Computational Geoscience manuscript No.
(will be inserted by the editor)

Fully Implicit Higher-Order Schemes Applied to Polymer Flooding

Trine S. Mykkeltvedt · Xavier Raynaud · Knut-Andreas Lie

Received: date / Accepted: date

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. We also demonstrate that using lagged evaluation of slope limiters and WENO weights alleviates the nonlinearity of the discrete systems and improves the computational efficiency, without having an adverse effect on the stability and accuracy of the higher-resolution schemes.

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.

Keywords fully implicit · higher-order schemes · polymer flooding

1 Introduction

In many oil recovery processes, water is injected to maintain reservoir pressure and to force the oil towards production wells. If water is less viscous than oil, 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 bypassed oil [12].

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 [25]. 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

T.S. Mykkeltvedt
IRIS, Thormøhlensgt. 55, N-5006 Bergen, Norway
E-mail: trine.s.mykkeltvedt@iris.com

Xavier Raynaud
SINTEF Digital, Mathematics and Cybernetics, P.O. Box 124
Blindern, N-0314 Oslo, Norway
E-mail: xavier.raynaud@sintef.no

Knut-Andreas Lie
SINTEF Digital, Mathematics and Cybernetics, P.O. Box 124
Blindern, N-0314 Oslo, Norway
E-mail: knut-andreas.lie@sintef.no

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 volumetric sweep [25,35]. 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. Many authors have discussed the use of high-resolution spatial discretization schemes to overcome this and similar challenge in subsurface flow simulation, see, e.g., [2,18,22,10,19,21,30,20,8,6,38] and references therein. 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.

In industry, the predominant approach to reservoir simulation with black-oil type models is to use a fully implicit discretization. On one side, this accounts for the coupling between fluid pressure and transport of phases and components by solving for all unknowns simultaneously. 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., [4,3,28,33,11], higher-order simulators reported in the literature are based on explicit time stepping. Implicit total-variation diminishing (TVD) schemes date back to the seminal work of Harten [15] and Yee et al. [43], who showed that five-point TVD schemes with backward Euler temporal discretizations are conservative and unconditionally stable for a scalar 1D conservation law as long as the discrete nonlinear equations are solved exactly using e.g., a Newton method. Later, Gottlieb et al. [14] showed that implicit time-integration schemes of order higher than one are only conditionally TVD. Whether an implicit approach is efficient will thus depend on whether the cost of solving the nonlinear equations is offset by the ability to take larger time steps. In many applications, this is not the case and work on implicit high-resolution schemes has focused on linearization methods or methods that only treat fast waves implicitly for hyperbolic systems with large differences in characteristic wave speeds.

A more prosaic reason why fully implicit formulations are not widely used together with high-resolution discretizations such as 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 (AD) as implemented in the open-source MRST software [24,27,1,31] using operator overloading in MATLAB [32]. 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. AD techniques have mainly been pioneered for subsurface flow simulation through the AD-GPRS simulator [41,44,42], but was also used in an early version of a commercial simulator [7]. The interested reader should also consult [26] for a discussion of an alternative backward-mode approach to AD.

In this work, we first review several high-resolution spatial discretizations and discuss how these can be incorporated as part of a fully implicit simulator. The novelty of our work lies in part in our choice of high-resolution spatial reconstructions, which are different from previous work (e.g., [28]), and in part with the use of automatic differentiation and lagged evaluation of slope limiters and WENO weights to simplify the linearization and solution of the discrete flow equations. Through a series of idealized test cases, we compare and contrast the resulting high-order schemes with standard first-order schemes. 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., [5]. These errors are particularly strong for unstable gas floods where the physical diffusion terms are not fully resolved, and can actually increase when switching from a first-order scheme to a higher-order scheme for the advective terms. For polymer flooding, however, our numerical experiments in Section 4.4, and several other that are not reported herein, indicate that using a high-resolution spatial discretization significantly diminishes grid-orientation errors.

2 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, mixing of polymer in water, and compressibility of fluids and rock. Polymer adsorption and entrapment causes a reduction in permeability to water but not to oil. Moreover, the long-chained polymer molecules are generally not able to access all the pore volume accessible to the much smaller water and hydrocarbon molecules and injected polymer will therefore flow faster than the injected water. The effect of inaccessible pore space is left out for brevity, but could have easily been included; see [17] and references therein for more details on modeling. Last but not least, pseudoplastic effects of the diluted polymer solution may cause shear-thinning or shear-thickening behavior in the near-well region and strongly impact polymer injectivity. In a recent paper [1], we demonstrate that this type of non-Newton fluid rheology can be efficiently resolved by introducing a new inner-Newton loop within the global nonlinear iteration. The same approach can be applied almost verbatim with the higher-order schemes discussed herein, and for brevity we therefore disregard the pseudoplastic effects.

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

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

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

Here, ρ_α, s_α, and $\mathbf{u}_\alpha$ denote density, saturation, and velocity of phase α, whereas c is the polymer concentration given in units of mass per volume of water (and herein take values in the interval $[0,3]$). 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 $\rho_\alpha(p) = b_\alpha(p)\rho_\alpha^S$, 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 to gradients of pressure. For oil, this reads

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

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–Longstaff [37] 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. The two aqueous components will move with different velocities. The water component is assumed to move with a speed given by the phase velocity $\mathbf{u}_w$ of water, whereas the polymer component will move with a speed relative to the water component that is inversely proportional to the ratio between the effective viscosity of polymer and water. This gives us Darcy equations

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

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

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

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

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

3 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*,

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

Here, q denotes one of the primary variables p, s, or c and $\Omega_i = [(i_1 - \frac{1}{2})\Delta x, (i_1 + \frac{1}{2})\Delta x] \times [(i_2 - \frac{1}{2})\Delta y, (i_2 + \frac{1}{2})\Delta y]$ 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

$$\begin{aligned} \mathcal{R}_w &= \big[\rho_w(p_i)\phi(p_i)s_i\big]^{n+1} - \big[\rho_w(p_i)\phi(p_i)s_i\big]^n \\ &+ \frac{\Delta t}{|\Omega_i|} \sum_{|i-j|=1} \int_{\Gamma_{ij}} \big[\rho_w(p)\,\lambda_w\big]^m_{ij} \left(\mathbf{v}^m_w \cdot \mathbf{n}\right)_{i,j} \mathrm{ds} \;= 0. \end{aligned} \tag{7}$$

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 $m = n$ (i.e., forward Euler) for an explicit scheme and $m = n+1$ (i.e., backward Euler) 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 subscript w and superscript m.

3.1 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

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

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

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

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 or second-order scheme, it is sufficient to use the midpoint rule for the integral. The first-order scheme assumes 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. For the second-order scheme, the point values $s^\pm_{i,j}$ are computed by evaluating the two local polynomial reconstructions at the midpoint of the cell face. When using spatial reconstructions of third order or higher, the midpoint rule does not offer sufficient accuracy and the following fourth-order Gaussian quadrature rule should be used instead to evaluate the edge integral in (9),

$$\int\limits_{-1/2}^{1/2} \lambda(x)\mathrm{d}x = \frac{1}{2}\left[\lambda\left(\frac{-1}{2\sqrt{3}}\right) + \lambda\left(\frac{1}{2\sqrt{3}}\right)\right]. \tag{10}$$

This results in similar one-sided values $s^\pm_{i,j,\pm\alpha}$, where subscript $\pm\alpha$ denotes the two different integration points. Given one-sided values, we use the standard upstream method to evaluate the integrand at each integration point

$$\lambda_{i,j}(s,c) = \begin{cases} \lambda(s^-_{i,j}, c^-_{i,j}), & \text{if } v_{i,j} \geq 0, \\ \lambda(s^+_{i,j}, c^+_{i,j}), & \text{otherwise.} \end{cases} \tag{11}$$

Notice that the maximum size of the resulting stencil is determined by the choice of reconstruction and not by the quadrature rule as long as the rule is open and only involves internal points on the cell interface. The quadrature rule affects the flux integration and hence the coefficients in the stencil.

3.2 Second-order slope-limiter schemes

The next question is 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

$$\tilde{q}(\mathbf{x}) = q_i + \sigma_i^x(x - x_i) + \sigma_i^y(y - y_i), \tag{12}$$

where the slopes σ_i^x and σ_i^y 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 [39,40]. Let $\varPhi$ 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 $\varPhi$, the slope σ_i^x is set to

$$\varDelta x \sigma_i^x = \varPhi\left(q_i - q_{i-(1,0)}, q_{i+(1,0)} - q_i\right). \tag{13}$$

Herein, we will consider three different limiters,

$$\varPhi^{mm}(a,b) = \frac{1}{2}\big(\mathrm{sgn}(a) + \mathrm{sgn}(b)\big)\ \min\big(|a|,|b|\big), \tag{14}$$

$$\varPhi^{sb}(a,b) = \frac{1}{2}\big(\mathrm{sgn}(a) + \mathrm{sgn}(b)\big) \tag{15}$$
$$\max\big(\min(\theta|a|,|b|),\ \min(|a|,\theta|b|)\big),$$

$$\varPhi^{vL}(a,b) = \big(b + a\,\frac{|b|}{|a|}\big)/\big(1 + \frac{|b|}{|a|}\big). \tag{16}$$

The *minmod limiter* $\varPhi^{mm}$ 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 [36]. The *superbee* limiter $\varPhi^{sb}$ [34] 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. The parameter θ determines the steepness of the slope chosen; herein, we use $\theta = 1.5$. When used as part of an *implicit* discretization, both limiters have the disadvantage that they are not smooth functions of their arguments: $\varPhi^{mm}$ has one kink, whereas $\varPhi^{sb}$ 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 $\varPhi^{vL}$, which reconstructs steeper slopes than the minmod limiter but gentler slopes than the superbee limiter; see [3]. At the boundary, the reconstruction is computed using a ghost-cell approach in which cell values from the interior are mirrored across each no-flow boundary.

