigate the spatial accuracy more systematically, Figure 10 shows the result of a grid refinement study in which we compare the implicit first-order scheme with the corresponding second-order scheme using the minmod limiter. The plots show that as a simple rule of thumb, the second-order scheme provides (at least) as good resolution as we would get from the first-order scheme on a 2×2 refined grid with twice as many time steps. Figure 11 reports a similar refinement study with respect to the time step. Here, we see that the implicit discretization contributes significantly to numerical dissipation; compare the explicit solution with the implicit solution for CFL=1. We also see that we can safely increase the CFL number for the implicit scheme to one order of magnitude beyond the stability limit for the explicit scheme before the increased numerical dissipation causes a significant widening of the computed displacement fronts.

Five spot: grid-orientation errors

The idealized quarter five-spot test case corresponds to an infinite reservoir produced by a symmetric pattern consisting of four injectors surrounding a producer that is repeated to infinity in each direction as illustrated in Figure 12. With a standard two-point spatial discretization, as used in the first-order scheme, any displacement front will preferentially move along the axial directions of the grid and this will introduce grid-orientation errors. To assess how increasing the order of the scheme affects these errors, we compare solutions to the five-spot problem computed using the quarter five-spot and the rotated five-spot setups shown Figure 12. To get comparable spatial resolution, the rotated grid is set to have approximately twice as many grid cells, i.e., $\sqrt{2}$ as many grid cells in each axial direction. In the original setup, the preferential flow along the grid axes will tend to overestimate the frontal movement