3.3 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 $q_{i\pm1}$ and a quadratic polynomial $q^c(x)$ based on all three cell averages. The classical ENO idea [16] 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 [23]. In the so-called weighted ENO (WENO) schemes [29], the key idea is to instead use a convex combinations of the three polynomials $w^-q^-(x){+}w^cq^c(x){+}w^+q^+(x)$, 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

$$q^{NE}(\mathbf{x}) = q_i + \sigma_i^E(x - x_i) + \sigma_i^N(y - y_i), \qquad (17)$$
$$\sigma_i^E = \frac{q_{i+(1,0)} - q_i}{\Delta x}, \quad \sigma_i^N = \frac{q_{i+(0,1)} - q_i}{\Delta y},$$

and a corresponding smoothness indicator,

$$\beta_i^{NE} = \tfrac{1}{4}\big[\big(\Delta x\, \sigma_i^E\big)^2 + \big(\Delta y\, \sigma_i^N\big)^2 + \varepsilon\big]^{-\ell}, \qquad (18)$$

where ε is a suitable small parameter introduced to avoid division by zero. 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 one-sided point values at the integration points,

$$q_i(\mathbf{x}) = \sum_{\nu = NE,NW,SE,SW} w^\nu q^\nu(\mathbf{x}). \qquad (19)$$

In the default version of the implicit WENO scheme, the nonlinear weights are functions of the unknown solution at time $n + 1$. This contributes to enhance the nonlinearity of the resulting scheme. An alternative, and possibly more efficient, approach is to define the nonlinear weights based on the previous iterate or the values from the previous time step; see e.g., [13]. Whereas this should reduce the computational cost of each time step, we are generally not guaranteed to maintain accuracy and stability, since we now solve a slightly different discrete problem. In Section 4, we will compare accuracy, efficiency, and stability of fully implicit and lagged evaluation of the slope limiters in the TVD scheme and the weights in the WENO scheme.

3.4 Motivation for the temporal discretization

The use of first-order forward or backward Euler in combination with a higher-order spatial discretization may seem a bit strange for readers familiar with high-resolution schemes. Our motivation for this is as follows: by the results of Gottlieb et al. [14] it follows that higher-order temporal discretizations are only conditionally stable. This means that using, for instance, the standard θ rule of the form $y^n + 1 - \theta f(y^{n+1}) = y^n + (1-\theta) f(y^n)$ to get second-order implicit temporal discretization will inevitably introduce a time-step restriction stemming from the explicit part of the stencil. For reservoir models with large differences in porosities and/or large differences in the face areas between neighboring cells, this time-step restriction can be quite severe. For efficiency, the parameter θ should be chosen locally and many cells will need to be treated fully implicitly to avoid too small overall time steps that would otherwise adversely affect computational efficiency. There exist alternative approaches, see e.g., [9], to ensure non-oscillatory solutions for significantly larger time steps than what can be achieved by linear time-integration schemes. However, these introduce additional nonlinearities in the temporal discretization and for simplicity, we have chosen to only consider the case with $\theta = 1$ for the implicit schemes. For the explicit schemes, we could have used a second-order TVD Runge–Kutta method [14], which computes the value at time $t + \Delta t$ as the average of two forward-Euler steps, but we chose to also use first-order temporal discretization for a more clean-cut comparison.

3.5 Solving the discretized systems

If we now pick one of the reconstruction methods introduced above to evaluate one-sided point values, plug (9) and (11) into (7), and repeat the same derivation with the obvious modifications for the oil and the polymer equations, we end up with a system of nonlinear equations that can be written in the residual form as

$$\mathbf{F}(\mathbf{y}) = \mathbf{0}, \qquad (20)$$

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

$$\mathbf{0} \approx \mathbf{F}(\mathbf{y}_0) + \mathbf{J}\delta\mathbf{y}, \qquad (21)$$

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. The Newton iteration process will exhibit quadratic convergence under certain requirements on the smoothness and differentiability of $\mathbf{F}$. Unfortunately, $\mathbf{F}$ is generally not smooth for our system because of the switch in the upstream formula (11). Moreover, the explicit schemes are 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.

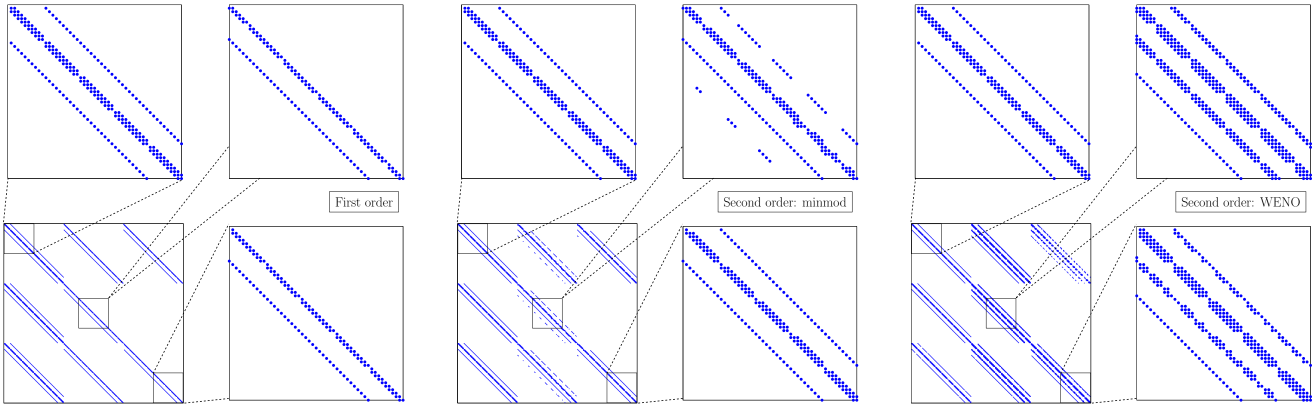

Fig. 1 Structure of the Jacobian matrix for various implicit schemes on a 10×10 quarter-five spot example. The four plots included for each scheme show, starting in the south-west corner and moving clockwise: the complete 3×3 block matrix and parts of the $\partial_p \mathcal{R}_w$, $\partial_s \mathcal{R}_o$, and $\partial_c \mathcal{R}_{wp}$ blocks, respectively.

Another practical challenge with (21) 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 [24,27]. 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 is simple to differentiate analytically and can easily be coded into a software using operator overloading so that evaluation of the overloaded function 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 computation of $\mathbf{J}$ in more detail for any of the schemes.

Let us look at the structure of the Jacobian in some detail. With three residual equations of the form (7), which we denote $\mathcal{R}_w$, $\mathcal{R}_o$ and $\mathcal{R}_p$, and three primary variables p, s, and c, the linearized system reads,

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathcal{R}_w}{\partial p} & \frac{\partial \mathcal{R}_w}{\partial s} & \frac{\partial \mathcal{R}_w}{\partial c} \\ \frac{\partial \mathcal{R}_o}{\partial p} & \frac{\partial \mathcal{R}_o}{\partial s} & \frac{\partial \mathcal{R}_o}{\partial c} \\ \frac{\partial \mathcal{R}_{wp}}{\partial p} & \frac{\partial \mathcal{R}_{wp}}{\partial s} & \frac{\partial \mathcal{R}_{wp}}{\partial c} \end{bmatrix}. \tag{22}$$

To utilize efficient vectorization in MATLAB, the uknowns are stored by concatenating global vectors for each primary variable, i.e., $\mathbf{y} = [\mathbf{p}, \mathbf{s}, \mathbf{c}]$. Thus, $\mathbf{J}$ becomes a *global* 3×3 block matrix that has a different sparsity pattern than in most other simulators that store all primary variables consecutively per cell.

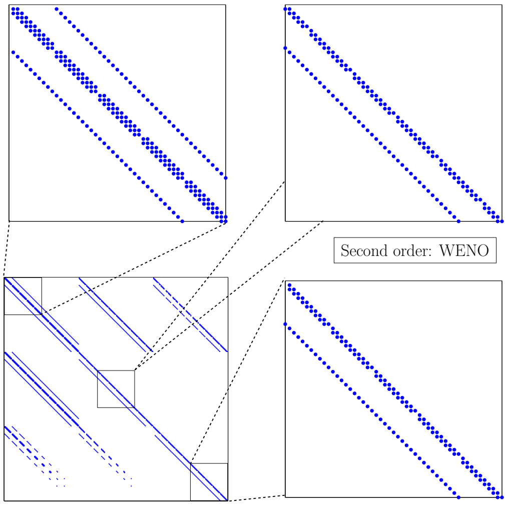

Fig. 2 Structure of the Jacobian matrix for the WENO scheme with lagged evaluation of weights for a quarter-five spot example on a 10×10 grid.

To show the structure of the different schemes, we consider a simple quarter-five spot setup on a 10×10 grid, which will be discussed in more detail in next section. Figure 1 shows the Jacobian matrix for the first-order scheme and the second-order schemes with minmod and van Leer limiters. Starting with the first-order scheme, we see that the matrix blocks representing derivatives of conservation equations with respect to pressure all have a pentadiagonal structure. The small 'gaps' in the diagonals correspond to the boundary of the domain. Looking next at the derivative of the oil equation with respect to saturation, we see that this has a tridiagonal structure since the flow is cocurrent and the conservation equations have unidirectional eigenvalues with respect to saturation. The structure is similar for all the other blocks representing derivative with respect to s and c.

Our numerical experiments have shown that it is advantageous to use a constant reconstruction for pressure also in the second order methods, and hence all blocks representing derivatives with respect to pressure have the same pentadiagonal structure as the first-order scheme. For the minmod limiter (14), the other blocks generally have a 7-diagonal structure as a result of cocurrent flow and monotone displacement profiles. Since the limiter chooses the slope in each axial direction as the minimal value of the backward and forward divided differences, or sets the slope to zero if these have different sign, at most five of the seven diagonal elements are nonzero in each row. The blocks for the superbee limiter (15) have a similar structure since this limiter chooses the maximum of the two divided differences. The van Leer limiter (16) computes the slope as a nonlinear average of the backward and forward divided differences and will generally have nonzero values for all seven diagonals. Whereas the TVD limiters use dimension-by-dimension slopes, the WENO scheme relies on a multidimensional approach and hence has denser matrix blocks consisting of ten nonzero diagonals in this particular case. The resulting Jacobian is generally smoother than for the TVD schemes, which to a certain extent compensates for the somewhat denser matrix blocks. When using lagged evaluation of slope limiters and WENO weights, the matrix blocks representing derivatives with respect to s and c reduce to the same tridiagonal structure as for the first-order scheme. This is shown for the WENO scheme in Figure 2. Interestingly, we observe that this linearization causes several of the derivatives of the polymer equation with respect to pressure and saturation to become zero.

4 Numerical Experiments

This section presents a series of numerical experiments that compare and contrast the various schemes described above. Whereas the schemes are not backed up by rigorous analysis of stability and accuracy when applied to the multidimensional polymer system, the classical analysis of implicit TVD schemes for scalar equations ([15,43]) and the large body of subsequent work on similar explicit schemes, have lead us to expect that these implicit schemes will be stable and free of spurious oscillations.

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 3. 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

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

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

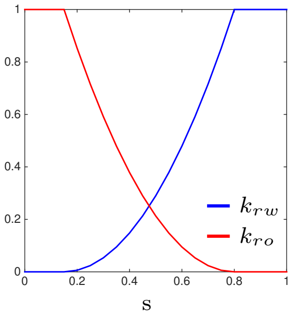

Fig. 3 Fluid parameters used for the numerical experiments. The relative permeabilities $k_{r\alpha}$ are formed by linear interpolation of tabulated data.

4.1 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 Buckley–Leverett form, which somewhat simplified reads

$$\frac{\partial}{\partial t}\begin{bmatrix} s \\ sc + a(c,x) \end{bmatrix} + \frac{u}{\phi}\frac{\partial}{\partial x}\begin{bmatrix} f(s,c) \\ m(c)f(s,c) \end{bmatrix} = 0,$$

$$f(s,c) = \frac{\lambda_w(s,c)}{\lambda_w(s,c) + \lambda_o(s)},$$

where the total flux u is constant and $m(c)$ denotes the ratio $\mu_{w,\text{eff}}/\mu_{p,\text{eff}}$. When a is a function of c only, this

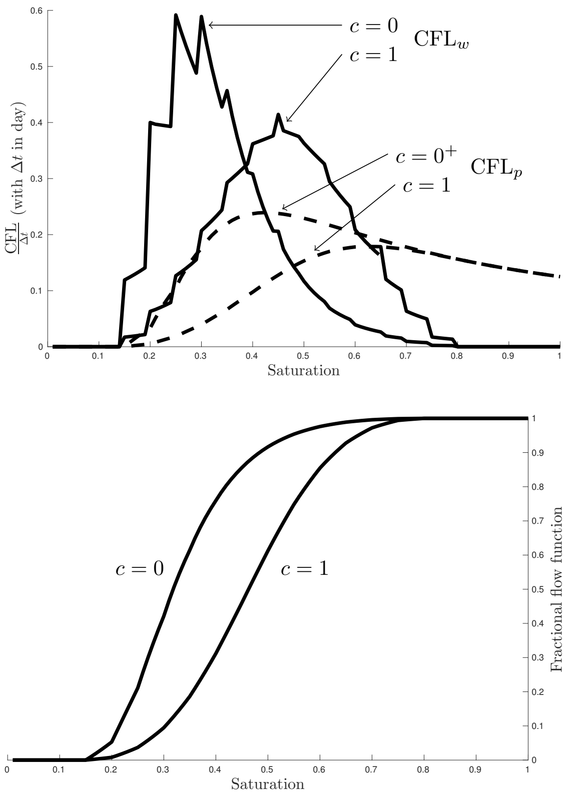

Fig. 4 The upper plot shows the CFL number for the s-waves (solid lines) and c-waves (dashed lines) as functions of saturation for two polymer concentrations, $c = 0^+$ and $c = 1$. The apparent piecewise linearity of the CFL curve comes from the fact that our data parameters are tabulated. We observe that there are several local extrema, meaning that we have a fractional flux function with several inflection points. The lower plot shows the fractional flow curve, where a higher polymer concentration results in a translation of the curve to the right, so that water breakthrough is delayed and recovery improved.

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 4),

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

Figure 5 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 analyze 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 $\mathcal{O}\big(\sqrt{t(\Delta\tau \pm \Delta t)}\big)$. Hence, 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 6, 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

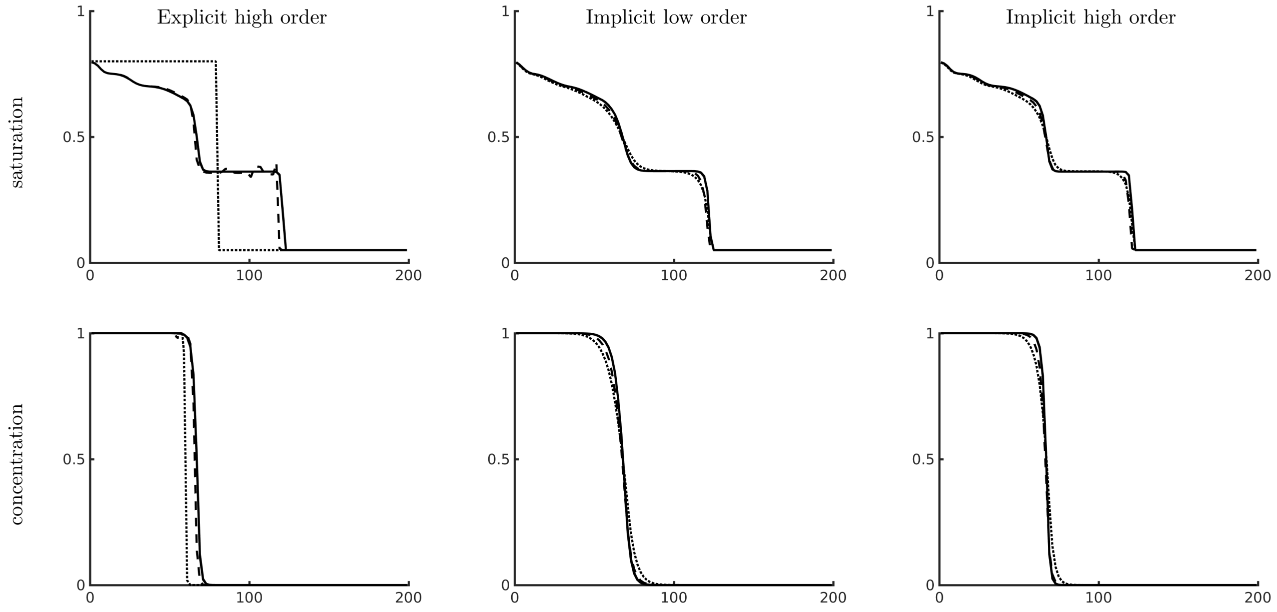

Fig. 5 Plot of saturation and concentration for the first 1D test case. The values are plotted for four different time steps, Δt equal to 0.01 days and 0.1 days (solid line), 1 day (dash-dot), and 3 days (dot). All methods have converged in time for Δt equal 0.1 days and hence solutions with time steps of 0.01 and 0.1 days cannot be distinguished.

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 all three time steps, as expected. Figure 6 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 7 for a case of polymer flooding. 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, the high-order implicit scheme resolves the displacement fronts as good as the explicit scheme. Notice also that the low-order implicit scheme fails to sharply resolve the structure of the oil bank that arises as the polymer front passes from the low-porosity to the high-porosity region. This wave is a transient introduced by the spatially-dependent adsorption term $a(c, x)$ and is not part of an otherwise self-similar structure. Table 1 reports a comparison of the computational costs for the explicit and implicit schemes. When using the same time step as the explicit schemes, the implicit schemes are significantly more costly. However, a time step of 0.1 days is beyond the stability limit of the explicit schemes, and when running these in a stable manner, we not only get higher simulation times, but significantly more smearing of the leading waves. In this example, it is therefore safe to conclude that the second-order implicit scheme is at least as effective as the explicit schemes.

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. The value of $\Delta t = 0.03$ days for the explicit scheme corresponds to the time step where oscillations in the low porosity region start to appear.

Simulation case			Linear	CPU
Δt (day)	method	order	solves	time
0.01	explicit	1st	15307	504.2 s
0.01		2nd	15330	541.0 s
0.03		1st	5297	175.5s
0.03		2nd	5328	184.6s
0.1	implicit	1st	2176	89.7 s
0.1		2nd	2960	223.6 s
1		1st	450	14.8 s
1		2nd	508	33.1 s
2		1st	267	8.5 s
2		2nd	330	20.5 s

4.2 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

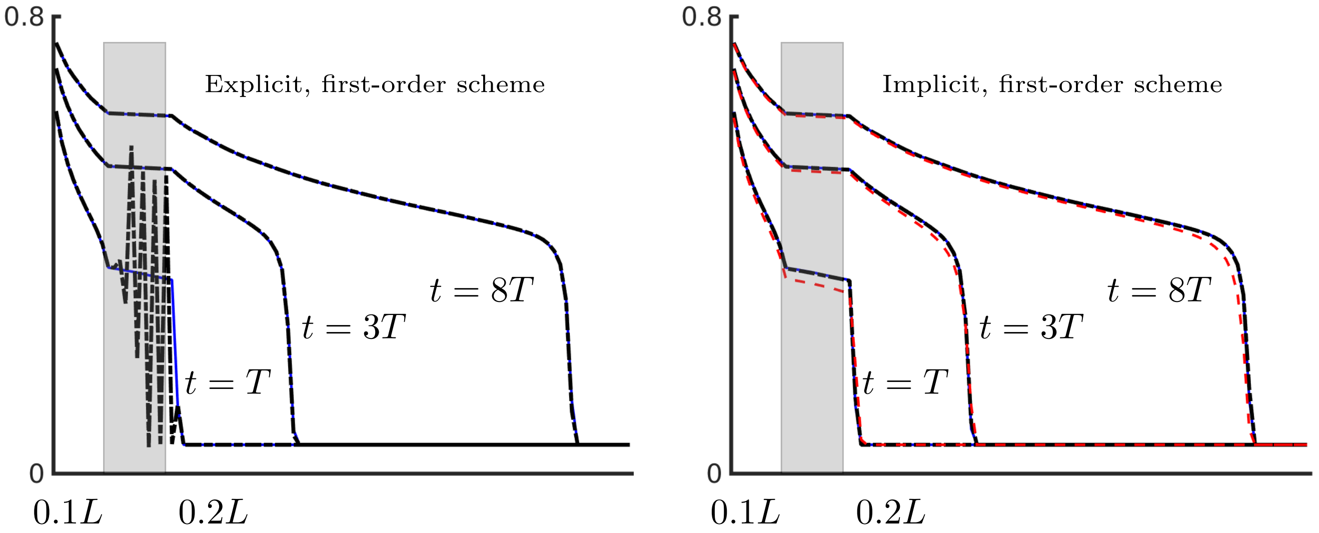

Fig. 6 Comparison plots for the pure waterflooding case with regions of different porosity ($\phi = 0.01$ for $x \in [0.1L, 0.2L]$ and $\phi = 0.2$ otherwise). Three different time steps are used, $\Delta t = 0.01$ day (blue solid lines), $\Delta t = 0.1$ (black dashed-dot lines) and $\Delta t = 1$ (red dashed lines) for the first-order scheme. For the explicit method, only the smallest time step is stable. For $\Delta t = 0.1$ day, the scheme is unstable and gives strong oscillations in the low-porosity regions, but these oscillations disappear in the high-porosity region as a result of wave interactions and numerical diffusion. The scheme did not converge for $\Delta t = 1$ day.