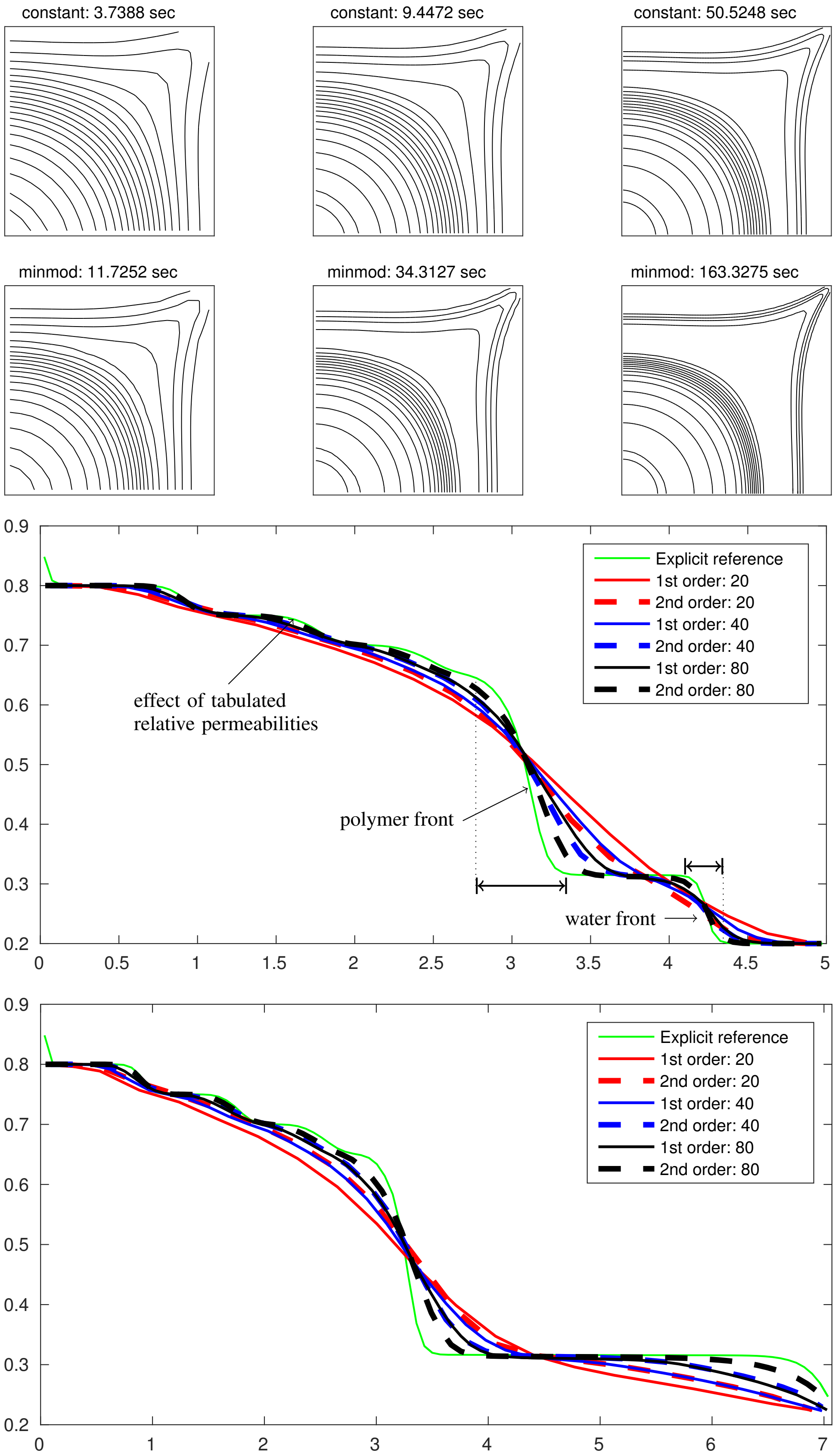

Figure 10 *Convergence study for the implicit first-order scheme (upper row) and the implicit minmod scheme (second row) on a sequence of* $n \times n$ *grids with* m *uniform control steps for* $n, m = 20, 40, 80$. *The two bottom rows show solution profiles along the x-axis and along the diagonal* $x = y$*, respectively. The leading water front is resolved much sharper than the trailing polymer front by most schemes.*

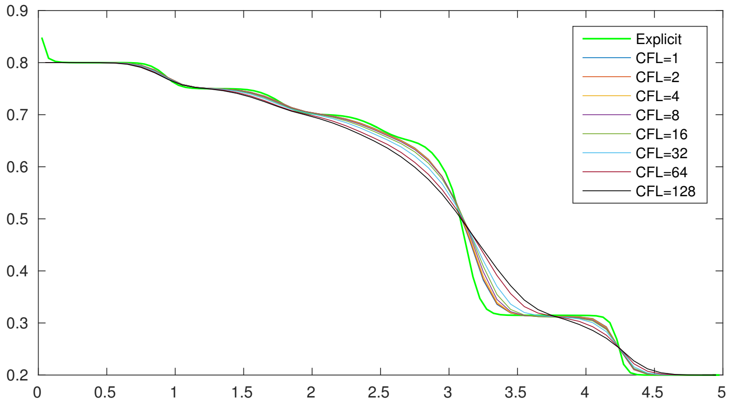

Figure 11 *Convergence study for the implicit minmod scheme on a* 50×50 *grid with decaying time steps. The solutions are sampled along the x-axis and compared with a solution computed by the explicit minmod scheme on a* 100×100 *grid.*

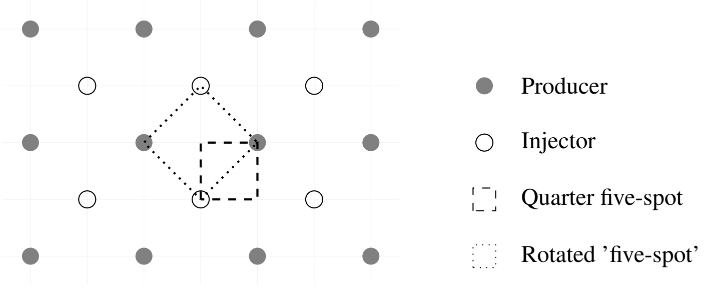

Figure 12 *Well setup for the five-spot test cases used to study grid-orientation errors.*