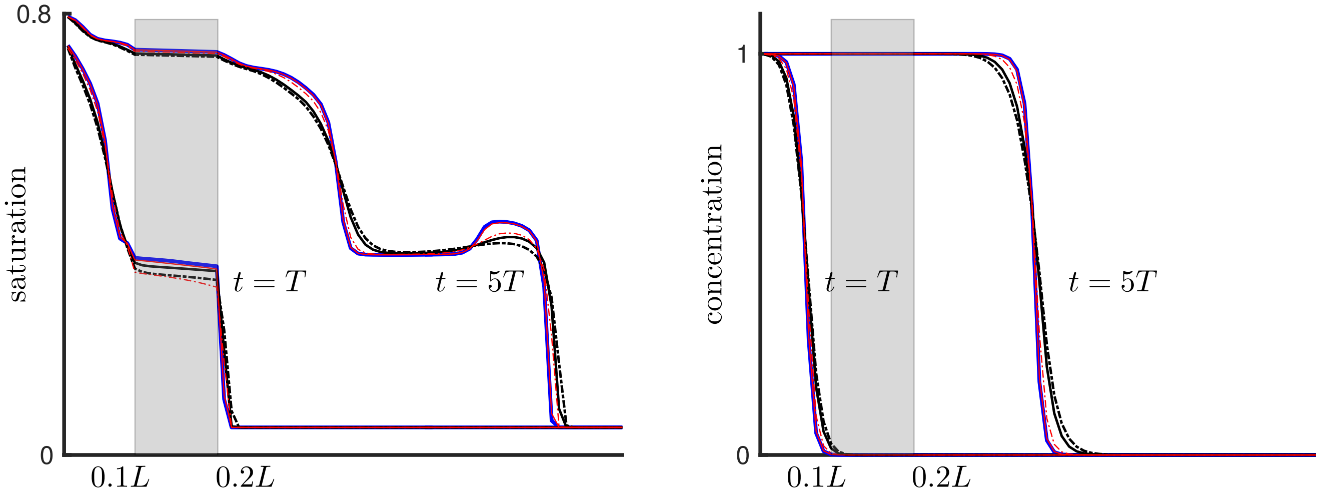

Fig. 7 Comparison between implicit high-order (red) and low-order (black) schemes for time steps equal to $\Delta t = 0.1$ day (solid line) and $\Delta t = 1$ day (dashed line). The blue line shows the solution obtained by the explicit scheme for a refinement of $\Delta t = 0.01$ days.

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 8 shows solutions just after water breakthrough computed with the first-order scheme (constant reconstruction) and with minmod, van Leer, superbee, 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 various 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

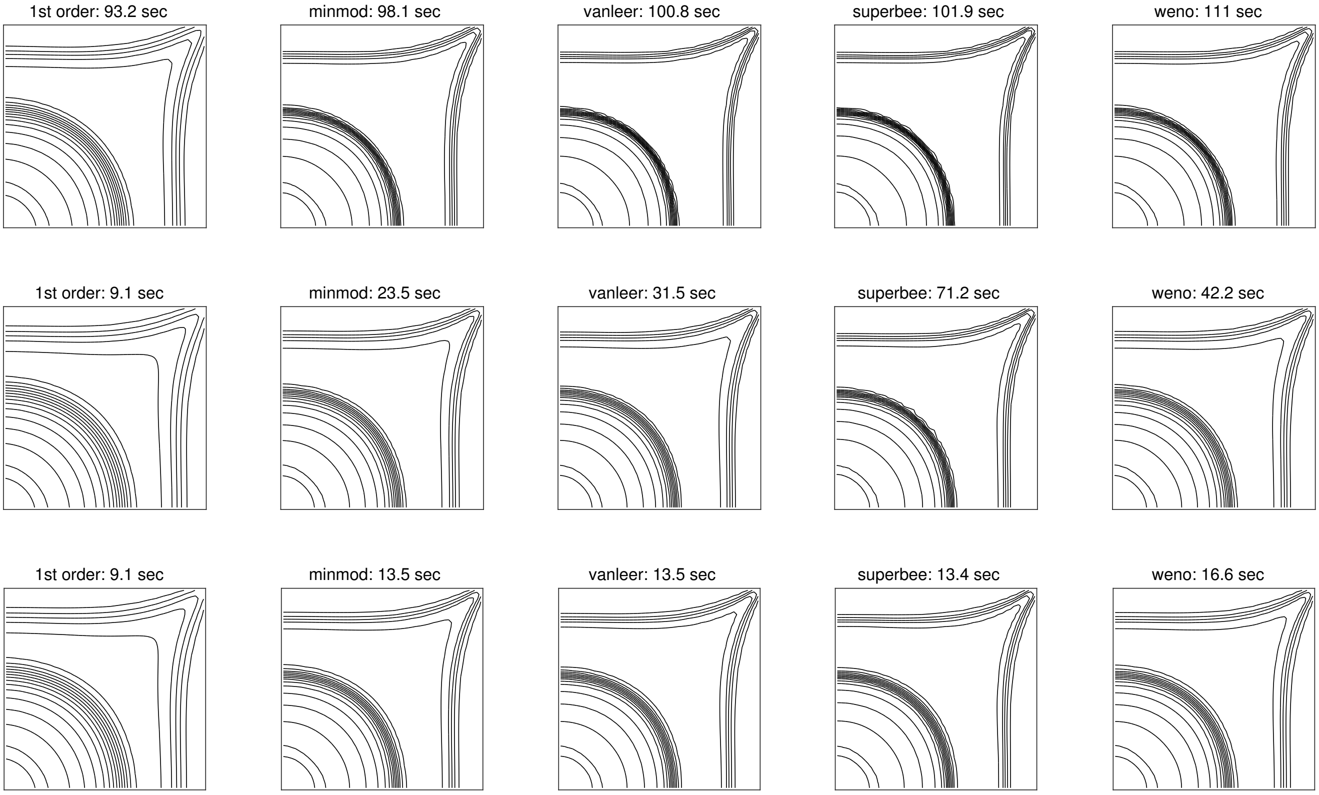

Fig. 8 Comparison of saturation profiles on a quarter five-spot problem computed on a uniform Cartesian grid with 50 × 50 cells. The upper row shows explicit schemes with 1600 time steps, the middle row shows implicit schemes with 50 time steps and fully implicit reconstruction, whereas lagged reconstruction is used in the bottom row. Notice how the leading water front is sharper resolved than the trailing polymer front. Contour lines: s=0.215, 0.245,. . . ,0.785.

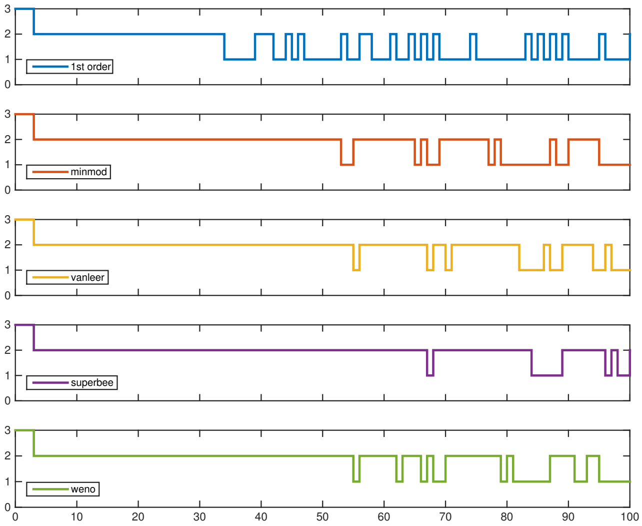

Fig. 9 Number of iterations during the first 100 time steps for the explicit schemes in Figure 8.

the initial pressure transient. From a nonlinear point of view, however, this is equivalent to IMPES. Figure 9 reports the number of iterations required by all five schemes. All five schemes require three iterations in the first three steps, and then two iterations or less thereafter. For the first-order scheme, the Newton solver converges in one iteration more often than for the second-order schemes. 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 can 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 → 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 10 reports the substeps and the corresponding number of iterations for each of the five implicit schemes with fully implicit reconstruction, whereas Figure 11 shows the corresponding cumulative number of iterations with fully implicit and lagged reconstruction. For the first-order scheme, the first step of the ramp-up sequence requires four 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 five ramp-up steps. In the sixth and last ramp-up step the number of iterations increases to four and hence the next step is chopped in two. The WENO and van Leer schemes have similar behavior; both are smoother than minmod (whose limiter has one kink) and can be expected to require fewer iterations. In this particular case, however, tht total number of iterations is only slightly less than for minmod. For the highly compressive superbee limiter, which has three kinks, the Newton solver struggles more and altogether requires four times 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 having a large number of redundant function evaluations. Setting the tolerance lower than 10^{-3} had no effect on the contour lines, but naturally gave more iterations for each of the schemes. The trends were still the same.

Introducing lagged evaluation of slope limiters and WENO weights reduces the number of iterations significantly, particularly for the superbee limiter, but seems to not have any adverse on stability and only reduces the accuracy slightly. With lagged reconstructions, all schemes approximately the same number of iterations as the constant scheme with fully implicit reconstruction. Improvements in runtime are disproportionally larger compared with the reduction in the number of iterations. Lagged evaluation not only reduces the nonlinearity of $\mathbf{J}$ but removes many functional dependencies, which in turn greatly simplifies the algorithmic complexity and the computational cost of the automatic differentiation. In other words, the AD library uses significantly fewer chain-rule evaluations and collects significantly fewer partial Jacobians in the forward accumulation used to construct the overall Jacobian matrix. An overall computational cost slightly less than twice that of the first-order scheme is quite promising.

Last, we test the ability of the implicit schemes to use large CFL number when propagating well-established displacement fronts. This is important if the schemes are to provide large-time-step capabilities for computational efficiency. To this end, we first use an explicit scheme to compute two well-established displacement fronts, corresponding to approximately 1/15 and 1/3 of the time to breakthrough for the simulation shown in Figure 8. Using these solutions as initial data, compute a single time step with each implicit solver corresponding to CFL numbers 2^k for $k = 0, \dots, 9$. Figure 12 reports the observed number of iterations to convergence. Lagged evaluation increases the robustness of second-order schemes significantly, particularly for the compressive superbee limiter. The ability to take large time steps improves as the displacement front propagates into the reservoir. Two factors contribute to this: For this type of setup, the CFL number is given by the high flow velocities in the near-well region. At time 5 days (which is approximately 1/3 of the time to water breakthrough), both discontinuities in the displacement profile have moved to regions of lower fluid velocities and thus propagate at CFL numbers that are effectively lower. Secondly, since the initial condition is obtained by numerical simulation, it will be more smooth when sampled at 5 days instead of at 1 day. Repeating the experiments discussed above using tighter tolerances gives higher number of iterations, but the same qualitative behavior with respect to schemes and time-step size.