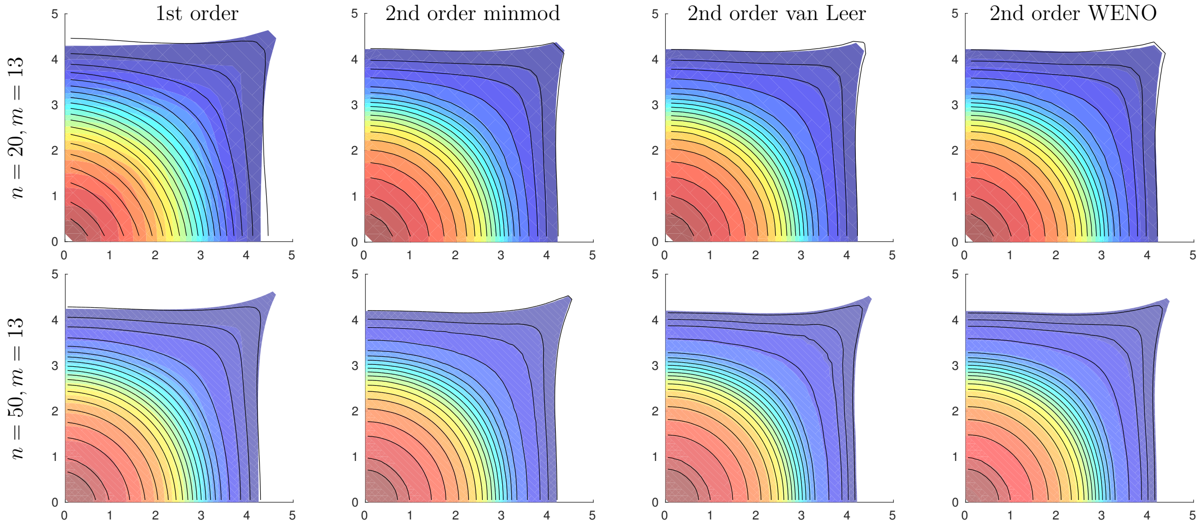

Figure 13 *Test of grid-orientation errors for the five-spot setup on a n × n grid with m uniform control steps. Colors shows the solution computed on the rotated setup and solid lines the solution computed on the original setup.*

into stagnant regions and underestimate the movement of the displacement front in the high-flow zones along the diagonal by smearing the tip of the finger. For the rotated setup, the preferential flow direction is from injector to producer and hence the two-point scheme will tend to overestimate the movement of the front in the high-flow zone and underestimate its movement towards the stagnant zones. Figure 13 confirms that using a second-order scheme counteracts these grid-orientation errors in the sense that the solutions are almost identical for the original and rotated computational setups. Since the underlying grids are uniform Cartesian, the solutions computed on the original and rotated geometries will converge towards each other with increased spatial (and temporal) resolution.

Channelized medium

In the last test case, we consider a model of a channelized 500×1000 m^2 reservoir represented on a uniform 100×100 grid. Mock-up petrophysical data are generated by sub-sampling the “Strebelle 2002” training image from (Mariethoz and Caers, 2016) and assigning a porosity of 0.3 or 0.2 and a isotropic permeability of 1 darcy or 100 md to the foreground and background cells, respectively. This creates a mild heterogeneity effect so that the displacement front will move somewhat faster in the channel than in the background. Water with polymer is injected at the south edge of the reservoir at a rate proportional to the porosity in each cell and fluids are produced at the north edge.

Figure 14 shows approximate solutions computed by the implicit and explicit schemes with constant, minmod, van Leer, and WENO reconstructions. For the explicit schemes we use a time step of 4 days to get to the final simulation time of 5 years, and for the implicit schemes we used control step of 100 days with a ramp-up as discussed above. Results are not shown for the superbee limiter since we were not able to get converged solutions with the implicit version of this scheme unless the time step was chosen close to the stability limit for its explicit counterpart.

First of all, we observe that using a second-order reconstruction gives better resolution for the implicit scheme than for the first-order explicit scheme. For the implicit scheme it also worth noting that the numerical smearing for a linear wave is proportional to the CFL number at which this wave propagates. That is, whereas slow waves experience higher numerical smearing than fast waves in explicit schemes, they experience *lower* numerical smearing than fast waves for implicit discretizations. This means that the implicit schemes essentially will smear the polymer front more inside than outside the channel. The case is the opposite for the explicit schemes. The water front, on the other hand, propagates faster than the polymer front and should thus be most severely affected by numerical smearing for high CFL numbers, but this front has (more) self-sharpening that tends to counteract the smearing.

Conclusions

In this work, we studied explicit and fully implicit schemes with second-order spatial accuracy applied to polymer flooding. We found that it is (relatively) simple to implement such schemes using automatic differentiation and that the resulting schemes work well with standard techniques for time-step control. The use of a high-resolution spatial stencil improves the accuracy both for smooth and discontinuous parts of the solution and reduces grid-orientation effects. We also presented a simple illustrative case, as well as several numerical experiments, that all demonstrate that implicit time discretizations are more suitable than explicit time integration. We expect that this will be even more evident for 3D cases with strongly heterogeneous geology. To make high-resolution methods amenable for implicit discretizations, preference should be given to spatial stencils and nonlinear limiter functions that are as smooth as possible to avoid exacerbating the nonlinearity of the implicit flow equations.

Acknowledgments

The work of Mykkeltvedt was funded by the National IOR Center of Norway, whereas Raynaud and Lie were funded by the Research Council of Norway through grant no. 244361.