4.3 Quarter five-spot: spatial and temporal convergence

Figure 8 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 13 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

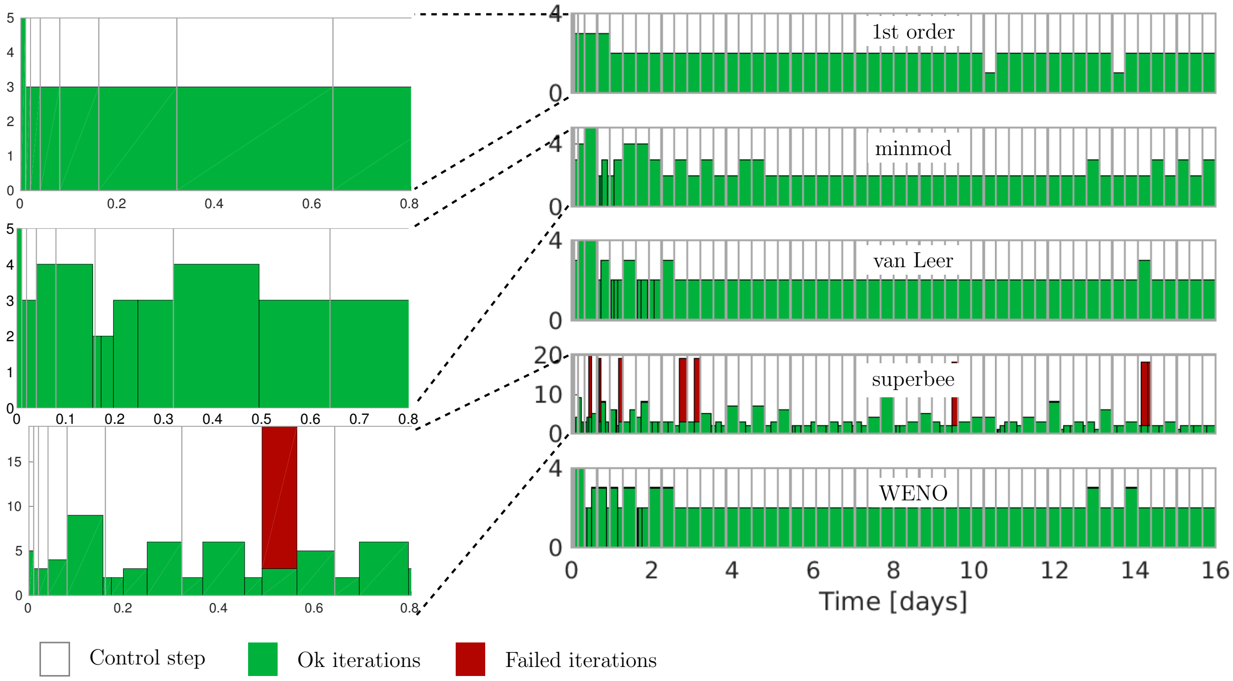

Fig. 10 Number of nonlinear iterations for the five implicit schemes from Figure 8. The bar graph to the right shows the number of iterations per targeted time step (control step) and how these have been divided into substeps to ensure that the number of iterations per step (green bars) do not exceed five. Steps that have not converged within 15 iterations are considered to fail and illustrated by a red bar.

twice as many time steps. Figure 14 confirms that although lagged evaluation of slope limiters will not generally reproduce the fully implicit solution exactly, even when the residuals are reduced to machine precision, the discrepancy of the two solutions seems to converge toward the same solution as the grid is refined.

Figure 15 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.

4.4 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 16. 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 for polymer flooding, we compare solutions to the five-spot problem computed using the quarter five-spot and the rotated five-spot setups shown Figure 16. 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. The mobility ratio in the experiment is approximately equal 21.5.

In the original setup, the preferential flow along the grid axes will tend to overestimate the frontal movement 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 17 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

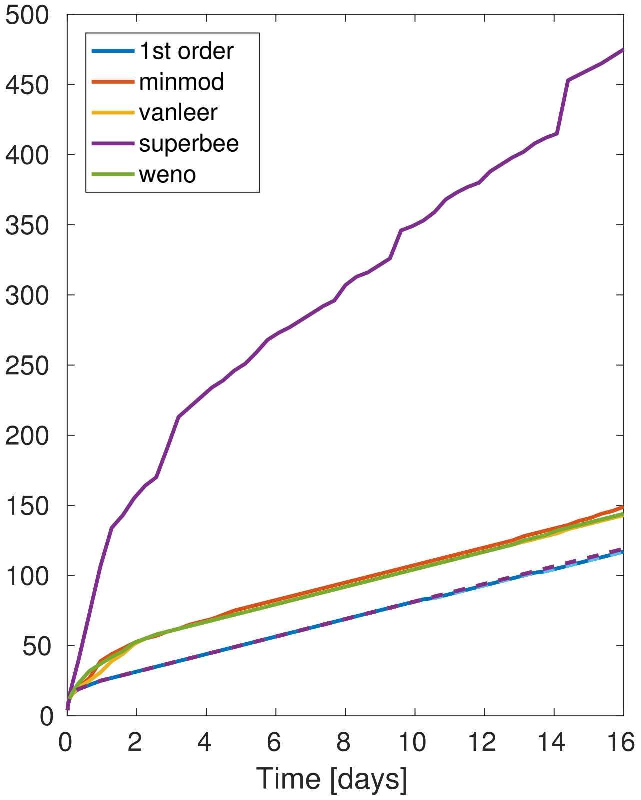

Fig. 11 Cumulative number of nonlinear iterations for the five implicit schemes from Figure 8. Solid lines represent fully implicit reconstruction and dashed lines lagged reconstruction.

with increased spatial (and temporal) resolution. For completeness, this study was repeated using fourth order Gauss quadrature in the flux evaluation, which gave virtually identical contour lines, but somewhat higher runtimes.

The plots discussed above only show a snapshot of the grid-orientation errors at one instant in time. Figure 18 compares water saturation in the cells containing the fluid sink, as computed by three of the different schemes on the original and the rotated geometry. (Grid sizes are 20×20 and 28×28, respectively.) Not only is the discrepancy in production curves significantly larger for the first-order scheme, but the scheme fails completely to distinguish the effect of the two fronts in the displacement profile.

4.5 More on numerical diffusion

Numerical diffusion generally manifests itself as a full tensor. When written in streamline coordinates, the numerical diffusion of a purely advective flow (i.e., viscous flow without buoyancy and capillary terms) can be separated into longitudinal and transverse components acting along and orthogonal to the flow direction, respectively. Since the diffusion tensor is generally not aligned with the flow direction, there will also be additional cross terms, but these will be negelected herein for the sake of the argument. Somewhat simplified, we can then say that the size of the longitudinal diffusion explains how numerical smearing affects the prediction of local displacement efficiency, whereas the transverse diffusion mainly affects the prediction of volumetric sweep efficiency.

In Section 4.1, we used a simple 1D example to illustrate how numerical smearing in the longitudinal direction is affected by heterogeneities and variation in effective CFL numbers. To quickly recap, numerical smearing decreases as the time step increases towards the CFL limit for an explicit scheme. For implicit schemes, the smearing increases with the size of the time step. In polymer flooding, and similar water-based EOR processes, chemical or biological components responsible for tertiary recovery will typically follow slow waves. To compare explicit and implicit schemes, we disregard self-sharpening effects which tend to counteract numerical smearing, and observe that slow waves experience more numerical smearing than fast waves for explicit schemes, whereas slow waves experience *less* numerical smearing than fast waves for implicit schemes. Transverse diffusion depends primarly on grid size and the local orientation of the flow relative to the individual grid cells and does, unlike longitudinal diffusion, not depend on the time step or the specifics of the time-integration scheme. Altogether, this suggests that even though implicit schemes tend to smear water fronts and other fast waves significantly more than similar explicit schemes, they may still predict trailing waves and volumetric sweep just as accurately.

In an attempt to illustrate and substiantiate this claim, we will study a simple and conceptual model of a channelized reservoir. Our setup consists of a rectangular 500×1000 m^2 domain with a homogeneous and isotropic background permeability of 100 md. The reservoir contains two channels of a somewhat higher permeability of 200 md oriented in the north–south direction for $x \in [100, 200]$ and $x \in [300, 400]$. The porosity is 0.25 inside the channels and 0.2 outside. 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 19 shows approximate solutions computed by the implicit and explicit schemes with constant and van Leer reconstructions. For the explicit schemes we use a time step of 2 days to get to the final simulation time of 5 years, and for the implicit

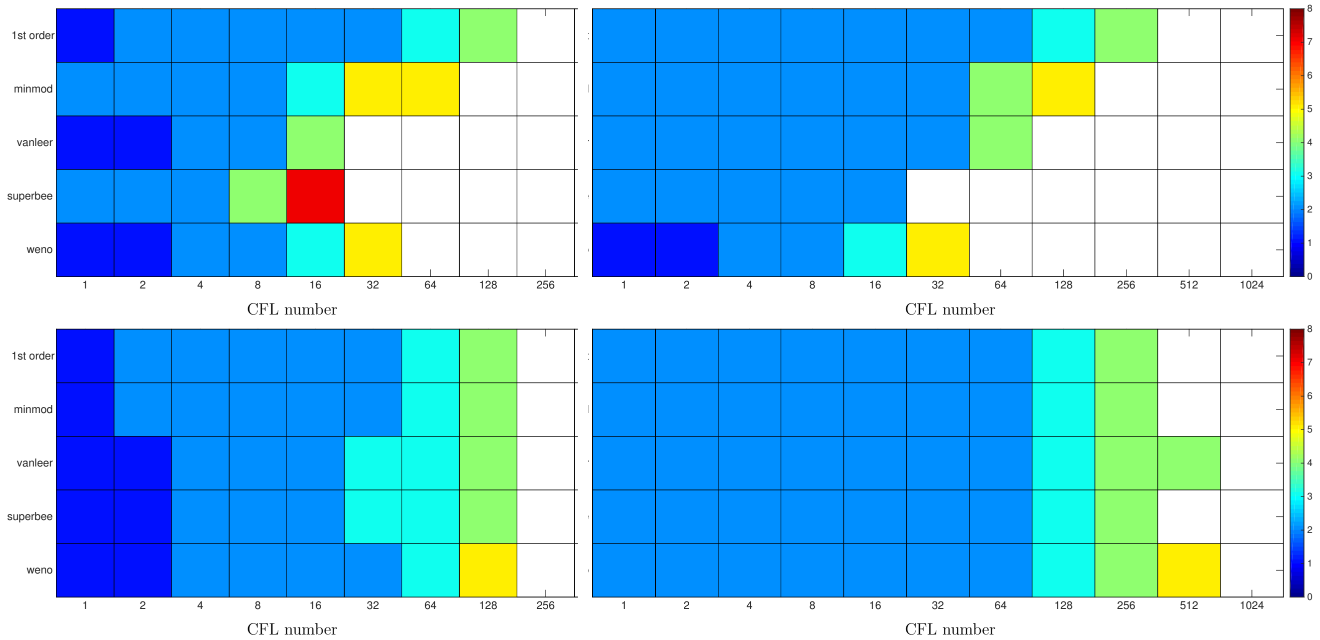

Fig. 12 Test of large-time-step capability on a homogeneous quarter-five spot for the various implicit schemes with fully implicit (top) and lagged (bottom) evaluation of slope limiters and WENO weights. The test consists of simulating a single time step, starting from a well-established displacement profile sampled at time 1 day (left) and at time 5 days (right), which correspond approximately to 1/15 and 1/3 of the time to water breakthrough. The color in each cell reports the number of iterations required to converge a single time step with the given CFL number. White color indicates no convergence.

schemes we used control step of 50 days with a ramp-up as discussed above. Using the runtime of the first-order implicit scheme as a reference, the explicit schemes are 7.1 and 7.3 times slower, whereas the implicit van Leer scheme is 1.2 times slower.

As expected, the explicit van Leer scheme gives by far the best resolution, but is also the most computationally costly. More interestingly, the implicit van Leer scheme appears to be at least as accurate as the first-order explicit scheme. Starting with the north–south cross-section inside the western channel, we observe that whereas the first-order explicit scheme computes a sharper water front, the implicit van Leer scheme gives slightly better resolution of the trailing chemical wave. For the cross-section in the middle of the lower-permeable region in between the channels, the implicit van Leer scheme resolves the trailing chemical wave significantly sharper and provides almost identical resolution for the leading water front. The two cross-sections sampled in the east-west direction seem to confirm that all four schemes have the same amount of transverse diffusion since each transition between flow in the background and a channel is resolved with the same number of grid cells. There are differences in saturation values inside each permeability region, but these are a result of differences in longitudinal diffusion. This is particularly evident for the fourth cross-section, where the second-order schemes capture the constant region between the water front and the chemical wave inside the channels most accurately. Notice also that the explicit van Leer scheme is the only one to capture that the piecewise linear mobility functions induce numerical artifacts in the form of small wiggles in the solution.

4.6 Including buoyancy

In the previous numerical examples, buoyancy was not included in the simulations. To study if including buoyancy makes nonlinear convergence more difficult for the numerical schemes discussed herein, we consider a simple yet challenging example. Let a square domain of length L initially be filled with water and oil. Water saturation and polymer concentration are set to $(s, c) = (0.8, 1)$ for $x \leq L/2$ and $(s, c) = (0.2, 0)$ for $x > L/2$. No-flow boundary conditions are imposed on the outer boundaries, and the initial pressure is equal and constant over the entire domain. Water, polymer, and oil will move only due to buoyancy. Figure 20 shows initial water saturation and polymer concentration and at two later times computed on a 50 × 50 grid using minmod reconstruction in a fully implicit framework.

Figure 21 reports the cumulative number of iterations needed by the nonlinear solver to perform the simulation in Figure 20 for the five different reconstructions for the fully implicit formulation and with

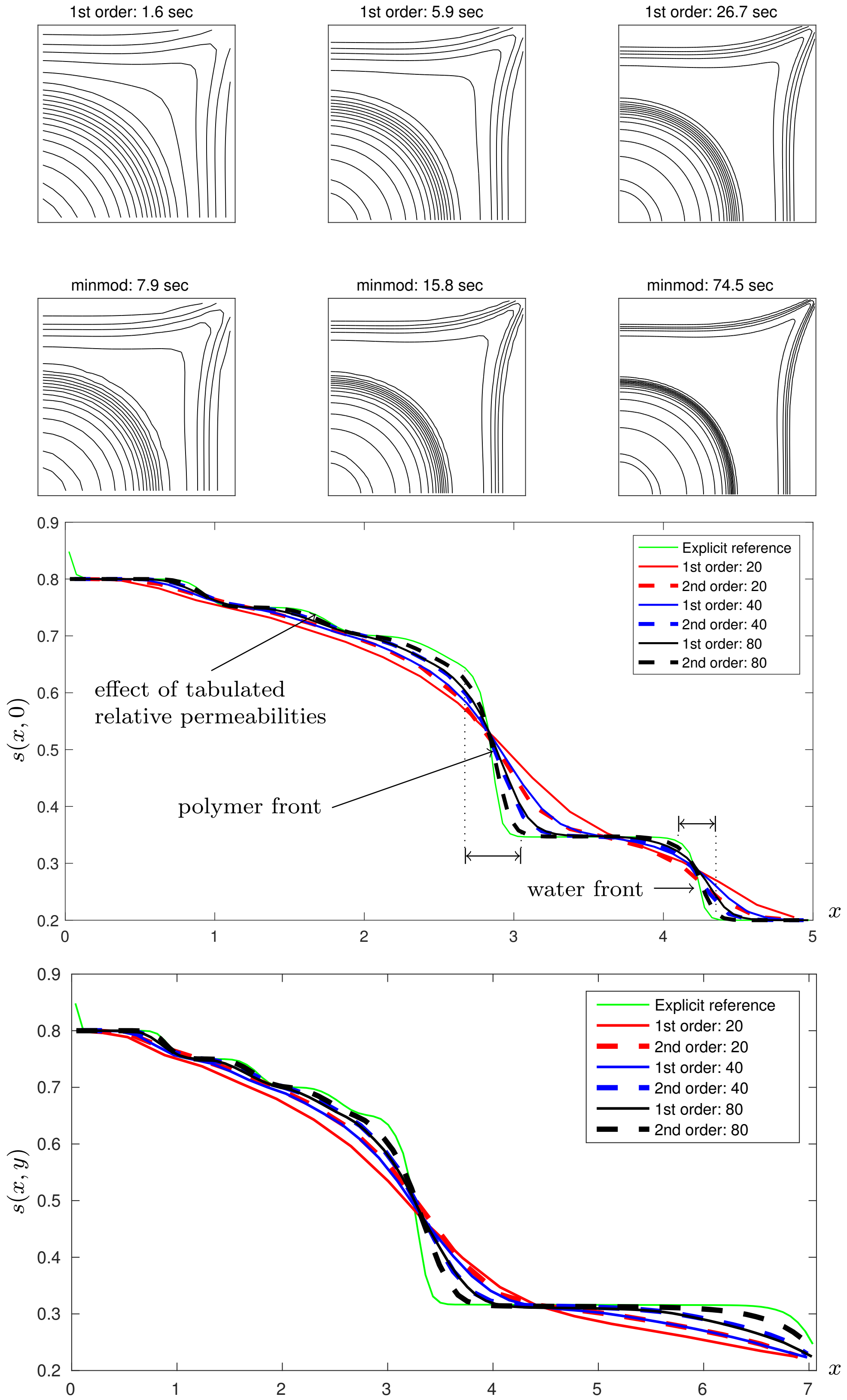

Fig. 13 Convergence study for the implicit first-order scheme (upper row) and the implicit minmod scheme (second row) on a sequence of $n \times n$ grids with m uniform control steps for $n, m = 20, 40, 80$. The two upper rows show the saturation field, whereas the two bottom rows show saturation profiles 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. Contour lines: s=0.215, 0.245,. . . ,0.785.

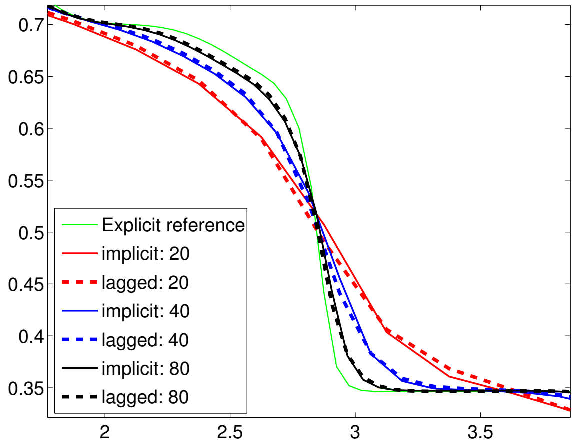

Fig. 14 Convergence study of the fully implicit and the lagged minmod schemes with the same setup as in Figure 13. The plot shows a zoom of water saturation near the polymer front along the diagonal $x = y$.

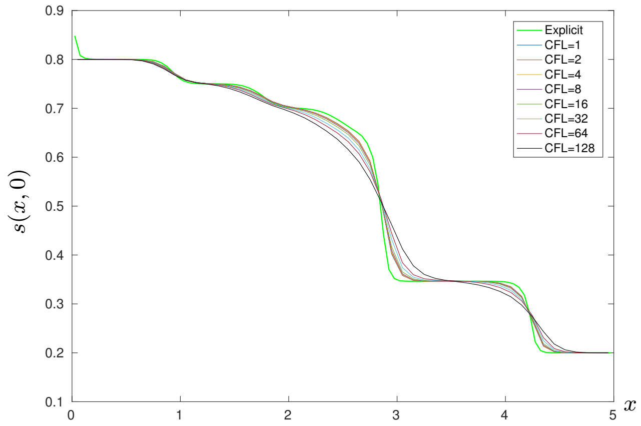

Fig. 15 Convergence study for the implicit minmod scheme on a 50 × 50 grid with decaying time steps. Saturations are sampled along the x-axis and compared with a solution computed by the explicit minmod scheme on a 100 × 100 grid.

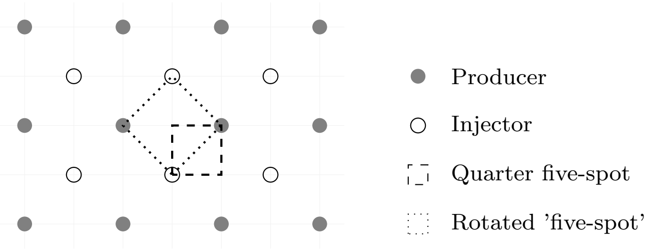

Fig. 16 Well setup for the five-spot test cases used to study grid-orientation errors.

lagged evaluation of slope limiters and WENO weights. All simulations use the same time-step control as discussed in Section 4.2. Consistent with our observations above, the first-order scheme uses less iterations than the high-resolution schemes. Among the latter, minmod with lagged evaluation is the least expensive, requiring approximately 15% more iterations than the first-order scheme. The superbee limiter needs significantly more iterations than the other reconstructions when used in a fully implicit setup, whereas with lagged evaluation, the scheme is as efficient the fully implicit scheme with van Leer limiter. Also important, no ”flip-flopping” of the upstream direction in equation (11), which potentially could destroy nonlinear convergence, is observed for any of the schemes.

5 Conclusions

In this work, we studied explicit and implicit schemes with formal 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 for the relative simple polymer-flooding scenarios studies herein. We also presented a simple illustrative case, as well as several numerical experiments, all demonstrating that implicit time discretizations are more suitable than explicit time integration. In particular, we demonstrated that although implicit discretization introduces signficant numerical smearing that may adversely affect the resolution of fast waves, the resolution of trailing chemical waves is as accurately for a high-order implicit scheme as for a first-order explicit scheme. Equally important, transverse diffusion introduced by the schemes seems to be largely independent of the time step and scheme used for time integration. We have also run a number of expriments using strongly heterogeneous rock properties sampled from the SPE 10 benchmark, which for brevity have not been reported herein but all confirm the discussion above. To what extent these observations carry over to more realistic 3D cases with strongly heterogeneous geology represented on grids with non-Cartesian cell geometries and complex topology is still an open question. Nonetheless, we believe that our results provide strong arguments for why implicit schemes with high-resolution stencils is an interesting technology that should be further researched.