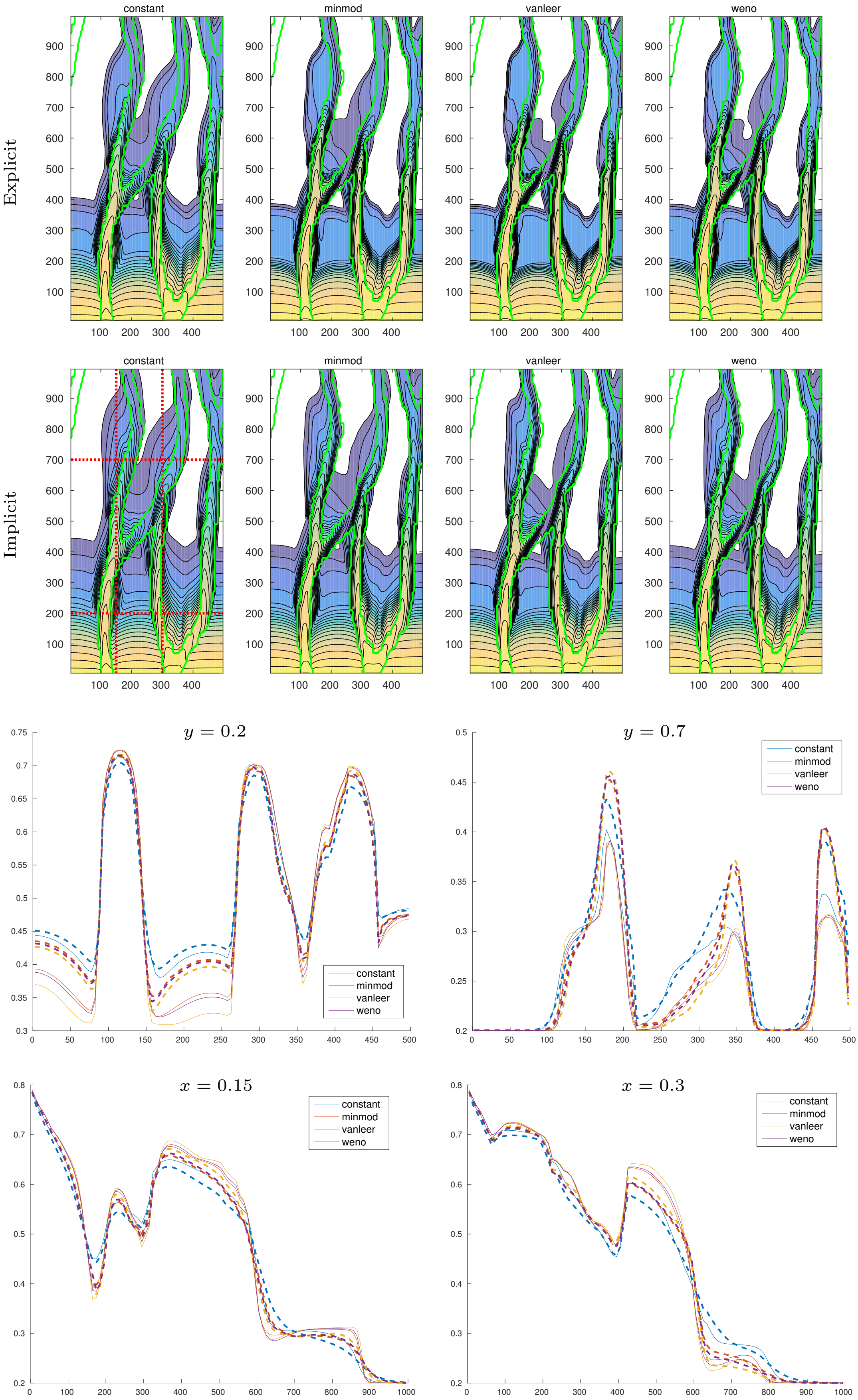

Figure 14 *Polymer flooding of a channelized reservoir simulated. The green lines show the channel borders, whereas the red dotted lines indicate the cross-sections shown in the lower half of the figure.*

References

Bell, J.B., Colella, P. and Trangenstein, J.A. [1989] Higher order Godunov methods for general systems of hyperbolic conservation laws. *J. Comput. Phys.*, **82**(2), 362–397.

Blunt, M. and Rubin, B. [1992] Implicit flux limiting schemes for petroleum reservoir simulation. *J. Comput. Phys.*, **102**(1), 194–210.