High computation cost is key argument against the use of higher-resolution discretizations as part of a fully implicit formulation. To make these methods amenable for realistic flow scenarios, 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. We also showed that using lagged evaluation of slope limiters or WENO

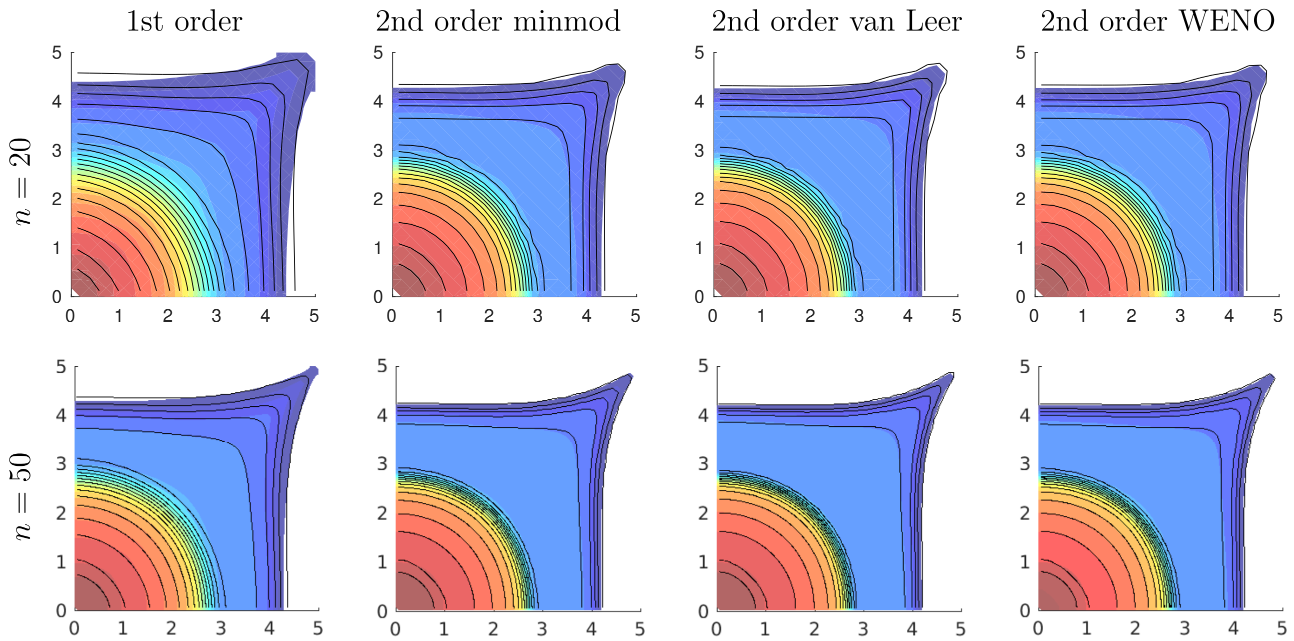

Fig. 17 Test of grid-orientation errors for the five-spot setup on a $n \times n$ grid with 56 uniform control steps. Colors show saturation fields computed on the rotated setup and solid lines saturations computed on the original setup, both defined at twenty equidistant saturation values: 0.215, 0.245, . . . ,0.785.

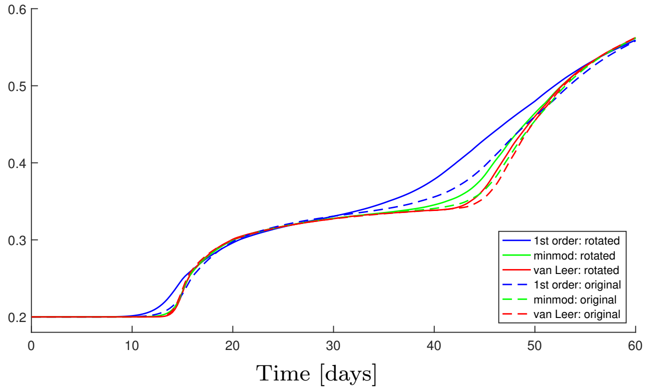

Fig. 18 Water saturation in the cell containing the producer for the five-spot case simulated using the original and rotated geometry.

weights in the higher-order reconstructions not only improves the iteration count of the nonlinear solver, while retaining the accuracy and stability of the original scheme, but also leads to a pronounced reduction in the computational cost of the automatic differentiation used for linearization. This is an important practical step towards utilization of fully implicit high-resolution schemes for contemporary simulation models of real reservoirs.

6 Acknowledgments

The work of Mykkeltvedt was funded by the Research Council of Norway and the industry partners (ConocoPhillips Skandinavia AS, Aker BP ASA, Eni Norge AS, Maersk Oil Norway AS, DONG Energy A/S, Denmark, Statoil Petroleum AS, ENGIE E&P NORGE AS, Lundin Norway AS, Halliburton AS, Schlumberger Norge AS, Wintershall Norge AS) of The National IOR Centre of Norway. Raynaud and Lie were funded by the Research Council of Norway through grant no. 244361.

References

1. Bao, K., Lie, K.A., Møyner, O., Liu, M.: Fully implicit simulation of polymer flooding with MRST. Comput. Geosci. (2017). DOI 10.1007/s10596-017-9624-5. Accepted
2. Bell, J.B., Colella, P., Trangenstein, J.A.: Higher order godunov methods for general systems of hyperbolic conservation laws. J. Comput. Phys. **82**(2), 362–397 (1989)
3. Blunt, M., Rubin, B.: Implicit flux limiting schemes for petroleum reservoir simulation. J. Comput. Phys. **102**(1), 194–210 (1992). DOI 10.1016/S0021-9991(05)80015-4
4. Chang, Y.B., Pope, G.A., Sepehrnoori, K.: A higher-order finite-difference compositional simulator. J. Petrol. Sci. Eng **5**(1), 35–50 (1990). DOI 10.1016/0920-4105(90)90004-M
5. Chen, W.H., Durlofsky, L.J., Engquist, B., Osher, S.: Minimization of grid orientation effects through use

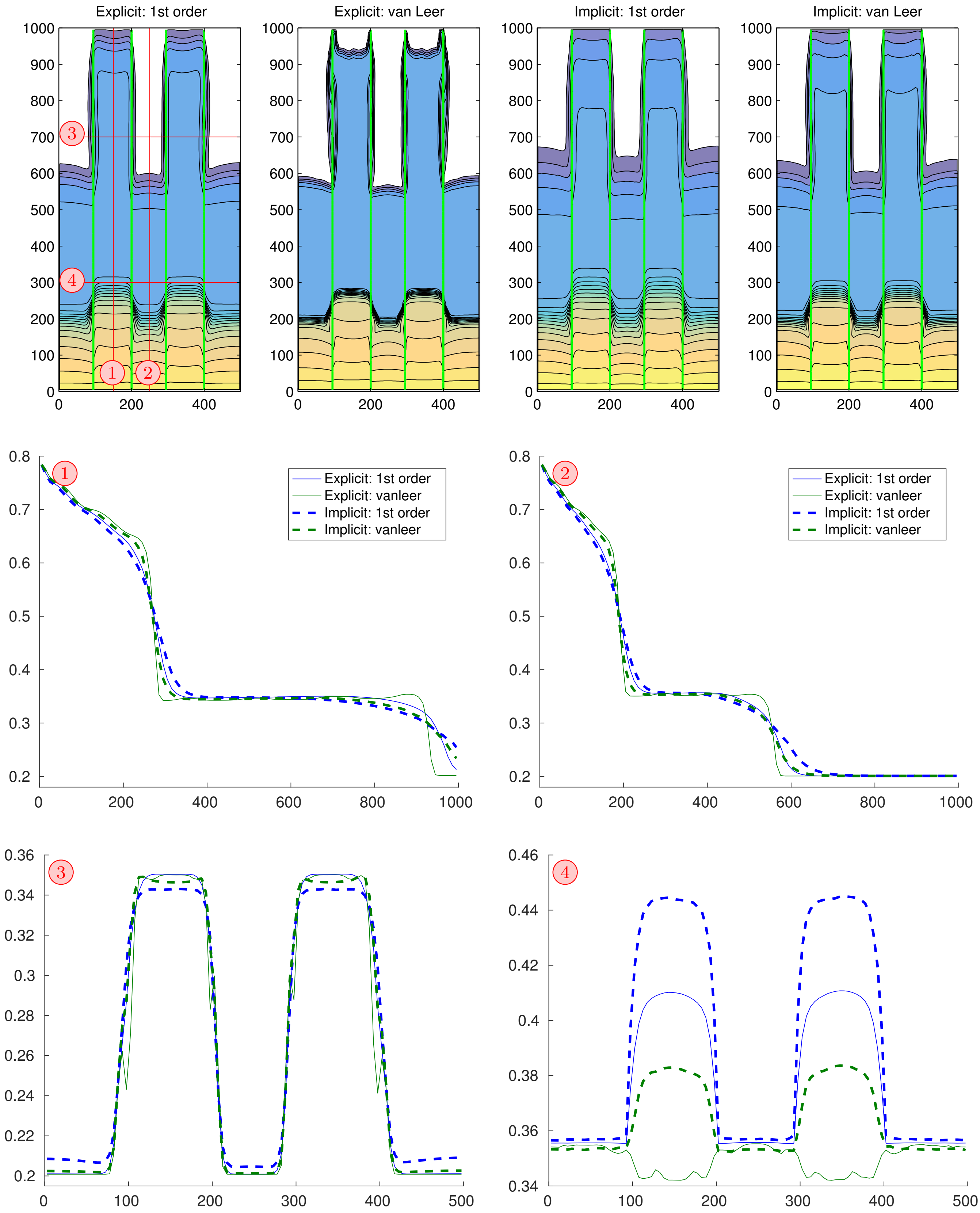

Fig. 19 Saturation profiles resulting from polymer flooding of a channelized reservoir simulated with explicit and implicit schemes with various spatial reconstructions. The green lines show the channel borders, whereas the red dotted lines indicate the cross-sections shown in the lower half of the figure. Contour lines are defined at saturation values: 0.215, 0.245,. . . ,0.785.

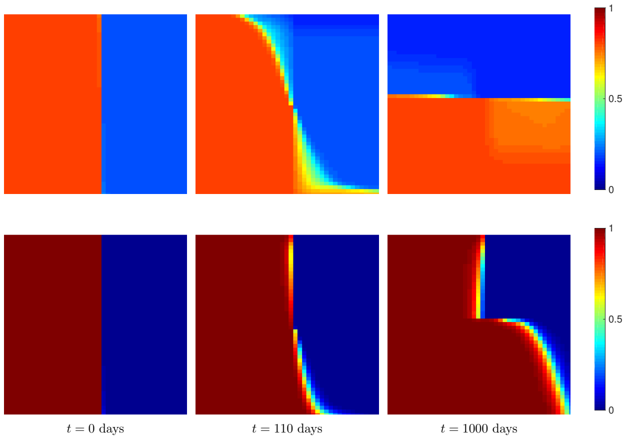

Fig. 20 Simulation of water, polymer, and oil moving due to buoyancy. The plots show water saturation (top) and polymer concentration (top) at initial time t_0 and at two later times t_1 and t_2.

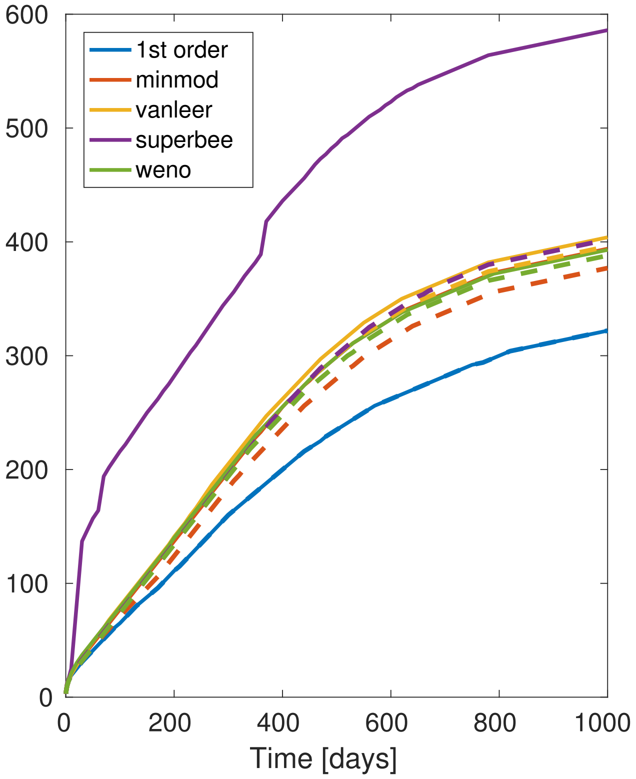

Fig. 21 Cumulative number of nonlinear iterations for the five implicit schemes; solid lines represent fully implicit reconstruction and dashed lines lagged reconstruction.

of higher order finite difference methods. SPE Advanced Technology Series **1**(2), 43–52 (1993). DOI 10.2118/22887-PA

6. Chertock, A., Kurganov, A., Petrova, G.: Fast explicit operator splitting method for convection–diffusion equations. Int. J. Num. Meth. Fluids **59**(3), 309–332 (2009)
7. DeBaun, D., Byer, T., Childs, P., Chen, J., Saaf, F., Wells, M., Liu, J., Cao, H., Pianelo, L., Tilakraj, V., Crumpton, P., Walsh, D., Yardumian, H., Zorzynski, R., Lim, K.T., Schrader, M., Zapata, V., Nolen, J., Tchelepi, H.A.: An extensible architecture for next generation scalable parallel reservoir simulation. In: SPE Reservoir Simulation Symposium, 31 January–2 Feburary, The Woodlands, Texas, USA (2005). DOI 10.2118/93274-MS
8. Delshad, M., Thomas, S.G., Wheeler, M.F.: Parallel numerical reservoir simulations of nonisothermal compositional flow and chemistry. In: SPE Reservoir Simulation Symposium, 2-4 February, The Woodlands, Texas (2009). DOI 10.2118/118847-MS
9. Duraisamy, K., Baeder, J.D., Liu, J.G.: Concepts and application of time-limiters to high resolution schemes. Journal of Scientific Computing **19**(1), 139–162 (2003). DOI 10.1023/A:1025395707090
10. Edwards, M.G.: A higher-order Godunov scheme coupled with dynamic local grid refinement for flow in a porous medium. Comp. Meth. Appl. Mech. Eng. **131**(3), 287–308 (1996). DOI 10.1016/0045-7825(95)00935-3
11. Epshteyn, Y., Riviére, B.: Fully implicit discontinuous finite element methods for two-phase flow. Appl. Numer. Math. **57**(4), 383 − 401 (2007). DOI 10.1016/j.apnum.2006.04.004
12. Gao, C.H.: Scientific research and field applications of polymer flooding in heavy oil recovery. J. Petrol. Explor. Prod. Technol. **1**, 65–70 (2011)
13. Gottlieb, S., Mullen, J.S., Ruuth, S.J.: A fifth order flux implicit WENO method. J. Sci. Comp. **27**(1-3), 271–287 (2006)
14. Gottlieb, S., Shu, C.W., Tadmor, E.: Strong stability-preserving high-order time discretization methods. SIAM Review **43**(1), 89–112 (2001). DOI 10.1137/S003614450036757X
15. Harten, A.: On a class of high resolution total-variation-stable finite-difference schemes. SIAM J. Numer. Anal. **21**(1), 1–23 (1984). DOI 10.1137/0721001
16. Harten, A., Engquist, B., Osher, S.J., Chakravarthy, S.R.: Uniformly high order accurate essentially non-oscillatory schemes. III. In: M.Y. Hussaini, B. van Leer, J. Van Rosendale (eds.) Upwind and High-Resolution Schemes, vol. 71, pp. 231–303. Springer Berlin Heidelberg, Berlin, Heidelberg (1987). DOI 10.1006/jcph.1996.5632
17. Hilden, S.T., Nilsen, H.M., Raynaud, X.: Study of the well-posedness of models for the inaccessible pore volume in polymer flooding. Trans. Porous Media **114**(1), 65–86 (2016). DOI 10.1007/s11242-016-0725-8
18. Holing, K., Alvestad, J., Trangenstein, J.A.: 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 (1990)
19. Hornung, R.D., Trangenstein, J.A.: Adaptive mesh refinement and multilevel iteration for flow in porous media. J. Comput. Phys. **136**(2), 522–545 (1997). DOI 10.1006/jcph.1997.5779
20. Jessen, K., Gerritsen, M.G., Mallison, B.T.: High-resolution prediction of enhanced condensate recovery processes. SPE J. **13**(2), 257–266 (2008). DOI 10.2118/99619-PA
21. Kaibara, M.K., Gomes, S.M.: Godunov Methods: Theory and Applications, chap. A fully adaptive multiresolution scheme for shock computations, pp. 497–503. Springer US, Boston, MA (2001). DOI 10.1007/978-1-4615-0663-8
22. Khan, S.A., Trangenstein, J.A., Horning, R.D., Holing, K., Schilling, B.E.R.: 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 (1995)

23. Knudsen, T.: Full implicit WENO scheme for two phase flow in reservoir simulation. Master's thesis, Norwegian University of Science and Technology (2014)
24. Krogstad, S., Lie, K.A., Møyner, O., Nilsen, H.M., Raynaud, X., Skaflestad, B.: 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, pp. 1–25 (2015). DOI 10.2118/173317-MS
25. Lake, L.W.: Enhanced Oil Recovery. Prentice-Hall (1989)
26. Li, X., Zhang, D.: A backward automatic differentiation framework for reservoir s imulation. Comput. Geosci. pp. 1–14 (2014). DOI 10.1007/s10596-014-9441-z
27. Lie, K.A.: An Introduction to Reservoir Simulation Using MATLAB: User guide for the Matlab Reservoir Simulation Toolbox (MRST). SINTEF ICT, www.sintef.no/MRST (2016)
28. Liu, J., Pope, G.A., Sepehrnoori, K.: 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, pp. 35–50 (1995)
29. Liu, X.D., Osher, S.J., Chan, T.: Weighted essentially non-oscillatory schemes. J. Comput. Phys. **115**(1), 200–212 (1994)
30. Matthäi, S.K., Mezentsev, A.A., Pain, C.C., Eaton, M.D.: A high-order TVD transport method for hybrid meshes on complex geological geometry. Int. J. Numer. Meth. Fluids **47**(10-11), 1181–1187 (2005). DOI 10.1002/fld.901
31. MRST: The MATLAB Reservoir Simulation Toolbox. www.sintef.no/MRST (2015b)
32. Neidinger, R.: Introduction to automatic differentiation and MATLAB object-oriented programming. SIAM Review **52**(3), 545–563 (2010). DOI 10.1137/080743627
33. Oldenburg, C.M., Pruess, K.: Simulation of propagating fronts in geothermal reservoirs with the implicit leonard total variation diminishing scheme. Geothermics **29**(1), 1–25 (2000). DOI 10.1016/S0375-6505(99)00048-6
34. Roe, P.L.: Characteristic-based schemes for the Euler equations. Ann. Rev. Fluid, Mech. **18**, 337–365 (1986). DOI 10.1146/annurev.fl.18.010186.002005
35. Sorbie, K.S.: Polymer-Improved Oil Recovery. Springer Science & Business Media (1991)
36. Sweby, P.K.: High resolution schemes using flux limiters for hyperbolic conservation laws. SIAM J. Numer. Anal. **21**(5), 995–1011 (1984)
37. Todd, M.R., Longstaff, W.J.: The development, testing and application of a numerical simulator for predicting miscible flood performance. J. Petrol. Tech. **24**(7), 874–882 (1972)
38. Trangenstein, J.A.: Numerical solution of hyperbolic partial differential equations. Cambridge University Press, Cambridge (2009)
39. van Leer, B.: Towards the ultimate conservative difference scheme. II. monotonicity and conservation combined in a second-order scheme. J. Comput. Phys. **14**(4), 361–370 (1974)
40. van Leer, B.: Towards the ultimate conservative difference scheme III. upstream-centered finite-difference schemes for ideal compressible flow. J. Comput. Phys. **23**(3), 263–275 (1977)
41. Voskov, D., Younis, R., Tchelepi, H.: General nonlinear solution strategies for multi-phase multi-component EoS based simulation. SPE Reservoir Simulation Symposium Proceedings **1**(February), 649–663 (2009)
42. Voskov, D.V., Tchelepi, H.A.: Comparison of nonlinear formulations for two-phase multi-component eos based simulation. J. Petrol. Sci. Engrg. **82-83**(0), 101–111 (2012). DOI 10.1016/j.petrol.2011.10.012
43. Yee, H., Warming, R., Harten, A.: Implicit total variation diminishing (tvd) schemes for steady-state calculations. Journal of Computational Physics **57**(3), 327–360 (1985). DOI 10.1016/0021-9991(85)90183-4
44. Zhou, Y., Tchelepi, H.A., Mallison, B.T.: Automatic differentiation framework for compositional simulation on unstructured grids with multi-point discretization schemes. In: SPE Reservoir Simulation Symposium, 21-23 February, The Woodlands, Texas (2011). DOI 10.2118/141592-MS. SPE 141592-MS