Chen, W.H., Durlofsky, L.J., Engquist, B. and Osher, S. [1993] Minimization of grid orientation effects through use of higher order finite difference methods. *SPE Advanced Technology Series*, **1**(2), 43–52.

Chertock, A., Kurganov, A. and Petrova, G. [2009] Fast explicit operator splitting method for convection–diffusion equations. *Int. J. Num. Meth. Fluids*, **59**(3), 309–332.

Gao, C.H. [2011] Scientific research and field applications of polymer flooding in heavy oil recovery. *J. Petrol. Explor. Prod. Technol.*, **1**, 65–70.

Gottlieb, S., Mullen, J.S. and Ruuth, S.J. [2006] A fifth order flux implicit WENO method. *J. Sci. Comp.*, **27**(1-3), 271–287.

Harten, A., Engquist, B., Osher, S.J. and Chakravarthy, S.R. [1987] Uniformly high order accurate essentially non-oscillatory schemes. III. In: Hussaini, M.Y., van Leer, B. and Van Rosendale, J. (Eds.) *Upwind and High-Resolution Schemes*, 71, Springer Berlin Heidelberg, Berlin, Heidelberg, 231–303.

Holing, K., Alvestad, J. and Trangenstein, J.A. [1990] The use of second-order Godunov-type methods for simulating EOR processes in realistic reservoir models. In: *2nd European Conference on the Mathematics of Oil Recovery*.

Kaibara, M.K. and Gomes, S.M. [2001] *Godunov Methods: Theory and Applications*, chap. A fully adaptive multiresolution scheme for shock computations. Springer US, Boston, MA, 497–503.

Khan, S.A., Trangenstein, J.A., Horning, R.D., Holing, K. and Schilling, B.E.R. [1995] Application of adaptive mesh-refinement with a new higher-order method in simulation of a North Sea micellar/polymer flood. In: *SPE Reservoir Simulation Symposium, 12–15 February, San Antonio, Texas*. Society of Petroleum Engineers.

Knudsen, T. [2014] *Full implicit WENO scheme for two phase flow in reservoir simulation*. Master's thesis, Norwegian University of Science and Technology.

Krogstad, S., Lie, K.A., Møyner, O., Nilsen, H.M., Raynaud, X. and Skaflestad, B. [2015] MRST-AD – an open-source framework for rapid prototyping and evaluation of reservoir simulation problems. In: *SPE Reservoir Simulation Symposium, 23–25 February, Houston, Texas*. 1–25.

Lake, L.W. [1989] *Enhanced Oil Recovery*. Prentice-Hall.

Lie, K.A. [2015] *An Introduction to Reservoir Simulation Using MATLAB: User guide for the Matlab Reservoir Simulation Toolbox (MRST)*. SINTEF ICT, www.sintef.no/MRST.

Liu, J., Pope, G.A. and Sepehrnoori, K. [1995] A high-resolution, fully implicit method for enhanced oil recovery simulation. In: *Proceedings of SPE Reservoir Simulation Symposium, San Antonio, Texas, USA, 12 - 15 February*. 35–50.

Liu, X.D., Osher, S.J. and Chan, T. [1994] Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, **115**(1), 200–212.

Mariethoz, G. and Caers, J. [2016] Multiple-point geostatistics stochastic modeling with training images. `http://www.trainingimages.org/`. Accessed 19 April 2016.

MRST [2015b] The MATLAB Reservoir Simulation Toolbox. www.sintef.no/MRST.

Roe, P.L. [1986] Characteristic-based schemes for the Euler equations. *Ann. Rev. Fluid, Mech.*, **18**, 337–365.

Sorbie, K.S. [1991] *Polymer-Improved Oil Recovery*. Springer Science & Business Media.

Sweby, P.K. [1984] High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Numer. Anal.*, **21**(5), 995–1011.

Todd, M.R. and Longstaff, W.J. [1972] The development, testing and application of a numerical simulator for predicting miscible flood performance. *J. Petrol. Tech.*, **24**(7), 874–882.

Trangenstein, J.A. [2009] *Numerical solution of hyperbolic partial differential equations*. Cambridge University Press, Cambridge.

van Leer, B. [1974] Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme. *J. Comput. Phys.*, **14**(4), 361–370.

van Leer, B. [1977] Towards the ultimate conservative difference scheme III. Upstream-centered finite-difference schemes for ideal compressible flow. *J. Comput. Phys.*, **23**(3), 263–275.